



UADY
UNIVERSIDAD
AUTÓNOMA
DE YUCATÁN

PLAN DE ESTUDIOS
DE LA
LICENCIATURA EN INGENIERÍA DE
SOFTWARE

FACULTAD DE MATEMÁTICAS
Campus de Ciencias Exactas e Ingenierías

Aprobado en Sesión Extraordinaria por el H. Consejo
Universitario

12 de agosto de 2016

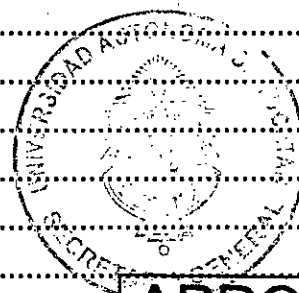


Mérida, Yucatán

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

INDICE

1. DATOS GENERALES	3
2. FUNDAMENTACIÓN.....	4
2.1. Antecedentes	4
2.1.1 Justificación	5
2.2. Estudio de referentes.....	6
2.2.1 Referente social	6
2.2.2 Referente disciplinar	14
2.2.3 Referente profesional.....	22
2.2.4 Referente institucional	23
2.3. Justificación de la pertinencia social y factibilidad del programa.....	26
2.4. Pertinencia social.....	26
2.5. Factibilidad.....	28
2.6. Evaluación interna y externa del programa	31
2.6.1 Evaluación interna	31
2.6.2 Evaluación externa	36
2.7. Conclusiones generales	42
2.7.1 Justificación de las Áreas de Competencia.....	43
3. INTEGRACIÓN DE LOS EJES DEL MEFI.....	45
4. OBJETIVO GENERAL DEL PLAN DE ESTUDIOS.....	47
5. PERFIL DE INGRESO	48
6. PERFIL DE EGRESO	51
6.1. Áreas de competencia	51
6.2. Competencias de egreso	51
6.3. Desagregado de saberes	52
6.4. Competencias disciplinares	58
7. ESTRUCTURA CURRICULAR.....	59
7.1. Características del Plan de Estudios.....	59
7.2. Organización del Plan de Estudios.....	59
8. MALLA CURRICULAR	63
8.1. Asignaturas optativas.....	65



APROBADO

17 AGO 2016

Comisión Permanente Académica
Consejo Universitario



9.	ESQUEMA DE CONSISTENCIA.....	66
9.1.	Matriz de consistencia de las asignaturas en relación con las competencias de egreso	66
9.2.	Esquema de consistencia por competencia de egreso	68
9.3.	Matriz de las competencias genéricas por asignatura.	80
10.	PROGRAMAS DE ESTUDIO	82
11.	METODOLOGÍA DE EVALUACIÓN DEL PLAN DE ESTUDIOS.....	266
12.	FUNCIÓN ACADÉMICO ADMINISTRATIVA	268
12.1.	Introducción.....	268
12.2.	Calendario escolar	268
12.3.	Ingreso.....	268
12.4.	Permanencia	268
12.5.	Sugerencia de asignaturas obligatorias por período lectivo	270
12.6.	Prácticas Profesionales.....	271
12.7.	Servicio Social	271
12.8.	Emprendedores	272
12.9.	Movilidad	272
12.10.	Inglés como segundo idioma.....	272
12.11.	Titulación	273
12.12.	Plan de liquidación	273
13.	PLAN DE DESARROLLO.....	275
13.1	Visión a 2020.....	275
13.2	Objetivos estratégicos	275
14.	REFERENCIAS.....	283
	ANEXO A. PROFESORES QUE ELABORARON LOS PROGRAMAS DE ESTUDIOS.....	286
	ANEXO B. CUESTIONARIO PARA PROFESORES.....	288
	ANEXO C. CUESTIONARIO PARA ALUMNOS	297

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



1. DATOS GENERALES

Nombre del programa:

Licenciatura en Ingeniería de Software

Título a otorgar:

Ingeniero(a) de Software

Responsable de la propuesta:

Dr. Ramón Peniche Mena
Director

Cuerpo directivo de la Facultad:

M. en C. Martha Imelda Jarero Kumul, Secretaria Académica
M. en C. Carlos Benito Mojica Ruiz, Secretario Administrativo
Dr. Jorge Armando Argáez Sosa, Jefe de la Unidad de Posgrado e Investigación

Grupo diseñador la propuesta:

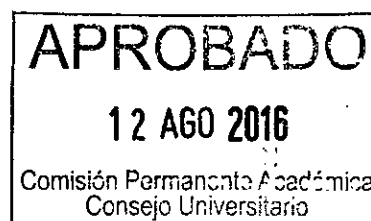
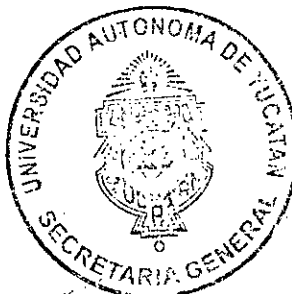
Dr. Raúl Antonio Aguilar Vera
M. en C. Edgar Antonio Cambranes Martínez
M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
M.T.I. Julio César Díaz Mendoza, E.D.
Dr. Juan Pablo Ucán Pech

Asesores:

M.I.N.E. Sandra Trinidad Martín Tun
Mtra. Jéssica Zumárraga Ávila

Fecha propuesta de inicio:

Agosto de 2016



2. FUNDAMENTACIÓN

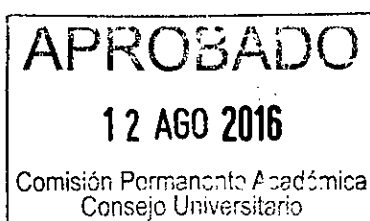
2.1. Antecedentes

Con el propósito de elevar y extender la competitividad del país mediante la promoción del uso y aprovechamiento de las Tecnologías de la Información (TI), la difusión de la cultura tecnológica local y extranjera, el impulso al desarrollo de la industria de TI, así como el fomento y propagación de la industria de desarrollo de software, la Facultad de Matemáticas propuso la Licenciatura en Ingeniería de Software para coadyuvar a la sociedad mediante la generación de recursos humanos para el mercado local, donde dicho personal tenga la capacidad de continuar sus estudios para realizar investigación en el área de la Ingeniería de Software, aspecto que marca la diferencia en la profesionalización de esta disciplina.

El plan de estudios de La Licenciatura en Ingeniería de Software (LIS) se integró a la oferta académica de la Facultad de Matemáticas en septiembre de 2004 —luego de ser aprobado por el H. Consejo Universitario en la sesión extraordinaria del 29 de junio del mismo año— y corresponde a uno de los cuatro perfiles de profesionistas en Informática y Computación propuestos por la Asociación Nacional de Instituciones de Educación en Tecnologías de la Información (ANIEI).

El plan de estudios ha sufrido una sola modificación, realizada en el año 2009 y consistió en incorporar elementos del modelo educativo de la Institución, específicamente en relación a la flexibilidad curricular, con lo que, actualmente los alumnos pueden definir el ritmo de su avance por medio de la decisión del número de asignaturas a cursar en un período y la elección de asignaturas que le permita especializarse en un área de su interés. Desde su creación, a esta licenciatura ha ingresado un total de 496 alumnos. En el período agosto-diciembre de 2015, se registró una matrícula de 186 estudiantes.

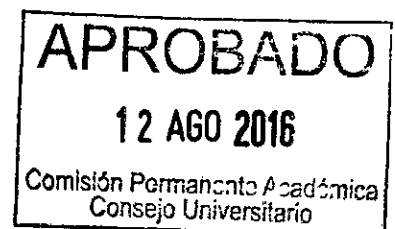
Conforme al Plan de Desarrollo Institucional 2014-2022 de la Universidad (PDI, 2015) y al Modelo Educativo para la Formación Integral (MEFI, 2012), se ha realizado una evaluación interna y externa teniendo como resultado la presente propuesta de modificación, que contempla alinear el plan de estudios al MEFI, satisfacer las necesidades actuales del sector productivo, así como responder a los estándares nacionales e internacionales en la formación de un Ingeniero de Software. Todo esto con la finalidad de proporcionar a los egresados una educación integral mediante un programa de calidad, que les permita una rápida incorporación a la industria de las TI.



2.1.1 Justificación

La evaluación y modificación al plan de estudios de la Licenciatura en Ingeniería de Software, que se presenta en este documento, se justifica por las siguientes razones:

1. El plan de estudios original tiene 12 años de haber sido implementado y aunque sufrió una modificación en 2009 —flexibilidad en el régimen académico-administrativo— a la fecha no se ha realizado una evaluación integral. Sin embargo, la industria del software, el desarrollo tecnológico y científico en la disciplina de Ingeniería de Software ha tenido un vertiginoso avance en los últimos años. Debido a esto, se tiene la necesidad de contar con profesionales en TI y, en particular, Ingenieros de Software que tengan la preparación adecuada para afrontar los retos en esta área.
2. Es necesario contar en todo momento con un programa educativo pertinente, que satisfaga las necesidades del sector productivo y permita a sus egresados contribuir con sus conocimientos, habilidades y aptitudes en ámbitos locales, nacionales e internacionales, para lo cual, es importante considerar las opiniones de egresados, expertos en el área disciplinar y empleadores, para validar o modificar, en su caso, el perfil de egreso.
3. De acuerdo con el Plan de Desarrollo Institucional se requiere alinear el plan de estudios al Modelo Educativo de Formación Integral (MEFI).
4. La Guía del Cuerpo de Conocimientos de la Ingeniería de Software (SWEBOK), propuesta por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) en 2004 ha sido actualizada en 2014.
5. Es necesario atender las recomendaciones emitidas por los organismos evaluadores externos (CIEES, CONAIC) en lo que respecta al Plan de Estudios.





2.2. Estudio de referentes

2.2.1 Referente social

Contexto social global

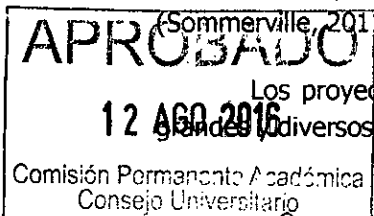
En la actualidad, nuestras sociedades no podrían funcionar sin grandes sistemas de software profesionales. Para construir sistemas empresariales existe una gran variedad de tecnologías que apoyan el desarrollo y la implementación de grandes aplicaciones empresariales. Los servicios públicos y la infraestructura nacionales (energía, comunicaciones y transportes) se apoyan en sistemas de cómputo complejos y fiables. El software ha permitido la exploración del espacio y la creación de la World Wide Web, el sistema de información más significativo en la historia de la humanidad. El mundo ahora enfrenta un nuevo conjunto de desafíos: cambio climático y temperaturas extremas, agotamiento de los recursos naturales, una creciente población mundial que demanda alimentos y vivienda, terrorismo internacional y la necesidad de ayudar a los adultos mayores a tener vidas satisfactorias y plenas. Necesitamos nuevas tecnologías para enfrentar todos esos problemas y, desde luego, el software desempeñará un papel central en dichas tecnologías (Sommerville, 2011).

En una sociedad tan competitiva como la nuestra el manejo de grandes cantidades de información hace necesario el uso de las computadoras, pero los vertiginosos cambios en la industria de la computación pocas veces van aparejados con los cambios que se dan en la sociedad, por lo cual sólo refuerzan la visión reduccionista o mecanicista en donde se ve al hombre como a una máquina y a su cerebro como a una computadora, lo que de continuar así nos conducirá a ser una sociedad deshumanizada que ve a las personas únicamente como procesadores de información. Para combatir tan limitada percepción es necesario que el cambio hacia el paradigma sistémico que se está gestando permee en la sociedad; para ello es importante formar recursos humanos que hagan suyo el conocimiento de la computación, así como su relación con las demás ciencias (naturales, exactas, humanidades y sociales) para desarrollarse integralmente, y con sus ideas y sus habilidades, contribuyan a la solución responsable de problemas planteados por la sociedad y así coadyuven al desarrollo integral de la sociedad; entendiendo por solución responsable aquella que no afecte negativamente a las otras partes del sistema y por lo menos permita brindar a las generaciones futuras las mismas oportunidades con las que nosotros contamos (UPAEP, 2014).

La ingeniería de software es una disciplina relativamente joven, fue propuesta inicialmente en 1969 durante una conferencia de la Organización del Tratado del Atlántico Norte (OTAN) y “comenzó su desarrollo en respuesta a los problemas de construcción de sistemas de software personalizados para las fuerzas armadas, el gobierno y las aplicaciones industriales. Actualmente se desarrolla software de amplio espectro, desde juegos en consolas especializadas, productos para computadoras personales y sistemas basados en Web, hasta sistemas distribuidos de gran escala”

(Sommerville, 2011, p. v).

Los proyectos exitosos donde se han aplicado métodos de ingeniería de software son tan diversos que incluyen el software del transbordador espacial, sistemas de control de tráfico





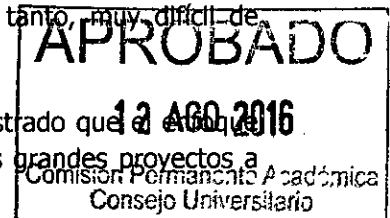
aéreo, sistemas para realizar o asistir operaciones quirúrgicas, sistemas administrativos empleados por corporativos multinacionales, sistemas bancarios que en segundos realizan grandes transacciones monetarias, por citar tan sólo algunos dominios de aplicación.

Como puede observarse, la ingeniería de software ha revolucionado la cultura del mundo debido al uso extendido de la computadora. Algunos servicios de Internet como el correo electrónico, la World Wide Web y la mensajería instantánea permiten a las personas interactuar en nuevas formas. De igual manera, el software disminuye el costo y mejora la calidad de un gran número de servicios sociales.

Desde el punto de vista económico, el software es un medio de producción englobado dentro del capital intelectual de la empresa (Baetjer, 1998); y es precisamente este capital el que debe desarrollarse dentro de las universidades.

Según Sommerville (2011), la ingeniería de software afronta varios problemas importantes:

1. Cada vez más, se requiere que los sistemas operen como sistemas distribuidos en redes que incluyen diferentes tipos de computadoras y con diferentes clases de sistemas de soporte. A menudo es necesario integrar software nuevo con sistemas heredados más viejos escritos en diferentes lenguajes de programación. Dicha problemática nos conduce a desarrollar técnicas para construir software confiable que sea lo suficientemente flexible para adecuarse a la heterogeneidad.
2. Muchas técnicas tradicionales de ingeniería de software demandan tiempo. El tiempo que éstas consumen es para producir un software de calidad. Sin embargo, los negocios actuales deben tener una gran capacidad de respuesta y cambiar con mucha rapidez. Su software de soporte debe también debe cambiar con la misma rapidez. Por lo que es preciso reducir los tiempos de entrega para sistemas grandes y complejos sin comprometer la calidad del sistema.
3. Puesto que el software tiene relación con todos los aspectos de nuestra vida, es esencial que podamos confiar en él. Esto es especialmente importante en sistemas remotos de software a los que se accede a través de páginas Web o de interfaces de servicios Web. Este reto de confianza consiste en desarrollar técnicas que demuestren que los usuarios pueden confiar en el software.
4. El desarrollo costeable de sistemas de software es abstracto e intangible; no está restringido por materiales, o gobernado por leyes físicas o por procesos de manufactura. De alguna forma, esto simplifica la ingeniería de software ya que no existen limitaciones físicas del potencial del software. Sin embargo, esta falta de restricciones naturales significa que el software puede llegar a ser extremadamente complejo y, por lo tanto, muy difícil de entender.
5. La experiencia acumulada en la construcción de sistemas ha demostrado que el enfoque informal para el desarrollo de software no resulta conveniente. Los grandes proyectos a





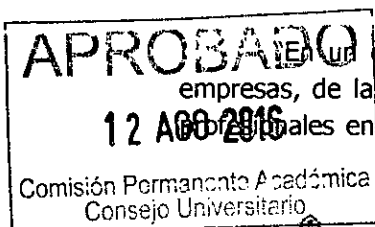
menudo tienen años de retraso, llegan a costar mucho más de lo presupuestado, son irrealizables, difíciles de mantener y exhiben un desempeño pobre. Aunado a esto, en el mundo existe un gran número de personas que están escribiendo software sin apearse al aspecto disciplinario de la ingeniería de software. Dicha problemática no radica en el hecho de que personas sin una formación especializada estén creando programas de computadora, sino en la calidad de los productos creados, los cuales muchas veces pueden desencadenar situaciones catastróficas debido a que en algunos casos las fallas tienen un precio elevado. "Cuando el software de computadora falla es posible que se pierdan vidas o surjan graves consecuencias económicas" (Pressman, 2006, p. 830). Lo expuesto anteriormente nos lleva a dejar de lado la informalidad y propiciar el uso y aplicación de nuevas técnicas y métodos para controlar la complejidad inherente de los grandes sistemas.

La amplia variedad de sistemas y de organizaciones que los usan, indican que se necesita una diversidad de enfoques en el desarrollo de software. Sin embargo, las nociones fundamentales de procesos y la organización del sistema son la base de todas estas técnicas y éstas son la esencia de la ingeniería de software. Los ingenieros de software deberán comprometerse a convertir el análisis, especificación, diseño, implementación, pruebas y mantenimiento de software en una profesión respetada y benéfica.

A partir de las ideas presentadas, se puede apreciar que en el área de ingeniería de software existe una gran demanda de profesionales para la construcción de software de alta calidad. Después de reflexionar sobre los aspectos socioprofesionales que influyen de una u otra manera en el desarrollo del currículum informático, es importante tenerlos presentes durante todo el diseño de un plan de estudios (ANECA, 2004).

Las oportunidades de desarrollo profesional a nivel nacional e internacional en las áreas de Ingeniería de Software y Tecnologías de la Información y Comunicaciones (TIC) son grandes. Con la finalidad de establecer un marco de referencia basado en términos conceptuales de actualidad, se define Tecnologías de Información y Comunicaciones como: "Las TIC son el conjunto de conocimientos científicos-tecnológicos, prácticas y métodos que permiten el procesamiento de información y comunicación de datos a través de computadoras y medios digitales para crear, generar u optimizar el desempeño de productos, servicios, procesos, equipos, instalaciones y soluciones integrales de interés para la sociedad, instituciones, gobierno y empresas en general que les permitan ser más eficientes y cumplir sus propósitos de bienestar económico y social" (RedTIC, 2011, p. 8).

Hace algunos años el contexto profesional de las Tecnologías de la Información (TI), particularmente en el ámbito de la Ingeniería de Software, venía marcado por la enorme necesidad de profesionales cualificados en el mercado laboral. Hoy en día, esta realidad continúa marcada por la gran demanda de estos profesionales y, paralelamente, puede apreciarse un interés que va en aumento por consolidar lo que constituye la identidad profesional informática (ANECA, 2004).



cohesión social de los países” (ANECA, 2004, p. 149), principalmente de aquellos en vías de desarrollo. “No obstante, existen datos que indican que el futuro próximo seguirá marcado por un déficit de profesionales de TI” (ANECA, 2004, p. 149).

Al igual que otras disciplinas de la ingeniería, la ingeniería de software se lleva a cabo dentro de un marco legal y social que limita la libertad de los ingenieros. Los ingenieros de software deben considerar que su trabajo comprende responsabilidades mucho más amplias que la simple aplicación de habilidades técnicas; deben comportarse de una forma ética y moral responsable si es que desean ser respetados como profesionales. No resulta suficiente con afirmar que poseen estándares normales de honestidad e integridad, pues se asume que no utilizarán su capacidad y sus habilidades para comportarse de forma deshonesto; sin embargo, existen áreas donde los estándares de comportamiento aceptable no están acotados por las leyes, sino por la más tenue noción de responsabilidad profesional. Algunas de éstas son: confidencialidad, competencia, derechos de propiedad intelectual y uso apropiado de las computadoras (Sommerville, 2011).

Como muestra de la preocupación que ha generado la existencia de áreas no acotadas dentro del marco de la legalidad, organizaciones tales como la Association for Computing Machinery (ACM), el Institute of Engineering Electricians and Electronics (IEE) y la British Computer Society han publicado un código de conducta profesional o de ética adherido a ocho principios básicos: sociedad, cliente y empresario, producto, juicio, administración, profesión, colegas y personal (ACM, 1992).

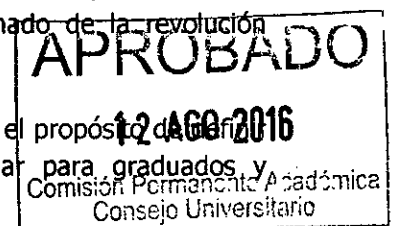
Por lo anterior, tal parecería que la educación teórica de la ingeniería de software no es suficiente para proveer las habilidades y el conocimiento de los ingenieros de software que los empleadores esperan. Varol y Bayrak (2005), señalan que es necesario otorgar a los estudiantes de ambientes reales de aprendizaje que los preparen para las expectativas del futuro.

Como consecuencia de lo que se ha venido exponiendo, “podríamos preguntarnos si nuestras universidades están cumpliendo con su objetivo social de cubrir adecuadamente la demanda de profesionales del mercado. Ya que no únicamente interesa la cantidad de profesionales sino, mayormente, la calidad y la adecuación a la formación impartida. Cabe plantearse si las universidades, en general, están satisfaciendo adecuadamente la demanda social o deberían replantearse algunos aspectos para consolidar la Informática como profesión y no sólo como una disciplina científica” (ANECA, 2004, p.150). Es decir, posicionar a la Ingeniería de Software así como a los programas educativos que la impulsan, de acuerdo a la alta relevancia que poseen.

Contexto social nacional

La Asociación Nacional de Instituciones de Educación en Tecnologías de Información (ANIEI) contribuye a la formación de profesionales en Informática y Computación sólidamente preparados, e impulsa la difusión y la asimilación de una cultura computacional en la sociedad, acorde a las exigencias de un entorno cada vez más informatizado, y el futuro emanado de la revolución informática presente.

Por otro lado, la IEEE y la ACM han realizado trabajos conjuntos con el propósito de definir un código de ética y estándares profesionales, definir el mapa curricular para graduados y Comisión Permanente Académica Consejo Universitario





posgraduados en Ingeniería de Software y definir un cuerpo de conocimientos y recomendaciones prácticas para la ingeniería de software.

Estas organizaciones marcan las tendencias académicas y profesionales a nivel nacional e internacional en el área de cómputo y proporcionan una orientación más precisa para definir el cuerpo de conocimientos necesario para ejercer con las competencias pertinentes que se demandan a un profesionista de ingeniería de software.

En los últimos años, después de un profundo estudio de las necesidades de la industria y respondiendo a sus demandas, la ANIEI propuso los modelos o perfiles paracurriculares enfocados en cubrir las carencias en el área de programación, diseño y administración de proyectos de software: Ingeniero de Software, Emprendedor de Negocios de Software, Arquitecto de Software, Administrador de Proyectos, Desarrollador de Software. Estos modelos permiten a un egresado especializarse en su área de interés en cuanto a sus competencias, de manera que pueda ser más competitivo y prestar mejores servicios a la industria.

Entre las competencias específicas que la industria requiere se encuentran las siguientes: Tecnológicas (Gerencia de proyectos, Servicios Orientados a Web, Arquitecturas Orientadas a Servicios, Inteligencia de Negocios, entre otras), Competencias de Negocios (Visión, Mejora de procesos, comunicación oral y escrita, técnicas de negociación, cultura de calidad, entre otras) y Personales (Liderazgo, trabajo en equipo, innovación y creatividad, dominio del idioma inglés, y otras).

Una investigación realizada por Villalobos y Gutiérrez (2001) sobre las prácticas de ingeniería de software en México, ofrece una perspectiva de la administración de proyectos informáticos y el impacto de la calidad del software como ventaja competitiva. A pesar de que el estudio fue realizado hace más de una década, éste refleja problemáticas y necesidades que lamentablemente aún se siguen presentando en nuestro país:

El área de gestión y control de la calidad en proyectos de desarrollo de software en México es relativamente nueva y las empresas si bien reconocen la importancia de la calidad, no se encuentran suficientemente preparadas para aceptar los nuevos retos que trae consigo y poner en práctica sus principios y técnicas.

En el caso de los modelos de gestión y aseguramiento de la calidad, aun es poca la difusión de los mismos en la comunidad dedicada a esta área, y se necesita no sólo conocer las perspectivas internacionales de la calidad, sino también informar de las consecuencias que podría ocasionar el implementar alguno de los modelos.

Las organizaciones que desarrollan software en México pertenecen en su mayoría a pequeños y microempresarios que no se interesan por utilizar métodos de control en sus proyectos, su perspectiva radica en cumplir sólo con los requerimientos en las fechas estipuladas, sin ofrecer a los clientes servicios de valor agregado. Los líderes de proyecto deberían probar controles efectivos suficientemente flexibles para ajustarse a los cambios o tomar ventajas de nuevas oportunidades. La necesidad de crear e implementar un proceso continuo para la administración efectiva incluye a todos

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

"Educando con Pertinencia, Trascendiendo con Relevancia" 10

los miembros (proveedores, clientes, desarrolladores). Balanceando los costos, programas y las consideraciones técnicas, pensando en futuro identificando incertidumbres, anticipando las entradas potenciales, administrando los recursos de los proyectos y actividades. Por lo mismo es importante centrarse en la calidad del producto como un medio para obtener mejores productos y a partir de ahí, influir en la calidad del proceso.

Por su parte, el Plan Nacional de Desarrollo (PND, 2013) menciona que “una de las características más notables del caso mexicano es la desvinculación entre los actores relacionados con el desarrollo de la ciencia y la tecnología y las actividades del sector empresarial”. Y enfatiza la relevancia de “alinear las visiones de todos los actores del Sistema de Ciencia, Tecnología e Innovación para que las empresas aprovechen las capacidades existentes en las instituciones de educación superior y centros públicos de investigación”.

De forma tal que sí se “requiere consolidar la continuidad y disponibilidad de los apoyos necesarios para que los investigadores en México puedan establecer compromisos en plazos adecuados para abordar los problemas científicos y tecnológicos relevantes permitiéndoles situarse en la frontera del conocimiento y la innovación, y competir en los circuitos internacionales” (PND, 2013).

Contexto social regional y local

En este apartado se presentan datos sociodemográficos que brindan un panorama del crecimiento, necesidades y problemáticas de la región donde los egresados de la Licenciatura en Ingeniería de Software tendrán impacto.

El Estado de Yucatán está conformado por 106 municipios y la zona metropolitana de Mérida es la más densamente poblada; está conformada por los municipios de Mérida, Kanasín, Umán, Conkal y Ucú, que en su conjunto aglutinan al 50% de la población del estado. Según el Instituto Nacional de Estadística y Geografía (INEGI, 2011), la población total del estado de Yucatán era de 1,956,000 habitantes y la edad mediana de la población es de 26 años. Cabe destacar que el 65.7% de la población se encuentra en el rango de edad de 15 a 64 años, los menores de 14 años representan 27.4% y los mayores de 65 años el 6.9%.

La evolución de la población del grupo en edad de asistir a la educación superior en la entidad federativa (18 a 22 años, respectivamente), en el período 2014-2025, representa un aspecto fundamental a considerar en la planeación de la demanda educativa, en particular ante los cambios recientes a la Ley General de Educación.

En términos de rezago, Yucatán está clasificado como un estado con un alto grado de rezago social, ocupando la posición número 9 en el país. Esta problemática constituye uno de los principales retos en materia de política social, en la cual la Universidad Autónoma de Yucatán (UADY) puede participar aportando iniciativas de solución, aprovechando sus capacidades.

En 2010, el porcentaje de la población de 16 años o más nacida hasta 1981 con rezago educativo registró un porcentaje de 35.6%; en lo que respecta a la población de 16 años o más



APROBADO

17 AGO 2016

Comisión Permanente Académica
Consejo Universitario





nacida a partir de 1982 representó el 25.2%. Esta problemática ha sido identificada como un área de oportunidad para la UADY.

El estado cuenta con una importante infraestructura educativa que ha ido creciendo a lo largo de las últimas décadas. En la actualidad cuenta con 184 instituciones de educación superior. Un reto importante en los próximos años consiste en aumentar significativamente la tasa de eficiencia del sistema educativo estatal para continuar incrementando las tasas de cobertura de la educación superior (PDI, 2015).

La tasa de cobertura de todos los tipos educativos se ha incrementado en los últimos cinco ciclos escolares como resultado de la ampliación y diversificación de los servicios educativos en el estado. Sin embargo, éstas disminuyen conforme se transita de la educación primaria a la educación superior. La matrícula total en el ciclo escolar 2011-2012 ascendió a 55,119 estudiantes, de los cuales 31,008 estaban realizando sus estudios en instituciones públicas (56.3%) y 24,111 en instituciones particulares (43.7%). La matrícula de la Universidad Autónoma de Yucatán (12,914 estudiantes) representa el 23.4% de la matrícula total y el 42% de la correspondiente al régimen público, lo que da cuenta de la importancia de su participación en la educación superior en el estado. No obstante, es importante resaltar que del total de la población ocupada en Yucatán, sólo 145 mil (15%) cuenta con estudios de nivel superior.

En cuanto al panorama laboral, la evolución de la población económicamente activa que se reporta en la Encuesta Nacional de Ocupación y Empleo (ENOE), muestra un crecimiento del 14% en el período 2008-2013. De acuerdo al INEGI, Yucatán ocupa el lugar 26 en el ámbito nacional, con una tasa de condiciones críticas de ocupación del 15.5%; el lugar 28 en el ámbito nacional, con una tasa de ocupación en el sector informal del 33.1%; el lugar 2 en el ámbito nacional, con una tasa de desocupación del 2.5%; el lugar 9 en el ámbito nacional, con una tasa de subocupación del 6% y el lugar 15, con el 34.4% de trabajadores remunerados que no tienen acceso a ninguna de las prestaciones laborales de ley (aguinaldo, vacaciones con goce de sueldo y reparto de utilidades).

Por otra parte, de acuerdo con los resultados de la Encuesta de Expectativas de Empleo para el primer trimestre de 2014, de la empresa consultora Manpower, los empresarios de Yucatán reportaron expectativas de contratación por debajo de la tendencia neta nacional. En particular, la capital del estado se ubicó como la tercera ciudad con menores expectativas de contratación por arriba de la ciudad de Hermosillo y de Guadalajara, al presentar una tendencia neta de empleo de +13%, lo que constituye un importante reto para la incorporación de los egresados de la educación media superior y superior en el estado en la medida en que estos egresados cuenten con una sólida formación para el desempeño de las ocupaciones y actividades requeridas.

En la información proporcionada por el INEGI con respecto a los Censos Generales de Población de 1930 al 2010, se observa que en los últimos 20 años, en Yucatán la población ocupada en el sector terciario se ha incrementado en 14.8 puntos porcentuales y el sector primario ha

decrecido en 15.4 puntos porcentuales. El 88.3% del personal ocupado total de la entidad se concentró en las actividades económicas de Servicios (33.4%), Comercio (32.3%) e Industrias manufactureras (22.6%).

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





En Yucatán los clústeres actuales son el logístico y el turístico. Se prevé que en el futuro sean el agroindustrial, el de investigación, el de innovación y el de energía renovable.

En cuanto a la participación de los sectores económicos, es preciso mencionar que el Producto Interno Bruto (PIB) del estado ascendió a más de 187 mil millones de pesos en 2011, con lo que aportó 1.35% al PIB nacional. Yucatán cuenta con un total de 85,844 unidades económicas (Censo Económico, 2009), de las cuales 42.71% corresponde al sector de comercio; 21.46%, a la industria manufacturera; y 32.76%, a servicios privados no financieros.

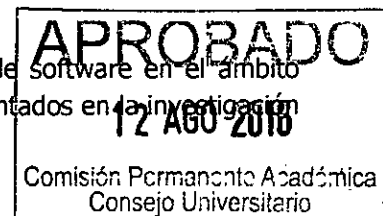
Según datos del Instituto Mexicano para la Competitividad (IMCO), Yucatán se colocó en el lugar 19 de 32 en el Índice de Competitividad Estatal, y ganó dos posiciones entre 2008 y 2010. Es importante señalar que la Ciudad de Mérida ocupa el sexto lugar a nivel nacional en competitividad.

De acuerdo con la Organización para la Cooperación y el Desarrollo Económicos (OCDE), Yucatán es un estado con un gran potencial de crecimiento. El estado se desempeña mejor que los promedios de esta organización en seis de 12 factores, sobre todo con respecto a las categorías de permisos de construcción y registro de la propiedad. En contrapartida, el estado se desempeña abajo del promedio de la OCDE en los tres factores relacionados con la apertura de una empresa: número de trámites, tiempo y costo. Yucatán se encuentra arriba de los promedios nacionales en cinco de 12 factores. Entre las cuatro categorías, se desempeña mejor en el nivel nacional en lo que respecta a la apertura de una empresa, al situarse en el cuarto lugar (Plan de Acción, 2012).

Por otro lado, el reporte de la OCDE sobre innovación regional señala que en Yucatán, la tasa de crecimiento del PIB estatal es superior a la media nacional, las tasas de participación y de empleo son altas; que los sectores turísticos y de logística están muy avanzados y son competitivos, que la capacidad científica está bien instalada, que Mérida se encuentra entre las mejores ciudades en calidad de vida y que la tasa de desempleo se encuentra por debajo de la media nacional. Sin embargo, no se ha logrado propiciar un entorno sólido para la innovación con un enfoque sistemático donde se involucren diversos actores (administración pública federal y estatal, IES públicas y particulares y empresas) y se contemplen los siguientes elementos: generación de conocimientos e ideas; una demanda por ideas y soluciones; la integración de un mercado nacional e internacional, el financiamiento de proyectos y de empresas, la formación de recursos humanos básicos y técnicos, así como un marco regulatorio que proteja la propiedad intelectual.

Para contrarrestar esta situación, el Plan Estatal de Desarrollo 2012-2018 (PED, 2013), señala en el apartado sobre Innovación y Economía del Conocimiento que "la industria de tecnologías de la información es un elemento transversal para el fortalecimiento de los sectores productivos. Para ello se impulsará el agrupamiento de las empresas de la industria y se apoyará la implementación de tecnologías de la información en las empresas". Lo anterior se enfatiza en los objetivos establecidos para fomentar la innovación y economía del conocimiento. Entre los que destaca el "impulsar la industria de tecnologías de la información y comunicación".

En cuanto a los problemas existentes en el área de ingeniería de software en el ámbito regional-local, éstos no difieren substancialmente con respecto a los presentados en la investigación





sobre las prácticas de ingeniería de software en México de Villalobos y Gutiérrez (2001). Por su parte, la Universidad Autónoma de Yucatán contribuye con la formación de capital humano para las compañías de software de nuestro estado y del sureste, que producen software para sistemas de información, área con mayor demanda de las empresas de desarrollo de software en nuestro país.

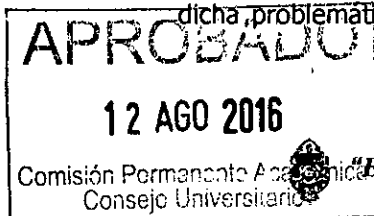
Para elevar y extender la competitividad del país mediante la promoción del uso y aprovechamiento de la tecnología y de la información, la difusión de la cultura tecnológica local y extranjera, el impulso al desarrollo de la industria de tecnologías de la información, así como el fomento y propagación de la industria de desarrollo de software, la Facultad de Matemáticas inició en 2004 la oferta de la Licenciatura en Ingeniería de Software con el propósito de coadyuvar al desarrollo regional y nacional antes citado. Con lo que se pretende generar recurso humano altamente capacitado en la solución de problemas, mediante el desarrollo de sistemas de software de calidad; y con profesionistas capaces de continuar sus estudios para realizar investigación dentro de la Ingeniería de Software; aspectos que marcan una gran diferencia a la sola profesionalización, misma que resulta de interés para muchos industriales de software. Es importante señalar que el plan de estudios de la Licenciatura en Ingeniería de Software ha considerado como guías para definir su estructura básica las tendencias de educación superior nacionales junto con las recomendaciones del modelo educativo de la Universidad.

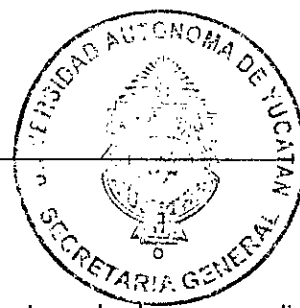
También, es relevante señalar que los resultados de la encuesta aplicada a distintos empleadores locales, como parte del Estudio de Mercado Laboral realizado a finales de 2013, reflejan un panorama alentador; en general, los empresarios están satisfechos con las habilidades y actitudes demostradas por los egresados. El mercado laboral de la Licenciatura en Ingeniería de Software se encuentra en crecimiento, ya que las empresas en el estado y en la región están modernizándose y requieren de personal capacitado para usar nuevos equipos o nueva tecnología. La Facultad de Matemáticas está bien posicionada entre los empleadores e informantes clave y consideran que los jóvenes egresados poseen una buena preparación. No obstante, algunos empresarios no están debidamente informados acerca de la existencia de la licenciatura ni sobre las necesidades que pudieran satisfacer en su negocio al contratar a un Ingeniero de Software.

Por consiguiente, la información presentada en este apartado constituye un insumo importante para la planeación, desarrollo y pertinencia de la oferta educativa de la UADY en todos sus niveles y modalidades, así como para la identificación de áreas en las cuales puede participar a través de la realización de proyectos de investigación que resulten de interés para las partes.

2.2.2 Referente disciplinar

La Ingeniería de Software (IS) como disciplina tiene pocas décadas de existencia; de hecho, el término "Ingeniería de Software" fue propuesto hace poco menos de medio siglo por Friedrich L. Bauer en una reunión del Comité de Ciencia de la Organización del Tratado del Atlántico Norte (OTAN) celebrada en otoño de 1967. En aquella época se discutía la creciente complejidad que la Programación presentaba al tratar de solucionar problemas cada vez más complejos (Wirth, 2008); dicha problemática, bautizada como la "Crisis del Software", dio origen a un par de conferencias





auspiciadas por la OTAN entre 1968 y 1969, en las que se sentaron las bases para diversos aspectos vinculados con la nueva disciplina (Naur y Randell, 1969; Buxton y Randell, 1970).

En el ámbito educativo, el desarrollo de la disciplina comenzó en 1978 con programas de posgrado en los Estados Unidos de Norteamérica; el primer programa denominado Master en Ingeniería de Software (MIS) surge en la Universidad Cristiana de Texas (Gibbs & Fairley, 1987). En el caso de programas en el nivel de licenciatura, Inglaterra fue el primer país en adoptar programas en el área de la Ingeniería de Software; Lehman (1987) describe las estructuras de los programas de nivel licenciatura en Inglaterra y cita al Imperial College como la primera Institución en ofrecer en 1987 un programa de licenciatura; así mismo, Lutz y Naveda (1997) reportan que el primer programa de licenciatura en los Estados Unidos de Norteamérica, fue implementado en 1996 por el Rochester Institute of Technology. En México, los programas curriculares en el área de Ingeniería de Software no vieron la luz en el siglo pasado; incidentemente es 2004 el año en el que se comienzan a ofrecer tanto el primer programa de maestría en el Centro de Investigación en Matemáticas Aplicadas (CIMAT), y en el caso de la Licenciatura, en la Universidad Autónoma de Yucatán (UADY).

El rápido desarrollo de la Ingeniería de Software en el ámbito académico recibió fuerte influencia por entidades como el Software Engineering Institute (SEI, por sus siglas en inglés) de la Universidad de Carnegie y Mellon; su aportación consistió en un Modelo de Currículo para un Master en Ingeniería de Software (Ardis y Ford, 1989) sustentado en un conjunto de reportes publicados por el SEI con los que se argumenta la profesión del Ingeniero de Software; así mismo, presenta el análisis de quince programas de posgrado con reconocimiento internacional y la descripción detallada de seis cursos básicos.

Otro de los organismos importantes para el desarrollo de la disciplina, ha sido el Institute of Electrical and Electronics Engineers (IEEE, por sus siglas en Inglés), dicho Instituto organizó en 1973, el primer congreso internacional en dicha disciplina, y dos años más tarde inició su revista titulada "IEEE Transactions on Software Engineering", actualmente referente internacional en temas de IS. El Glosario de Términos de Ingeniería de Software define a la Ingeniería de Software como:

(1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; esto es la aplicación de la ingeniería al Software. (2) el estudio de enfoques como en (1). (IEEE, 1990)

Otra referencia importante es la Association for Computing Machinery (ACM, por sus siglas en inglés) quién junto con la IEEE publicó en 2004 la guía SWEBOK, documento que presenta un esquema de conocimientos y competencias que todo ingeniero de software debe conocer por ser relevante para su actividad profesional (IEEE-CS, 2004). A 10 años de su publicación, Bourque y Fairley (2014) presentan —bajo el auspicio de la IEEE Computer Society— una versión revisada y actualizada del cuerpo de conocimiento denominada SWEBOK V3.0, el cual se encuentra estructurado en 15 áreas de conocimiento (AC). La Tabla 1 resume las Áreas de Conocimiento, así como sus principales temas o saberes; en la tercera columna de dicha Tabla se identifican asignaturas existentes (E) del Plan de Estudios 2009 en las que se consideran dichos temas, así como algunas asignaturas propuestas (P) para aquellos temas que no son cubiertos.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Tabla 1 Áreas de Conocimiento del SWEBOK con sus temas o saberes.

Área de Conocimiento	Temas o saberes	Asignaturas Plan LIS 2009
1. Requisitos de Software	Fundamentos de Requisitos de Software; Proceso de Requisito; Obtención de Requisitos; Análisis de Requisitos; Especificación de Requisitos, Validación de Requisitos; Gestión de Requisitos; Herramientas Informáticas de Apoyo.	Requisitos de Software (E)
2. Diseño de Software	Fundamentos de Diseño de Software; Cuestiones Relevantes del Diseño de Software; Estructura y Arquitectura de Software; Diseño de Interfaces Usuario; Análisis y Evaluación de la Calidad en el Diseño de Software; Notaciones en el Diseño de SW; Estrategias y Métodos en el Diseño de Software; Herramientas Informáticas de Apoyo.	Diseño de Software (E) Arquitecturas de Software (E) Interacción Humano Computadora (E)
3. Construcción de Software	Fundamentos en la Construcción de Software; Gestión de la Construcción; Consideraciones Prácticas; Tecnologías para la Construcción; Herramientas Informáticas de Apoyo.	Construcción y Evolución de Software (E) Programación en la Web Arquitecturas de Software (E)
4. Pruebas de Software	Fundamentos de Pruebas de Software; Niveles de las Pruebas de SW; Técnicas de Pruebas; Mediciones Relacionadas con las Pruebas; Proceso de Pruebas; Herramientas Informáticas de Apoyo.	Verificación y Validación de Software (P)
5. Mantenimiento de Software	Fundamentos del Mantenimiento de Software; Cuestiones Relevantes del Mantenimiento de Software; Proceso de Mantenimiento; Técnicas de Mantenimiento; Herramientas Informáticas de Apoyo.	Mantenimiento de Software (P)
6. Gestión de la Configuración del Software	Gestión del Proceso de Configuración del Software; Identificación de la Configuración del Software; Contabilidad del estado de la Configuración de Software; Auditoría de la Configuración de Software; Entrega y Gestión de la Liberación de Software; Herramientas Informáticas de Apoyo.	Administración de Proyectos I (E) Administración de Proyectos II (E)
7. Gestión de Proyectos de Software	Inicialización y Definición del Alcance; Planificación de Proyectos de Software; Formulación del Proyecto de Software;	Administración de Proyectos I (E)

APR 12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Área de Conocimiento	Temas o saberes	Asignaturas Plan LIS 2009
	Evaluación y Revisión; Clausura; Medición; Herramientas Informáticas de Apoyo.	Administración de Proyectos II (E)
8. Proceso de Ingeniería de Software	Definición del Proceso de Software; Ciclos de Vida del Software; Mejora y Evaluación del Proceso de Software; Medición del Software; Herramientas Informáticas de Apoyo.	Fundamentos de Ingeniería de Software (E) Métricas de Software (E)
9. Modelos y Métodos de la IS	Modelado; Tipo de Modelos; Análisis de Modelos; Métodos en Ingeniería de Software.	Métodos Formales en la Especificación y Diseño de Software (E)
10. Calidad de Software	Fundamentos de Calidad de Software; Procesos para Gestionar la Calidad de Software; Consideraciones Prácticas; Herramientas Informáticas de Apoyo.	Aseguramiento de la Calidad del Software (E)
11. Práctica Profesional de la IS	Profesionalismo; Dinámica de Grupo y Psicología; Habilidades de Comunicación.	Fundamentos de Ingeniería de Software (E)
12. Economía de la IS	Fundamentos sobre Economía en Ingeniería de Software; Ciclo de Vida Económico; Riesgo e Incertidumbre; Análisis de Métodos Económicos; Consideraciones Prácticas.	Evaluación de Proyectos (E)
13. Fundamentos de Computación	Técnicas para la Resolución de Problemas; Fundamentos de Programación; Abstracción; Estructura y Representación de Datos; Organización de Computadoras Fundamentos de Sistemas Operativos Fundamentos de Lenguajes de Programación Fundamentos de Comunicación en Redes Computación Paralela y Distribuida Fundamentos de Bases de Datos y Gestión de Datos Herramientas y Técnicas de Depuración; Algoritmos y Complejidad; Conceptos Elementales de un Sistema; Fundamentos sobre Factores Humanos del Usuario; Fundamentos sobre Factores Humanos del Desarrollador; Desarrollo y Mantenimiento de Software Seguro. Fundamentos de Compiladores	Fundamentos de Programación (E) Programación (E) Estructuras de Datos (E) Teoría de la Computación (E) Arquitectura de Computadoras (E) Sistemas Operativos (E) Teoría de Lenguajes de Programación (E) Redes y Seguridad de Computadoras (E) Cómputo Distribuido y paralelo (E) Bases de datos (E)
14. Fundamentos Matemáticos	Conjuntos, Relaciones y Funciones; Lógica Elemental, Estructuras Algebraicas. Grafos y Árboles.	Álgebra Intermedia (E) Álgebra Superior I (E) Álgebra Lineal (E)



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





Área de Conocimiento	Temas o saberes	Asignaturas Plan LIS 2009
	Máquinas de Estado Finito; Gramáticas Probabilidades Discretas Técnicas de Demostración; Fundamentos de Conteo; Precisión Numérica, Exactitud y Errores; Teoría de Números;	Probabilidad (E) Geometría Analítica (P) Cálculo Diferencial (E) Cálculo Integral (E) Matemáticas Discretas (E)
15. Fundamentos de Ingeniería	Métodos Empíricos y Técnicas Experimentales; Análisis Estadístico; Medición; Diseño Ingenieril; Modelado, Simulación y Prototipos; Estándares; Análisis de Causa Raíz	Diseño de Experimentos en IS (E) Inferencia Estadística (E)

En el ámbito nacional, la Asociación Nacional de Instituciones de Educación en Informática (ANIEI) constituida en octubre de 1982, y actualmente denominada Asociación Nacional de Instituciones de Educación en Tecnologías de la Información, ante la ausencia de un núcleo básico de conocimientos para un profesionista en Computación, comenzó a trabajar desde 1983 en un conjunto de Modelos Curriculares del Nivel Superior de Informática y Computación (García et al, 2015). La propuesta presenta cuatro perfiles profesionales en Informática y Computación, un conjunto de áreas de conocimientos en estos campos del saber, y una matriz de ponderaciones porcentuales de las áreas de conocimiento con base en la cual se configura cada uno de los cuatro perfiles: El segundo perfil, conocido inicialmente como Licenciado en Sistemas Computacionales, adoptó en la cuarta edición (2006) el nombre de Licenciatura en Ingeniería de Software. La Tabla 2 presenta las ocho Áreas de Conocimiento definidas por ANIEI, sus temas o saberes, y el porcentaje acorde con el perfil del Ingeniero de Software.

Tabla 2. Áreas de Conocimiento y temas o saberes de la ANIEI.

Áreas de Conocimiento de la ANIEI	Temas o saberes	Perfil del IS (%)
Entorno Social	-Problemática Global, Nacional y Regional -Las Organizaciones -Las Unidades de Informática -Ética y Normatividad	12.5
Matemáticas	-Matemáticas Básicas -Matemáticas Aplicadas -Matemáticas Discretas -Teoría Matemática de la Computación	12.5
Arquitectura de Computadoras	-Física -Sistemas Digitales -Tipos y Configuraciones de Computadoras -Instalaciones y Equipos -Nuevos Paradigmas Tecnológicos	7.5

APROBADO
12 AGO 2016

Áreas de Conocimiento de la ANIEI	Temas o saberes	Perfil del IS (%)
Redes	-Transmisión y Comunicación de Información -Modelos -Protocolos -Intercomunicación de Redes -Seguridad e Integridad de Información	7.5
Software de Base	-Traductores -Sistemas Operativos -Utilerías y Manejadores -Alto Rendimiento	7.5
Programación e Ingeniería de Software	-Algorítmica -Paradigmas de Programación y Lenguajes -Desarrollo de software -Industria del Software	22.5
Tratamiento de Información	-Bases de Datos -Recuperación de la Información -Sistemas de Información	20
Interacción Humano-Máquina	-Interfaces Humano-Máquina -Graficación -Sistemas Inteligentes	10

Cabe mencionar que los Modelos Curriculares de la ANIEI son considerados por el Consejo Nacional de Acreditación en Informática y Computación (CONAIC) en su Marco de Referencia definido para el proceso de acreditación de programas académicos en el nivel de Licenciatura (Ver Tabla 3).



APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Tabla 3. Ponderación por AC para el perfil profesional de Licenciatura en IS.

Área de Conocimiento Consideradas por el CONAIC		Unidades por AC (CONAIC)	Unidades del Plan LIS-2009 (UADY)
Informática (2009)	Entorno Social	236	321
	Arquitectura de Computadoras	88	65
	Redes	118	130
	Software de Base	88	100
	Programación e Ingeniería de Software	206	555
	Tratamiento de Información	176	205
	Interacción Hombre-Máquina	88	70
Ciencias Sociales, Humanidades y Otras		300	321
Matemáticas y Ciencias Básicas		420	830
Total		1,720	2,587

Otra de las instituciones nacionales que actualmente tiene impacto en la formación de jóvenes en disciplina de la Ingeniería de Software, es el Centro Nacional de Evaluación para la Educación Superior (CENEVAL); su actividad principal ha sido el diseño y aplicación de instrumentos estandarizados de evaluación para los procesos de enseñanza-aprendizaje; uno de los instrumentos de los que dispone, es el Examen General de Egreso de Licenciatura (EGEL), dicho instrumento representa una prueba de cobertura nacional que evalúa el nivel de conocimientos y habilidades académicas de los recién egresados de una carrera específica y carreras afines; el caso de la Ingeniería de Software, el CENEVAL dispone del EGEL-ISOFT, su estructura vigente —aprobada por el Consejo Técnico en junio de 2012— incluye 11 subtemas para las cuatro áreas de conocimiento actualmente consideradas en la prueba.

La Tabla 4 ilustra la cobertura del plan de estudios de la Licenciatura en Ingeniería de Software (planes 2004 y 2009).

El plan de estudios ofrece una cobertura del 61.76% del Examen General de Egreso para Ingeniería de Software; y el 38.24% restante corresponde a las asignaturas disciplinarias del área de Matemáticas (20.6%), de las Ciencias Computacionales (11.76%), y otras que ofrecen competencias para el área de la Investigación (5.88%).

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



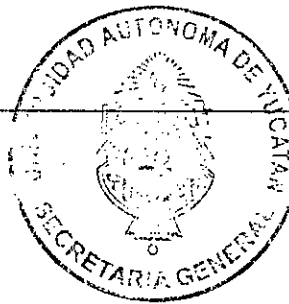


Tabla 4. Cobertura del Plan de Estudios (2004 y 2009) de LIS-UADY para el EGEL-ISOFI.

Área de conocimiento	% en el examen	Asignaturas del Plan de Estudios	% asignaturas obligatorias
A. Análisis de sistemas de Información. A1. Diagnóstico del problema y valoración de la factibilidad para el desarrollo de sistemas de información. A2. Modelado de los requerimientos de un sistema de información.	13.26	Gestión Tecnológica, Desarrollo de Requisitos de Software.	5.88
B. Desarrollo e implantación de aplicaciones computacionales. B1. Diseño de la solución del problema de tecnología de información B2. Desarrollo de sistemas B3. Implantación de sistemas B4. Aplicación de modelos matemáticos	40.88	Fundamentos de Programación, Programación, Estructuras de Datos, Matemáticas Discretas, Fundamentos de Ingeniería de Software, Arquitecturas de Software, Diseño de Software, Construcción y Evolución de Software, Programación en la Web, Métodos Formales en Especificación y Diseño de Software.	29.41
C. Gestión de proyectos de tecnologías de información. C1. Administración de proyectos de tecnologías de la información. C2. Control de calidad de proyectos de tecnologías de la información.	14.37	Evaluación de Proyectos, Administración de Proyectos I, Administración de Proyectos II, Aseguramiento de la Calidad del Software, Métricas de Software.	14.71
D. Implementación de redes, bases de datos, sistemas operativos y lenguajes de desarrollo. D1. Gestión de redes de datos D2. Gestión de base de datos D3. Gestión de sistemas operativos o lenguajes de desarrollo	31.49	Redes y Seguridad de Computadoras, Bases de Datos, Sistemas Operativos, Sistemas Distribuidos.	11.76
Total	100		61.76

APROBADO

12 AGO 2016

Comisión Permanente Académica
 Consejo Universitario



De acuerdo a Barry Boehm (2006), figura prominente en la Ingeniería de Software, las tendencias actuales de la disciplina se centran en:

- 1) el desarrollo rápido de aplicaciones siguiendo enfoques de desarrollo ágiles,
- 2) la IS basada en el valor donde se prioriza las necesidades del cliente,
- 3) aseguramiento de la calidad en sistemas software críticos,
- 4) desarrollo dirigido a partir de modelos,
- 5) mantenimiento y evolución de software legado así como desarrollado basado en código abierto,
- 6) interacción entre la ingeniería de software y la ingeniería de sistemas,
- 7) desarrollo de software a nivel global, e
- 8) integración de sistemas de sistemas.

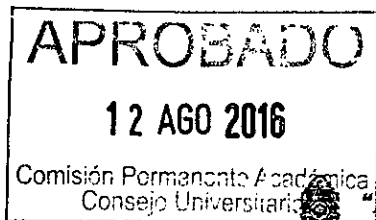
Cabe mencionar que en el documento Estado del Arte, elaborado como parte de la propuesta de modificación, se incluyó un análisis comparativo de planes de estudio de Ingeniería de Software, en el que se identificaron las similitudes y diferencias en cuanto al apego a los modelos curriculares nacionales e internacionales.

2.2.3 Referente profesional

En el ámbito internacional, el ingeniero de software desarrolla su trabajo de acuerdo con las prácticas, estándares y guías generalmente aceptadas, y establecidas por las sociedades profesionales, como la Association for Computing Machinery (ACM) y la IEEE Computer Society (IEEE CS).

En este contexto, la IEEE Computer Society (IEEE CS) a través del proyecto SWEBOK ofrece una guía del cuerpo de conocimientos generalmente aceptado en el ámbito de la Ingeniería de Software el cual provee fundamentos para el desarrollo curricular y para la certificación individual de los profesionistas en esta disciplina. En términos generales, el SWEBOK establece que la formación del ingeniero de software se alcanza mediante la integración de un conjunto de áreas de conocimiento en torno a los procesos de desarrollo de software:

- 1) requisitos de software,
- 2) diseño de software,
- 3) construcción de software,
- 4) pruebas de software,
- 5) mantenimiento de software; así como un conjunto de áreas de conocimiento en torno a los procesos de gestión:
- 6) gestión de la calidad,
- 7) gestión de la configuración,
- 8) gestión del desarrollo de proyectos de software.



En el ámbito nacional, la Asociación Nacional de Instituciones de Educación en Tecnologías de Información (ANIEI), comenzó a trabajar desde 1983 en un conjunto de Modelos Curriculares del Nivel Superior de Informática y Computación:

- 1) Licenciatura en Informática,
- 2) Licenciatura en Ingeniería de Software,
- 3) Licenciatura en Ciencias de la Computación, e
- 4) Ingeniería Computacional.

En esta propuesta se define al profesionalista del perfil 2) como un “profesional especialista en la producción de sistemas de software de calidad para la solución de diversas problemáticas del entorno. Es responsable de la formulación, planeación, implantación y mantenimiento de sistemas de información que garanticen la disponibilidad de altos niveles de servicio” (ANIEI, 2014, p. 6).

La ANIEI, con el propósito de alinear las capacidades de capital humano a través de la vinculación de los requerimientos de la industria de tecnologías de información con la oferta académica de las instituciones educativas y los programas de apoyo gubernamentales, de manera conjunta con Impulsa-TI y NYCE, promueven desde 2010 un modelo paracurricular que brinda un esquema de certificación en los siguientes perfiles:

- 1) Emprendedor de negocios de software,
- 2) Administrador de proyectos y procesos de software,
- 3) Arquitecto de software,
- 4) Ingeniero de software, y
- 5) Desarrollador de software.

En particular, el perfil de ingeniero de software establece, para el desempeño de su práctica profesional, un conjunto de competencias genéricas, entre las que se encuentran: análisis y síntesis de información, capacidad crítica, comunicación oral y escrita, control de cambios, solución de problemas, autodidacta, redacción de informes y trabajo en equipo; así mismo, establece un conjunto de competencias específicas, entre las que podemos destacar: manejo de metodologías y estándares para el desarrollo de software, conocimiento y codificación en lenguajes de programación, análisis de requisitos de software, especificación de diseño de software, diseño de interfaces de software, pruebas de software, seguridad del software, bases de datos, mantenimiento de software, transferencia de conocimiento y documentación de software.

En el Estudio de Seguimiento de Egresados se destaca que la totalidad de los entrevistados está satisfecho con la decisión de haber estudiado la Licenciatura en Ingeniería de Software, la razón significativa o la más mencionada se relaciona con la preparación y los conocimientos recibidos para desarrollarse en el mundo laboral.

2.2.4 Referente institucional

La UADY, en el Plan de Desarrollo Institucional 2014-2022, establece como su Misión: “la formación integral y humanista de personas, con carácter profesional y científico, en un marco de



APROBADO

12 AGO 2016

Comisión Permanente Académica

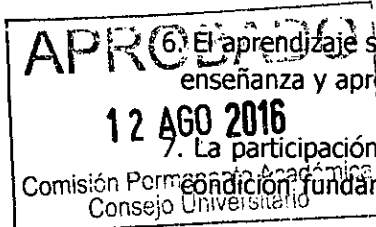




apertura a todos los campos del conocimiento y a todos los sectores de la sociedad. Como tal, proporciona un espacio de análisis y reflexión crítica sobre los problemas mundiales, nacionales y regionales, conduciendo al desarrollo sustentable de la sociedad, apoyándose en la generación y aplicación del conocimiento, en los valores universales y en el rescate y preservación de la cultura nacional y local dando respuesta de esta manera a la nueva era del conocimiento en su papel como transformadora de su comunidad. Como institución, incorpora cuatro principios básicos de la educación: aprender a conocer, aprender a hacer, aprender a ser y aprender a vivir y a convivir. Esta perspectiva sirve de punto de partida para el desarrollo e implementación de acciones que contribuyan al logro de la Misión en alineación con la Visión Institucional, la cual declara que —En el año 2020 la UADY es reconocida como la institución de educación superior en México con el más alto nivel de relevancia y trascendencia social”.

Esta actualización de la Visión Institucional, proyectada al 2022, sirve de base para la formulación del PDI. En él se establecieron objetivos, políticas y estrategias que la Universidad acordó impulsar durante esta década y en dirección a las cinco líneas de trabajo consideradas fundamentales para el desarrollo institucional: formación integral de los alumnos, desarrollo de programas académicos, organización y desarrollo de los académicos, servicios de apoyo al desarrollo académico y planeación, gestión y evaluación institucional. La UADY, en su filosofía, declara como principios fundamentales que sustentan su tarea educativa los siguientes:

1. La educación será fundamentalmente humanística, enfocada a la razón (crítica), a la voluntad (valores) y a la vida, ya que debe ser un espacio fundamental que ayude a formar ciudadanos y profesionales como miembros de su comunidad para que actúen de una manera responsable.
2. La educación es el desarrollo del individuo como persona, bajo la acción consciente e inteligente de su voluntad, reconociendo las diferencias individuales.
3. Educar no es aumentar desde fuera, sino propiciar que la persona crezca desde adentro. En el proceso educativo el agente principal es el alumno. Sin embargo, el maestro también es un agente cuyo dinamismo, ejemplo y dirección son fundamentales.
4. El interés por la totalidad del ser humano—congruencia entre su pensamiento, emoción y conducta— centrandolo en el alumno mismo como sujeto de su propia educación, creando las condiciones adecuadas para que esto pueda suceder.
5. El reconocimiento de que los estudiantes son seres humanos que tienen una naturaleza constructiva y digna de confianza.



6. El aprendizaje se facilita cuando el estudiante participa responsablemente en el proceso de enseñanza y aprendizaje, asignando a la enseñanza el papel estimulador.

7. La participación activa y responsable de todos los estudiantes en su proceso formativo es condición fundamental para fortalecer su capacidad de pensamiento crítico y de reflexión





acerca de sus sentimientos, valores, convicciones y futuras acciones como profesionales regidos por principios éticos.

8. El desarrollo de hábitos mentales y competencias que signifiquen estrategias para la realización humana y profesional.
9. El diálogo respetuoso en la relación maestro –alumno; guiar y proponer con razones el desarrollo responsable de la libertad.

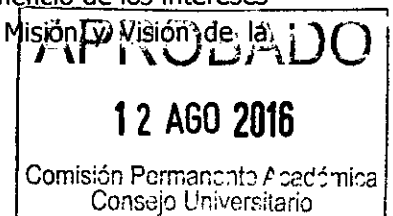
Para la UADY, el Modelo Educativo para la Formación Integral (MEFI) es su propuesta para promover la Formación Integral del estudiantado bajo una filosofía humanista. Esta propuesta se deriva de la necesidad de actualizar el Modelo Educativo y Académico (MEyA) (MEyA, 2002) después de un análisis de los resultados obtenidos, con el fin de producir un cambio en la UADY y en sus relaciones con la sociedad de tal manera que impacte en las funciones sustantivas, centradas en los actores que intervienen en la práctica educativa: el estudiante, el profesor, los directivos, administrativos y manuales.

La UADY, a través del MEFI, concibe la Formación Integral como un proceso continuo que busca el desarrollo del estudiante y su crecimiento personal en las cinco dimensiones que lo integran como ser humano: física, emocional, cognitiva, social y valoral-actitudinal. Esta formación integral del estudiantado se promueve en el MEFI por medio de la interacción de sus seis ejes de manera transversal en todos los Programas Educativos de la Universidad: responsabilidad social, flexibilidad, innovación, internacionalización, educación centrada en el aprendizaje y educación basada en competencias; los cuales orientan a su vez el trabajo académico y administrativo de la misma.

Los seis ejes del MEFI, además de su carácter transversal, tienen implicaciones en el diseño y elaboración de los planes y programas de estudio; el proceso de enseñanza y aprendizaje y la evaluación. De la misma manera, ejercen una influencia importante en los roles de los diversos actores: estudiante, profesor, personal administrativo, directivo y manual.

La Universidad ha establecido 22 competencias genéricas (MEFI, 2013) que deberán ser integradas en todos los PE de la UADY con el fin de asegurar que todos sus estudiantes desarrollen dichas competencias; su desarrollo se da de manera transversal en las asignaturas que integran los planes de estudio.

Además, el MEFI declara que en todos los planes de estudio se integrarán dos asignaturas institucionales obligatorias: Cultura Maya y Responsabilidad Social Universitaria (RSU). Esta inclusión tiene como objetivo la revaloración de las culturas originarias por parte del estudiantado y además, busca orientar hacia una opción ético-política de contribución al desarrollo humano y sustentable, la equidad, la inclusión social, los derechos humanos y la cultura de la paz así como la formación de recursos humanos capaces de transformar la sociedad en la que viven en beneficio de los intereses colectivos. Lo anterior establece las condiciones para dar respuesta a la Misión y Visión de la Universidad y contribuye a la formación de los futuros egresados.



2.3. Justificación de la pertinencia social y factibilidad del programa

En junio de 2013 se integró el comité para la modificación del plan de estudios de la Licenciatura en Ingeniería de Software, conformado por cuatro profesores de la Facultad de Matemáticas vinculados con este Programa Educativo, con la meta de realizar una propuesta de modificación y ponerla a consideración del H. Consejo Universitario de la UADY, la cual atendiera a las necesidades de pertinencia y vigencia del plan de estudios actual. La evaluación del plan de estudios se realizó de acuerdo al Modelo Educativo para la Formación Integral (MEFI, 2013), se incorporaron a esta modificación las bases metodológicas establecidas en dicho Modelo, así como los elementos orientadores de la nueva Guía para el Diseño y Elaboración de los Planes y Programas de Estudio que forma parte del Programa Institucional de Habilitación en el MEFI (PIH – MEFI) de la UADY.

Dentro de este proyecto, se realizaron los documentos de Pertinencia Social (EPS-LIS, 2015) y Factibilidad (FA-LIS, 2015) de acuerdo con el documento PIH-MEFI de la Dirección General de Desarrollo Académico.

2.4. Pertinencia social

Hoy en día los sistemas de software son ubicuos, es decir nos permiten estar conectados a la red en todo momento. La mayoría de los equipos electrónicos modernos incluyen algún tipo de software. Las aplicaciones de software se utilizan para ayudar al funcionamiento de la industria manufacturera, a las escuelas y universidades, al cuidado de la salud, a las finanzas y al gobierno; mucha gente utiliza software de diferente clase para el entretenimiento y la educación. La especificación, el desarrollo, la administración y la evolución de estos sistemas de software integran la disciplina de ingeniería de software. Incluso los sistemas de software más simples tienen una alta complejidad inherente. Por lo tanto, la ingeniería de software es una disciplina en la que los ingenieros de software aplican los métodos y la teoría de las ciencias de la computación de forma efectiva para resolver problemas complejos. Lo anterior se ve evidenciado en la gran cantidad de software moderno que proporciona buen servicio a sus usuarios (Sommerville, 2011).

El ingeniero de software es un profesional capaz de dirigir equipos de especialistas en desarrollo de software para la creación, implantación y administración de grandes sistemas de cómputo. También proveerá a las empresas soluciones a problemas relacionadas con el desarrollo de software, a través de los conocimientos de las tecnologías de usabilidad y lenguajes de programación adquiridos mediante el desarrollo de habilidades en aspectos teóricos, prácticos y metodológicos, para resolver los problemas de sistematización del cliente.

Desde una perspectiva económica, el software constituye un medio de producción inherente del capital intelectual de las empresas (Baetjer, 1998). Y es exactamente este capital el que debe desarrollarse en las universidades.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





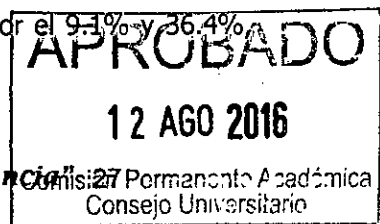
Entre los puestos y funciones que se vislumbran como emergentes y/o novedosos existe una gama muy amplia, en donde la ingeniería de software es una parte esencial. De acuerdo con Microsoft (2014), algunos de los puestos emergentes con mayores áreas de oportunidad son para: desarrolladores, administradores de programa, ingenieros de pruebas, ingenieros constructores, profesionales de publicación de contenido, diseñadores de juegos, planificadores de productos y especialistas en experiencia del usuario.

Como puede apreciarse, la profesión de ingeniero de software engloba un conjunto de disciplinas encargadas del diseño, desarrollo y entrega de productos de software. A los ingenieros de software les motiva facilitar la vida y el trabajo de los demás, así como ayudar a alcanzar nuevas metas a través de tecnologías en constante avance.

De acuerdo con los datos socioeconómicos y expectativas de las familias de aspirantes a la Universidad Autónoma de Yucatán (UADY), mismos que se incluyen en el estudio Socioeconómico y de Expectativas Educativas, se destaca que la UADY implementó un estudio de perfiles de estudiantes y trayectorias escolares, de manera metodológica y sistemática, con los estudiantes que ingresaron en el período 2010-2011. Como resultado de este estudio se obtuvo que entre los estudiantes de nuevo ingreso, el 47.2% son mujeres y el 52.8% son hombres. En cuanto al nivel de estudios de los padres de familia, el 56.8% de los padres y el 71.6% de las madres no poseen formación de nivel superior. Los estudiantes de nuevo ingreso se clasificaron en tres niveles socioeconómicos: el 34.9% son de nivel bajo, 43.1% de nivel medio y el 22% de nivel alto, es decir son en su mayoría de clase media. Entre los factores que pueden dificultar la permanencia en la universidad, el estudio arrojó que el 50.7% puede ser por el factor económico, 29.3% por el factor académico y 20% por otros factores.

La distribución de los estudiantes de nuevo ingreso a la Facultad de Matemáticas en el 2012 es la siguiente: 71.1% hombres y 28.9% mujeres; en el caso de la Licenciatura en Ingeniería de Software, 86.6% son hombres y 11.4% son mujeres. En otros resultados de este estudio, se tiene que el 20% de los estudiantes de nuevo ingreso a la Facultad de Matemáticas indicó que tiene un empleo. Entre las razones por las que trabajan predominan: pagarse los estudios, tener independencia económica y ayudar en el gasto familiar. Los jóvenes que ingresaron a la licenciatura en estudio, carecen de referentes culturales familiares necesarios para enfrentar sus estudios universitarios debido a que el 72.1% de los padres y el 58.1.2% de las madres de estos estudiantes no tuvieron oportunidad de cursar estudios universitarios.

Los estudiantes que ingresaron a la Licenciatura en Ingeniería de Software cuentan en su mayoría con espacio en el hogar para estudiar. El 100% de los estudiantes indica tener computadora, el 68.2% tiene impresora y el 86.6% tiene acceso a internet en el hogar. Los estudiantes de nuevo ingreso se clasificaron en tres niveles socioeconómicos: bajo, medio y alto. Las distribución de los niveles de todos los estudiantes que ingresaron a la facultad es 37.2% pertenecen al nivel bajo, 47.1% pertenece al nivel medio y el 15.75 pertenece al nivel alto. En el caso la Licenciatura en Ingeniería de Software hay un aumento de los estudiantes que pertenecen al nivel medio representando por 54.5%, los niveles bajo y alto tienen una representado por el 9.1% y 36.4% respectivamente.





Los factores valorados para la elección de carrera en la Facultad fueron, en orden de prioridad: posibilidad de empleo, opiniones de padres, opiniones de profesores, orientación vocacional, opiniones de amigos y opiniones de familiares. El 97.7% de los estudiantes de Ingeniería de Software considera posibilidades medias o altas de conseguir un empleo cuando termine sus estudios siendo la empresa privada el espacio laboral en el que la mayoría pretende desempeñarse, aproximadamente la cuarta parte pretende hacerlo en el sector público.

En cuanto a las necesidades que pueden ser atendidas por los ingenieros de software en el ámbito local, conviene mencionar también los resultados obtenidos en el Estudio de Mercado Laboral realizado por la Facultad de Contaduría y Administración de la UADY en 2013. El cual concentra información relevante de egresados y empleadores de la Licenciatura en Ingeniería de Software de la Facultad de Matemáticas.

El estudio indica que una las principales razones por la que los egresados decidieron estudiar la Licenciatura en Ingeniería de Software fue "porque el campo laboral es bastante amplio".

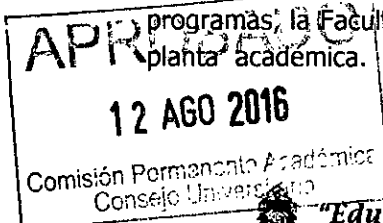
Del mismo modo, los empleadores están conscientes que actualmente todo gira en torno al desarrollo tecnológico, lo que trae como consecuencia el desarrollo de software y por consiguiente, la contratación de personal calificado que pueda solucionar estas demandas. Consideran que los profesionales en el área de ingeniería de software deberán tener conocimientos en áreas tales como: lenguajes de programación, plataformas de desarrollo de software actuales, programación para dispositivos móviles, gestión de proyectos, gestión de la calidad, contabilidad, aspectos legales y dominio del idioma inglés.

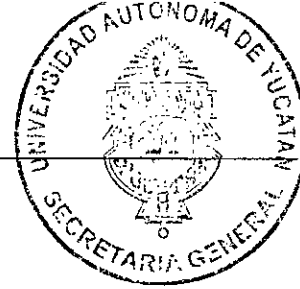
Con respecto a los factores que determinan la promoción laboral, los entrevistados no proporcionaron mayores detalles, salvo la expectativa de que los profesionistas tuvieran capacidad de aprendizaje y deseos de desarrollarse dentro de la compañía.

Por ende, un egresado de la Licenciatura en Ingeniería de Software puede elegir entre diversos espacios de trabajo inter y multidisciplinario. Entre ellos se encuentran la administración de proyectos, el desarrollo de software, el mantenimiento a programas con los que ya cuentan las empresas. En la capital del estado, el mercado laboral se está enfocando principalmente al área comercial cuyo objetivo es desarrollar software para la atención a sus clientes. En el ramo industrial se están contratando ingenieros de software para la implementación y operación de las máquinas.

2.5. Factibilidad

La Facultad de Matemáticas forma parte del Campus de Ciencias Exactas e Ingenierías (CCEI) e imparte seis programas curriculares en el nivel de licenciatura; tres de ellos se ubican en el área de las Tecnologías de la Información y Comunicaciones (TIC): las Licenciaturas en Ingeniería de Software, Ciencias de la Computación, Ingeniería en Computación. En la operación de estos programas, la Facultad de Matemáticas procura fortalecer, la infraestructura, los servicios en TIC y planta académica. En el aspecto académico, el número de profesores ha crecido de manera





importante desde el año 2004, así como también se ha elevado su grado de habilitación en el área de las TIC.

El trabajo académico de los profesores de la Facultad de Matemáticas está organizado en 10 Cuerpos Académicos (CA), y para la implementación de la Licenciatura en Ingeniería de Software, participan profesores de seis de dichos CA (ver Tabla 5).

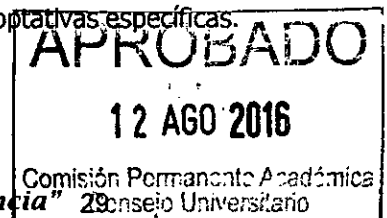
Es importante mencionar que los Cuerpos Académicos de Tecnologías para la Formación en Ingeniería de Software (CA-TFIS) y de Ingeniería de Software e Informática Educativa (CA-ISIE) que atienden el área disciplinar del programa educativo, se encuentran en proceso de formación y no cuentan con registro ante el PRODEP; así mismo, es recomendable que las próximas contrataciones sean de profesionistas con experiencia y posgrado en el área de Ingeniería de Software.

Tabla 5. Profesores de los CA que atienden la LIS.

Cuerpo Académico	Profesores	Maestría	Doctorado	PRODEP	SNI
Algebra	8	1	5	5	4
Tecnologías Emergentes de Computación	8	5	3	7	0
Tecnologías para la Formación en Ingeniería de Software	4	2	2	3	1
Ingeniería de Software e Informática Educativa	5	3	2	3	1
Estadística y Probabilidad	17	8	4	4	1
Modelado y Simulación Computacional de Sistemas Físicos	11	2	9	9	5

En relación con la atención de la carga docente, adicionalmente a los profesores que pertenecen a los cuerpos académicos, hay un conjunto de académicos que también participan en la impartición de las asignaturas del plan de estudios.

De manera adicional, el plan de estudios actual contempla asignaturas optativas, organizadas en áreas de concentración, las cuales son cubiertas por los Cuerpos Académicos vinculados con el área de Computación; en particular, las áreas que se ofrecen comúnmente son: Mejora de Procesos, Desarrollo de Aplicaciones Web, Videojuegos, Bases de Datos. No obstante, se reconoce que la vigencia de las áreas de concentración y de las asignaturas optativas asociadas a ellas, depende de la demanda de los estudiantes y la disponibilidad de los profesores. Para atenuar dicha debilidad, se ha recurrido a la contratación de profesores de asignatura para la impartición de optativas específicas.



Los profesores tienen esquemas de colaboración o convenios con otras instituciones u organizaciones nacionales e internacionales como:

- Universidad de Castillo-La Mancha España
- Universidad Politécnica de Madrid, España
- Pontificia Universidad Católica del Perú
- Red Temática de Ingeniería de Software
- Red Latinoamericana de Ingeniería de Software
- Universidad Autónoma de San Luis Potosí
- Universidad Autónoma de Baja California
- Universidad Autónoma del Carmen
- Universidad de la Fuerzas Armadas ESPE de Ecuador



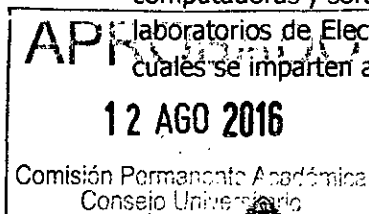
Además, profesores de los CA de TFIS y de ISIE participan como:

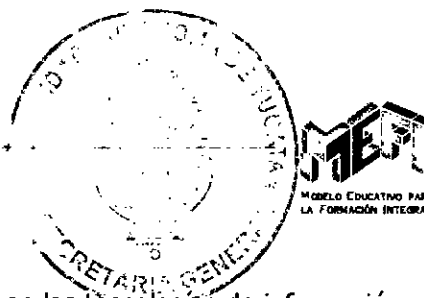
- Miembros de comité de la Feria de Ciencia y Tecnología
- Miembro del Comité Permanente de las Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento
- Miembro de la Mesa Directiva de la Asociación Nacional de Instituciones de Educación en Tecnologías de Información (ANIEI)
- Evaluador del Consejo Nacional de Acreditación en Informática y Computación (CONAIC).
- Miembro del Comité de Acreditación del Consejo Nacional de Acreditación en Informática y Computación (CONAIC).

La planta académica reportada ha permitido satisfacer las necesidades docentes para las asignaturas disciplinarias del área de Ingeniería de Software; no obstante, para atender las recomendaciones de las evaluaciones externas en torno a la vinculación con el sector productivo o de servicios, así como para diversificar la oferta de asignaturas optativas, es deseable considerar la contratación de personal especializado en la disciplina; así mismo, para fortalecer el desarrollo de las líneas de generación y aplicación del conocimiento de los cuerpos académicos vinculados con la disciplina, y con ello promover su consolidación, la cual incida en la formación de los estudiantes, es pertinente contratar personal con formación doctoral en Ingeniería de Software.

Infraestructura

La Licenciatura en Ingeniería de Software comparte los edificios de aulas C, D, F y H, así como las salas del Centro de Tecnologías de Información y Comunicaciones (CTIC), con los demás programas educativos de licenciatura y posgrado que se imparten en la Facultad. Actualmente se dispone de 23 aulas con equipo audiovisual y de 6 salas equipadas con equipo audiovisual, computadoras y software acorde a los requerimientos de las asignaturas. También, se dispone de los laboratorios de Electrónica y Circuitos Eléctricos, Robótica y Control, así como el de Redes, en los cuales se imparten asignaturas obligatorias y optativas.





Por la naturaleza de la licenciatura, los alumnos utilizan las tecnologías de información para consultar acervos bibliográficos y desarrollar proyectos; para realizar estas actividades, los estudiantes disponen de equipos de cómputo en el CTIC; así mismo, las instalaciones de los laboratorios pueden ser utilizados para prácticas y/o proyectos, en horarios establecidos.

Para la impartición de las asignaturas se dispone de sistemas de gestión de aprendizaje (Moodle), utilizadas como plataformas de apoyo para las clases presenciales.

Tabla 6. Infraestructura.

Espacio	Número
Aulas de Clase	21
Salas de Computo	7
Laboratorio de Redes	1

En términos generales, la Facultad proporciona la infraestructura de tecnología educativa y de información necesarias para que el plan de estudios se desarrolle de manera adecuada; sin embargo, debido a que la utilización de tecnología de la información es esencial para el desarrollo efectivo de la licenciatura, es necesario mantenerla actualizada.

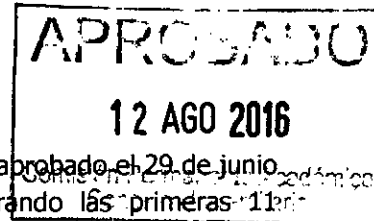
2.6 Evaluación interna y externa del programa

Para la evaluación interna se analizaron los datos históricos sobre la trayectoria escolar disponibles en el Departamento de Control Escolar, e información vinculada con los procesos de selección y titulación disponibles en la Secretaría Académica; así mismo, se administraron encuestas a alumnos y profesores respecto del plan de estudios vigente. En el caso de la evaluación externa se analizó la información obtenida de egresados y empleadores en el estudio de mercado laboral encargado al Centro de Desarrollo de Negocios (CEDENE) de la Facultad de Contaduría y Administración, opiniones de expertos en el área respecto del plan de estudios vigente, así como las recomendaciones hechas por el CONAIC como parte del proceso de acreditación, y los resultados obtenidos en el Examen General de Egreso (EGEL) del CENEVAL de 2008 a la fecha.

Cabe mencionar que el detalle de estos análisis se puede consultar en el Documento de Evaluación Curricular de la Licenciatura en Ingeniería de Software.

2.6.1 Evaluación interna

El Plan de Estudios de la Licenciatura en Ingeniería de Software fue aprobado el 29 de junio de 2004 y comenzó a operar en septiembre del mismo año. Considerando las primeras generaciones, han ingresado un total de 424 alumnos, de los cuales, el 8% ingresó mediante el proceso de revalidación; así mismo, sin contar a la primera generación, 866 alumnos han solicitado ingresar al programa, y en promedio, el 43% de la demanda ha sido atendida a través del proceso de selección. La Tabla 7 presenta la evolución histórica de la oferta y demanda atendida de 2004 a



2014, como se puede observar en dicha tabla, la demanda ha presentado una tendencia creciente a partir del ingreso en 2010, llegando a ser en 2014 el triple de la oferta atendida.

Tabla 7. Evolución histórica de la demanda y oferta de LIS.

	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
Demanda en Proceso de Selección (PS)	--	54	71	98	72	86	71	88	95	109	122
Oferta Atendida	17	31	32	35	40	34	51	38	39	40	35
Campos Vacantes	21	37	35	36	44	35	54	42	44	43	35

* Incluye la demanda atendida por revalidación

Con base en la información de la Tabla 8., el índice de deserción para las primeras tres generaciones (en las cuales se ha agotado el período de permanencia) es de aproximadamente el 60%. Por otro lado, del análisis a la trayectoria de los egresados, se obtuvo que en promedio les ha tomado diez semestres concluir sus estudios, y dos semestres adicionales para titularse.



Tabla 8. Evolución de la Matrícula.

Generación	Alumnos Inscr.	Evolución de la Retención											Deserción
		Primer Año	Segundo Año	Tercer Año	Cuarto Año	Quinto Año	Sexto Año	Séptimo Año	Octavo Año	E	T	A	
2004	21	16	12	9	1	1	-	-	-	7	7	0	14
2005	37	30	24	15	14	6	8	5	3	19	17	0	18
2006	35	24	18	19	15	3	2	-	-	12	11	0	23
2007	36	28	29	24	7	5	4	-	-	12	10	4	20
2008	44	40	35	22	11	5	2	-	-	18	17	7	19
2009	35	32	31	24	14	5	-	-	-	14	10	9	12
2010	54	47	36	33	24	-	-	-	-	3	1	28	23
2011	42	34	33	28	-	-	-	-	-	-	-	32	10
2012	44	39	34	-	-	-	-	-	-	-	-	38	6
2013	43	35	-	-	-	-	-	-	-	-	-	39	4
2014	35	-	-	-	-	-	-	-	-	-	-	35	-

E Egresados T Titulados A Activos

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

En relación con los indicadores de egreso y titulación, en las primeras seis generaciones han egresado 83 (40%) alumnos, de los cuales 81 (96%) ya se han titulado (ver Tabla 9).

Tabla 9. Indicadores de Egreso y Titulación

Año de Ingreso	Ingreso (#)	Egresados (#)	Índice de Egreso (%)	Titulados (#)	Índice de Titulación (%)
2004	21	7	33	7	100
2005	37	19	51	19	100
2006	35	12	34	12	100
2007	36	14	39	14	100
2008	44	18	39	18	100
2009	35	14	40	11	79
Total	208	83	40	81	96



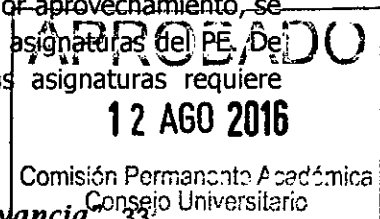
Como se puede apreciar en la Tabla 10, las modalidades de Promedio General (67%) y Examen General de Egreso (27%) han sido las modalidades que más han sido seleccionadas por los alumnos para obtener su título, lo anterior se puede explicar por la pronta incorporación al mercado laboral que tienen nuestros egresados.

Tabla 10. Modalidad de Titulación Utilizada.

Año de Ingreso	Titulados (#)	Modalidad de Titulación				
		Promedio General	EGEL	Curso en Opción a Titulación	Tesis	Otras
2004	7	6	1	-	-	-
2005	19	8	8	2	-	1
2006	13	7	5	-	-	-
2007	14	11	3	-	-	-
2008	18	15	3	-	-	-
2009	11	7	2	-	2	-
Total	81	54	22	2	2	1

En el periodo agosto diciembre de 2013 se realizaron encuestas a alumnos de la Licenciatura en Ingeniería de Software. Para lo anterior se consideraron a todos los alumnos inscritos en el equivalente a los semestres 5 al 8, y se obtuvo una muestra de 30 estudiantes (51%). El instrumento empleado consta de 24 preguntas divididas en cinco secciones, y se presenta en el Anexo D.

El 63% de la muestra de alumnos considera que para lograr un mejor aprovechamiento, se requieren temas o conocimientos previos adicionales a los ofertados en las asignaturas del PE. De acuerdo a los comentarios de los estudiantes, el 30% del total de las asignaturas requiere



conocimientos previos; entre ellas están: Arquitectura de Computadoras, Administración de Proyectos y Probabilidad.

El 83% de los alumnos considera que la estructura de la malla curricular del plan de estudios les permite aplicar los conocimientos y habilidades adquiridos en la elaboración de tareas y proyectos.

El 83% de los alumnos expresaron que algunas asignaturas, principalmente del área de matemáticas, contienen temas que no contribuyen en su formación. Entre las asignaturas mencionadas están: Álgebra Superior I y II, Álgebra Lineal, Gestión Tecnológica, Cálculos Diferencial e Integral.

El 80% de los alumnos señaló que en algunas áreas de conocimiento del PE faltan contenidos por incluir. Entre las áreas de conocimiento se mencionan: Programación e Ingeniería del Software, Matemáticas, y Software de Base. Por otro lado, el 67% de los estudiantes encuestados manifestó que los conocimientos y habilidades adquiridos durante su formación profesional han sido útiles en la elaboración de tareas, proyectos, prácticas profesionales y servicio social.

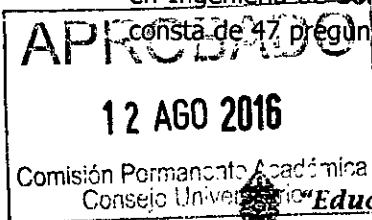
En cuanto a la infraestructura, la mayoría de los estudiantes considera que el campus cuenta con instalaciones académicas apropiadas, así como con el equipo de cómputo y el software apropiado y disponible para satisfacer las demandas académicas. No obstante, los alumnos indicaron una opinión desfavorable sobre el servicio de internet que provee la Facultad. Por otra parte, más de la mitad (aproximadamente el 70%) de los estudiantes considera que el acervo bibliográfico disponible en la biblioteca es apropiado y suficiente.

En lo referente a las asignaturas, el 41% de los estudiantes contestaron que los programas de estudio se cubren en su totalidad, mientras que el 100% considera que las asignaturas se evalúan de manera adecuada a través de exámenes, proyectos, y otras actividades. El 76% de los estudiantes considera que la licenciatura está alineada con los objetivos y el enfoque presentado en Plan de Estudios.

El 80% de los alumnos considera que las actividades extracurriculares que se realizan durante la licenciatura son adecuadas para complementar su formación, a través del Servicio Social y las Prácticas Profesionales. Además, el 70% de los estudiantes respondió que se presentan buenas oportunidades para integrarse en proyectos o en otras actividades de vinculación con el medio interno y externo.

En cuanto al grado de satisfacción general, todos los alumnos indicaron que se sienten satisfechos estudiando la Licenciatura en Ingeniería de Software.

Por lo que respecta a las opiniones de los profesores, en el periodo agosto diciembre de 2013 se les aplicaron encuestas a los 31 profesores que impartieron alguna asignatura de la Licenciatura en Ingeniería de Software durante los años escolares 2012 a 2013. El instrumento que se empleó consta de 47 preguntas divididas en cuatro secciones, y se muestra en el Anexo C.





Como resultado de la encuesta, el 68% de los profesores considera pertinente el objetivo general del plan de estudios, el 13% no emitió una opinión al respecto, mientras que el 19% restante externó que no es pertinente atribuyéndolo a la falta de claridad con respecto al uso del término áreas de concentración. Por otro lado, el 87% considera que los objetivos específicos son congruentes con el objetivo general, el 7% no emitió una opinión al respecto, y del 6% restante, un profesor comentó que no es perceptible el proceso de soporte en el objetivo general. Así mismo, el 81% de los encuestados considera que los objetivos específicos son congruentes con el perfil del egresado, el 16% no emitió una opinión al respecto y el 3% restante comentó que no es perceptible la actividad de administración de proyectos, tanto en el objetivo general como en los objetivos específicos.

El 45% de los profesores considera que sería conveniente agregar alguna asignatura obligatoria, el 3% no emitió una opinión al respecto, mientras que el 52% restante externó que no sería conveniente. Entre las propuestas de asignaturas a agregar se encuentran: Verificación y Validación de Software, Algoritmia, Administración de Bases de Datos, Desarrollo de Aplicaciones Web, Desarrollo de Aplicaciones Distribuidas, Sistemas Empotrados, Sistemas en Tiempo Real, Cálculo Vectorial y Geometría Analítica.

El 68% de los profesores piensa que no es necesario excluir alguna asignatura del plan de estudios, el 9% no omitió una opinión o no sabe, mientras que el 23% restante respondió que sería conveniente excluir algunas asignaturas obligatorias del plan de estudios. Entre las propuestas de asignaturas a excluir se encuentran: Métodos Formales en Especificación y Diseño de software, Programación en la Web e Interacción Humano Computadora.

En cuanto a infraestructura, el 94% de los profesores considera que las aulas de clase para el desarrollo de sus cursos están en buenas condiciones, mientras que 6% opinó que este recurso es regular. Esto contrasta con la percepción que se tiene acerca de la disponibilidad de las aulas de cómputo o laboratorios, ya que sólo el 26% de los docentes indicó que es excelente, el 42% opinó que este recurso es adecuado, mientras que el 16% restante externó que es regular.

En lo referente a la disponibilidad de los libros en la biblioteca del Campus, el 74% de los profesores considera que el material bibliográfico se encuentra disponible. No obstante, el número de ejemplares de los libros en la biblioteca es insuficiente, ya que sólo el 39% de los profesores encuestados considera que es suficiente, mientras que el 55% de los docentes considera que la cantidad de revistas y otros materiales bibliográficos es insuficiente.

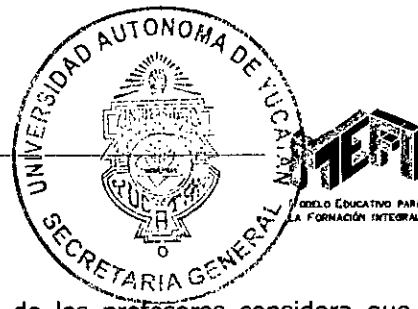
El 55% de los maestros de la LIS considera que el software disponible en el centro de cómputo es adecuado para la impartición de los cursos. La mayoría (87%) de los profesores indicó que está conforme con la disponibilidad del equipo de cómputo y audiovisual. Por otro lado, ningún profesor considera excelente el servicio de internet; sólo el 6% de los profesores encuestados lo considera bueno. En general, los espacios asignados a los profesores fueron bien evaluados, ya que el 77% está conforme con su cubículo. Finalmente, sólo el 55% de los profesores considera que el número de alumnos por salón es aceptable.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





En cuanto a los Programas de asignatura, el 95% de los profesores considera que su asignatura contribuye al cumplimiento de los objetivos del plan de estudios; el 92% de los profesores considera que su asignatura contiene objetivos pertinentes. En cuanto a la congruencia con el perfil de egreso, el 94% de los profesores contestó que los objetivos de las asignaturas son congruentes con el perfil de egreso; el 86% de los profesores considera que los objetivos están definidos de acuerdo con las tendencias del área.

El 74% de los profesores considera que el número de horas asignado para sus materias es suficiente, mientras que el 85% de los profesores contestó que sus materias tienen un balance adecuado entre horas teóricas y prácticas.

El 80% de los profesores considera que los criterios de evaluación de las asignaturas son adecuados, y en cuanto a la bibliografía utilizada en las asignaturas, el 82% manifestó que se cuenta con bibliografía actualizada y adecuada. El 83% considera que los antecedentes académicos para cursar una asignatura son adecuados.

En lo que se refiere a impartir asignaturas en el período de verano, el 63% considera que si es posible. Por otro lado, el 48% indicó que su asignatura se puede impartir en la modalidad en línea.

En lo referente a la sección del Proceso de Enseñanza Aprendizaje, los profesores utilizan las siguientes estrategias: el 98% presenta sus materiales a través de exposiciones; el 60% utiliza estudios de casos en sus asignaturas; el 87% indicó que utiliza el método de lluvia de ideas; el 97% alienta a los alumnos a trabajar en equipos; el 24% manifestó el manejo de portafolio.

Por lo que respecta al uso de material didáctico, el 83% de los profesores contestó que utiliza algún tipo de apoyo de este tipo; el 87% emplea las tecnologías de información en sus clases.

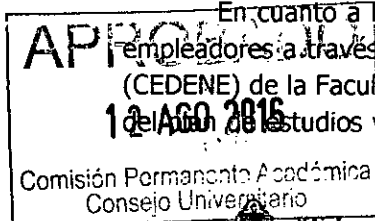
En lo que se refiere a propiciar la discusión en el aula, el 66% de los profesores contestó que si la realiza, y sólo el 25% indicó que promueve la discusión no presencial. El 46% contestó que utilizan taller o laboratorio para impartir sus clases; el 57% emplea talleres de solución de problemas.

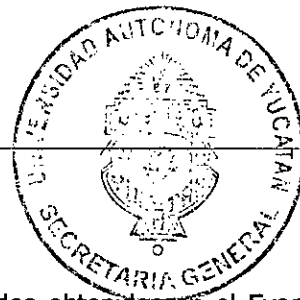
En cuanto a las visitas a centros profesionales, sólo el 6% de los profesores contestaron que realizan visitas durante la impartición de sus asignaturas; el 86% indicó que no realizan visitas a centros de investigación, mientras que el 9% indicó que integran grupos de investigación durante la impartición de sus asignaturas.

El 37% de los profesores manifestó el uso de un segundo idioma durante el curso. El 94% ofrece asesorías a sus alumnos, mientras que el 66% reporta actividades de tutoría.

2.6.2 Evaluación externa

En cuanto a la evaluación externa se analizó la información que se obtuvo de egresados y empleadores a través del estudio de mercado laboral a cargo de Centro de Desarrollo de Negocios (CEDENE) de la Facultad de Contaduría y Administración, opiniones de expertos en el área respecto del Área de Estudios vigente, así como las recomendaciones hechas por el CONAIC como parte del





proceso de acreditación; se incluyen también los resultados obtenidos en el Examen General de Egreso del CENEVAL de 2008 a la fecha.

Los egresados entrevistados mencionaron que su motivación en cursar la Licenciatura en Ingeniería de Software de la Facultad de Matemáticas se debió al interés que despertó en ellos el auge de las tecnologías de la información.

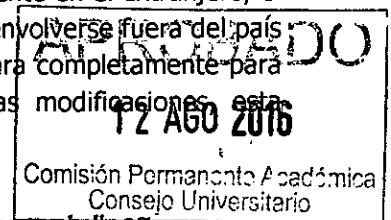
Con relación al plan de estudios, la gran mayoría de los egresados expresaron sentirse muy satisfechos o completamente satisfechos con la formación recibida. La misma tendencia se mantiene al evaluar el contenido de dicho plan, donde más de la mitad de las opiniones se mostraron a favor de mantener el nivel de los conocimientos teóricos y los aspectos metodológicos, pero ampliando los conocimientos técnicos de la disciplina y la práctica profesional. Fue posible constatar que al menos el 50% de los egresados considera que la instrucción obtenida les ha aportado capacidad analítica y lógica, capacidad para la identificación y solución de problemas en el campo profesional, habilidades para la búsqueda de información y habilidades para aplicar los conocimientos. Por otra parte se sugirió, de manera recurrente, reforzar el desarrollo de las habilidades para la comunicación oral y/o escrita así como los conocimientos generales de la naturaleza científica y/o humanística. Con menor frecuencia, se sugirió la conveniencia de incluir más asignaturas de desarrollo de software en los primeros semestres y no solamente otorgar protagonismo a las materias del área de matemáticas.

Al solicitar opiniones sobre la flexibilidad del plan, el estudio cita que las respuestas obtenidas estuvieron divididas, ya que algunos reprobaron por completo esta modalidad al encontrar en él más desventajas que ventajas, mientras que otros consideraron acertado dicho enfoque. Sólo un pequeño número de entrevistados mencionó que la estructura flexible del plan de estudios les resultó indistinta.

Con respecto a la planta académica, la mayoría de los egresados opinó que no realizaría cambios pues concuerdan con el hecho de que los profesores dominan sus áreas de especialidad y las técnicas de enseñanza empleadas son las adecuadas. Aunque enfatizaron su preferencia hacia aquellos que aún ejercían o habían ejercido su profesión fuera de la Facultad ya que podían hablar acerca de sus experiencias laborales en el área. Es importante mencionar que un porcentaje reducido externó su preocupación acerca de la importancia de ofrecer capacitación docente a ciertos profesores para la mejora de su práctica académica.

En cuanto a las asignaturas del plan de estudios, la mayor parte de los egresados externó que éstas fueron apropiadas. Únicamente recomendaron incluir algunas asignaturas del área de ciencias sociales y humanidades así como algunas más especializadas tales como Inteligencia Artificial, Visión Computacional o Ingeniería de Datos. Cabe señalar que el 30% de ellos considera que es importante la inclusión de asignaturas de formación de emprendedores al plan de estudios.

En cuanto a la percepción que tienen para desempeñarse exitosamente en el extranjero, el 50% de los egresados encuestados considera que está preparado para desenvolverse fuera del país satisfactoriamente. El 50% restante piensa que la licenciatura no los prepara completamente para este tipo de experiencia. Sin embargo, sugirieron que efectuando ciertas modificaciones, esta





preparación podría mejorar; por ejemplo, al fomentar aún más el estudio del idioma inglés y al incluir tecnologías más recientes en la impartición de las clases.

Con relación al nivel de satisfacción en el ámbito profesional, más de la mitad de los encuestados externaron sentirse muy satisfechos con el nivel de ingresos percibidos, la posición jerárquica alcanzada y la posibilidad de responder a problemas de relevancia social. Igualmente, expresaron sentirse completamente satisfechos con las actividades profesionales desarrolladas, el ambiente de trabajo, la aplicación de los conocimientos adquiridos en la licenciatura y la posibilidad de hacer algo de provecho para la sociedad.

Finalmente, fue de gran relevancia saber que el 100% de los entrevistados está satisfecho con la decisión de haber estudiado la licenciatura; la razón más significativa se relaciona con la preparación y los conocimientos recibidos para desarrollarse en el mundo laboral, seguida por la obtención de un panorama más amplio de lo que significa la industria del software.

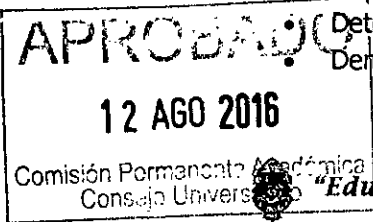
Con respecto a las opiniones de los empleadores, la mayoría considera que los egresados de la Licenciatura en Ingeniería de Software de la Facultad de Matemáticas son proactivos y están capacitados para realizar sus funciones. Señalan que los egresados poseen un alto grado de capacidad técnica para resolver los problemas que se les presentan, pero tienen dificultades para comunicar esos problemas de manera clara a las personas que no son del área de tecnologías de la información.

La mayoría de los entrevistados situaron a la Universidad Autónoma de Yucatán como una de las primeras opciones a elegir al contratar egresados de las carreras relacionadas con el área de tecnologías de la información. Los empleadores comentan que una de las razones por las que contratan a egresados de la Licenciatura en Ingeniería de Software se debe a que la carrera está bien enfocada al desarrollo de software y los egresados cuentan con una formación académica adecuada y de buen nivel. Igualmente, están actualizados con respecto a lenguajes de programación y siempre demuestran deseos de aprender, además de que saben documentar muy bien su trabajo. También cuentan con una alta habilidad de análisis y entendimiento de procesos, gran capacidad matemática, conocimientos sobre metodologías ágiles, facilidad para desarrollar sistemas en tiempo y forma, buen nivel y lógica de programación y capacidad para resolver problemas.

Las características que los empleadores buscan en una persona egresada de la Licenciatura en Ingeniería de Software son: responsabilidad, proactividad, habilidad para trabajar en equipo y poseer conocimientos teóricos y prácticos.

En el Estudio de Mercado Laboral (CEDENE, 2015), realizado para la presente propuesta, los empleadores señalan que, además del conocimiento de la disciplina, son importantes las siguientes habilidades y actitudes para que un profesionista se desempeñe adecuadamente:

- Poseen habilidades para trabajar en equipo.
- Detectan y diagnostican problemas en su área.
- Demuestran habilidades para aprender nuevos conocimientos y estilos culturales.





- Comunican adecuadamente ideas e información de forma escrita.
- Analizan situaciones y toman decisiones apropiadas.
- Demuestran tener habilidades para acceder y utilizar información relevante.
- Demuestran habilidades para realizar el trabajo para el cual fueron formados.
- Poseen iniciativa, respetan diferentes perspectivas y puntos de vista.
- Poseen capacidad para adaptarse al cambio, demuestran compromiso con la sociedad y con el medio ambiente.
- Mantienen un comportamiento ético.

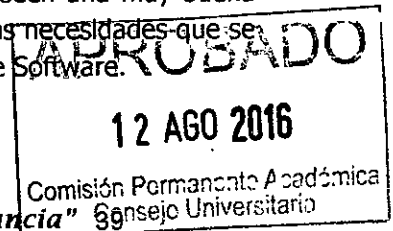
En general los empleadores expresaron que el grado de capacitación que requirieron los egresados fue mínima, más bien enfocada a necesidades especiales de cada empresa. No obstante, la mayoría coincidió en la conveniencia de reforzar ciertas áreas como:

- Administración de proyectos
- Bases de datos
- Desarrollo de sistemas y aplicaciones para dispositivos móviles
- Diseño e implementación de redes de computadoras
- Administración de servidores
- Seguridad en informática

Entre las sugerencias obtenidas sobre el modo de posicionar a la Facultad de Matemáticas como líder en el área de la Ingeniería de Software, se tienen:

- Mantener actualizados los planes y programas de estudio con base en los cambios tecnológicos que se presenten.
- Brindar mayor relevancia a la formación integral de las habilidades para el desarrollo humano.
- Fomentar una mayor vinculación laboral.
- Dar a conocer a los empleadores las diferencias que existen entre las licenciaturas del área de computación impartidas en la Facultad.
- Mantener un alto nivel académico.
- Mejorar las habilidades de comunicación oral y escrita de los egresados.
- Difundir entre los egresados información referente a opciones de estudios de posgrado.
- Incrementar el número de actividades extracurriculares.

En síntesis, los empleadores están satisfechos con las habilidades técnicas de los egresados. El mercado laboral de la Licenciatura en Ingeniería de Software se encuentra en crecimiento, ya que las empresas en el estado y la región están modernizándose y requieren de personal capacitado para usar nuevos equipos o nueva tecnología. La Facultad de Matemáticas está bien posicionada entre los empleadores e informantes clave y consideran que los jóvenes egresados poseen una muy buena preparación. Sin embargo, algunos empresarios no conocen la licenciatura y las necesidades que se pudieran satisfacer en su negocio al contratar a un Licenciado en Ingeniería de Software.





Con el propósito de conocer las opiniones de expertos en el área de Ingeniería de Software de diferentes instituciones nacionales y extranjeras respecto de los objetivos del plan de estudios, perfil de egreso, así como de las áreas de concentración, se contactó a especialistas nacionales y extranjeros. La información recabada proviene de investigadores que laboran en las siguientes instituciones:

- Universidad Veracruzana,
- Universidad de Baja California
- Universidad Autónoma de Aguascalientes
- Centro de Investigación en Matemáticas (Unidad Zacatecas)
- Universidad Politécnica de Madrid
- Pontificia Universidad Católica del Perú.



Con base en las opiniones vertidas en el instrumento administrado, los expertos consideran que los objetivos de la carrera son adecuados para cubrir el perfil enfocado a formar un Ingeniero de Software que cubra las necesidades actuales de las empresas.

En lo que se refiere al perfil de egreso, la opinión de los expertos coincide en que los conocimientos, habilidades y actitudes que se incluyen en el actual perfil de egreso son los adecuados para formar profesionistas con conocimientos técnicos, buenas prácticas en la aplicación de las metodologías, habilidades de pensamiento, capacidad de aprendizaje, que la industria demanda actualmente.

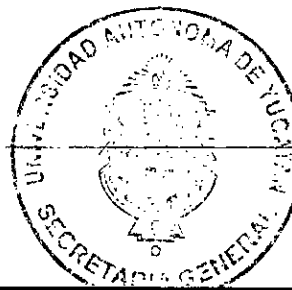
Con relación a la pertinencia de los conocimientos, habilidades y actitudes, algunos expertos señalan que sería conveniente agregar temas en las áreas de: diseño centrado en el usuario, estándares a lo largo del proceso (ISO, SPICE, etc.); en cuanto a las habilidades se recomienda promover: innovación para la creación de nuevos procesos y productos, capacidad de abstracción, diseño de soluciones con arquitectura viable; finalmente, respecto a las actitudes comentaron que se debe fomentar la disposición para el uso de tecnologías de punta.

En lo que se refiere a las Áreas de Concentración, los expertos consideran que las actuales son pertinentes en la formación de un Ingeniero de Software; para el área de Mejora de Procesos, algunos proponen incluir modelos de mejora de servicios y adquisición; otros proponen la creación del área de Métodos Ágiles. También se sugiere revisar el nombre del área de Desarrollo de Aplicaciones Web y se propone un nombre genérico que se ajuste a las tendencias de la tecnología actual.

Tabla 11a. Recomendaciones de los CIEES

Categoría	Recomendaciones	Justificación	Atención
	3. Restructurar el Plan de Estudios	Se detectó una inadecuada distribución de horas en los grupos de asignaturas, observándose que los requisitos académicos de algunas asignaturas y el orden de <i>Metodología de la Investigación y Gestión Tecnológica</i> se deben ajustar dado que no considera	Se propone la Modificación del Plan de Estudios en la que se incluyen algunas nuevas asignaturas (p.e. Mantenimiento de Software) y se eliminan otras (p.e. gestión tecnológica).

APR 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Categoría	Recomendaciones	Justificación	Atención
		una seriación mínima necesaria entre asignaturas. Además, las materias de matemáticas y probabilidad y estadística no están enfocadas a la Licenciatura en IS si no a las Licenciatura del área de Matemáticas, impartidas en la Facultad, y en algunas materias no se tiene congruencia entre los objetivos de formación integral del alumno.	En todas las asignaturas propuestas se incorporan competencias que pretenden contribuir al perfil de egreso.
	5. Asegurar la congruencia del Plan de Estudios con el Perfil de Egreso	El Perfil de Egreso debe hacer referencia a ciertas habilidades y competencias, que no se garantizan en el plan de estudios, como por ejemplo: emprendedor.	El perfil de Egreso se expresa en términos de competencias; se incorpora al plan de estudios un <i>Taller de Emprendedores</i> .

En lo referente a los procesos de evaluación externa por parte de los organismos externos, el programa educativo ha sido evaluado por los Comités Interinstitucionales de Evaluación de la Educación Superior (CIEES) en 2012, en particular, por el Comité de Ingeniería y Tecnología; en dicha evaluación el programa fue clasificado en el Nivel 2 del Padrón de Programas de ES reconocidos por su Buena Calidad, y emitió un conjunto de 18 recomendaciones sobre las 9 categorías consideradas en el proceso de Evaluación. La Tabla 11a presenta las recomendaciones sobre la categoría vinculada con el Modelo Educativo y Plan de Estudios, así como sus respectivas justificaciones, así mismo se incluye una tercera columna donde se sintetiza las atenciones a dichas recomendaciones.

El programa educativo también ha sido evaluado en 2013 por el Comité de Acreditación del Consejo Nacional de Acreditación en Informática y Computación, A.C. (CONAIC), quién otorgó la acreditación del programa educativo para un período de 5 años a partir del 12 de junio del mismo año, emitiendo un conjunto de 42 recomendaciones sobre las 11 categorías que considera el Marco de Referencia de dicha evaluación. La Tabla 11b presenta las recomendaciones realizadas categoría referente al Plan de Estudios, así como la atención a las mismas.

Tabla 11b. Recomendaciones del CONAIC.

Categoría	Recomendaciones	Atención
Plan de Estudios	a) Se recomienda que los programas analíticos de las asignaturas contemplen horas teóricas y prácticas y que tengan la bibliografía actualizada.	Los programas de estudio de las asignaturas en el MEFI consideran horas presenciales y horas no presenciales.
	b) Se recomienda que los estudiantes realicen trabajo en equipo de manera interdisciplinaria.	El MEFI promueve el trabajo en equipos interdisciplinarios.
	c) Se recomienda que el Plan de Estudios contenga asignaturas que fomenten el desarrollo de valores éticos y sociales.	El MEFI establece la inclusión de la asignatura <i>Responsabilidad Social Universitaria</i> en todos los planes de estudio.
	d) Se recomienda ofertar una mayor cantidad de asignaturas optativas por semestre, que sean de la especialidad y en horarios diferentes.	Por atender.

APROBADO
 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Categoría	Recomendaciones	Atención
	e) Se recomienda la actualización del programa educativo de fondo y no sólo de forma, es decir, incorporación del nuevo modelo educativo, bibliografía actualizada, contenidos, entre otros.	Se propone la Modificación del Plan de Estudios con base en el MEFI.
	f) Se recomienda contar con un procedimiento oficial y funcional, para la revisión y actualización del plan de estudios que incluya la participación de cuerpos colegiados, grupo de asesores externos, egresados e investigadores reconocidos.	El proceso de Modificación de Planes de estudio para su alineación al MEFI contempla un proceso riguroso y formal.
	g) Se recomienda implementar un procedimiento permanente de evaluación curricular.	Por atender.
	h) Se recomienda establecer estrategias que permitan que los alumnos opten por la titulación por tesis o por proyectos de investigación.	El MEFI contempla las modalidades de Titulación: (1) Tesis y (2) EGEL.

Como último aspecto de la evaluación externa, se reporta en la Tabla 12 el desempeño histórico de los sustentantes del Examen General de Egreso de Licenciatura (EGEL), examen estandarizado administrado por el Centro Nacional de Evaluación para la Educación Superior; en el caso del programa educativo se corresponde con el EGEL-ISOFT. Los datos presentados corresponden del período comprendido de enero de 2008 a junio de 2015.

Tabla 12. Histórico de sustentantes EGEL.

	2008	2009	2010	2011	2012	2013	2014	2015
EGEL-ISOFT	2	-	2	2	4	4	10	-
EGEL-ISOFT (S)	1	-	-	-	4	4	7	2
EGEL-ISOFT (N)	-	-	-	-	-	-	-	-
Total	3	0	2	2	8	8	17	2

Cabe mencionar que el desempeño de los egresados en el EGEL-ISOFT durante el período comprendido de julio de 2013 a junio de 2014, generó las condiciones para gestionar —en la convocatoria 2013-2014 del CENEVAL— la evaluación y el ingreso del programa educativo al Padrón de Programas de Licenciatura de Alto Rendimiento Académico-EGEL en el Estándar 1. Así mismo, el desempeño de julio de 2014 a junio de 2015 permitió refrendar la pertenencia del programa educativo en el Estándar 1 de dicho Padrón.

Finalmente se comentaron las evaluaciones externas al programa educativo, y el que en los dos últimos años ha recibido el reconocimiento a su calidad por parte de dos de los principales organismos de evaluación reconocidos a nivel nacional.

APROBADO
27
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario

Conclusiones generales





En la actualidad, es prácticamente imposible concebir una organización sin pensar en los sistemas de software inmersos en su operación. Así mismo, las actividades de nuestra vida cotidiana se ven altamente influenciadas por las Tecnologías de Información y Comunicación. No obstante, la sociedad enfrenta problemas cada vez más complejos que demandan sistemas que respondan a dichas necesidades.

Para resolver los problemas que presenta la sociedad se requiere capital humano altamente capacitado en el área de Ingeniería de Software, el cual por medio del uso sistemático y disciplinado de metodologías, técnicas y herramientas, desarrollen de manera cuantificable y bajo un comportamiento ético y moral, sistemas de software que den respuesta a las necesidades de servicios e información de la sociedad.

La Licenciatura en Ingeniería de Software de la UADY creada en 2004 fue la primera alternativa de formación en el nivel licenciatura en México, a lo largo de los 12 años de su corta historia ha presentado una mejora creciente en cuanto a sus indicadores de calidad. Por citar algunos, se tiene una demanda creciente del programa educativo, el estado actual de los indicadores de trayectoria escolar (p. e. deserción: 60%), el crecimiento en los indicadores de egreso (40%) y titulación (39%), tomado en consideración el período de permanencia del plan de estudios.

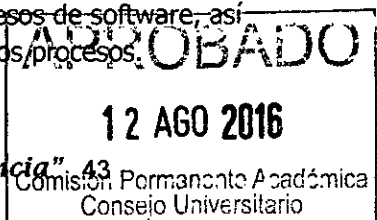
Por lo anterior, y con base en los estudios realizados sobre pertinencia social, factibilidad, estado del arte, así como el de evaluación curricular, en esta propuesta se han presentado los referentes social, disciplinar, profesional, institucional, así como la evaluación interna y externa, que sustenta la propuesta de modificación al plan de estudios para su alineación al Modelo Educativo para la Formación Integral, pero sobre todo, para mantener su vigencia y pertinencia, y con ello seguir contribuyendo al desarrollo científico y tecnológico, en nuestra sociedad.

Específicamente, en la actualización del Plan de Estudios de la Licenciatura en Ingeniería de Software toma en cuenta lo siguiente:

- Incorporar los elementos fundamentales del MEFI como la educación basada en competencias y la formación integral de los estudiantes.
- Incorporar las tendencias de la profesión y la disciplina.
- Considerar las sugerencias de los alumnos, profesores, egresados y empleadores.
- La necesidad de formar profesionales altamente capacitados que colaboren en la solución de los problemas de la sociedad, y promuevan la innovación de los procesos y la utilización de tecnología de punta.
- Las recomendaciones de los organismos evaluadores (p. e. CONAIC) externos para mantener una evaluación positiva del Programa Educativo.

2.7.1 Justificación de las Áreas de Competencia.

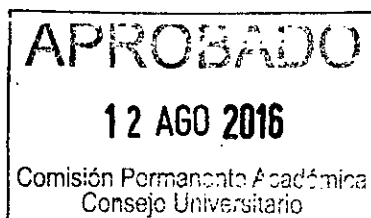
La Ingeniería de Software, como disciplina, establece un conjunto de conocimientos, vinculados con las áreas de desarrollo, mantenimiento y de gestión de los procesos de software, así como el estudio de fenómenos propios para proponer y validar mejoras en dichos procesos.



Para determinar las áreas de competencia se consideraron: la definición de Ingeniería de Software que propone la IEEE, las áreas de conocimiento definidas en el SWEBOK, así como el Perfil B (Licenciatura en Ingeniería de Software) de los Modelos Curriculares de la ANIEI. Cabe mencionar que estos elementos fueron descritos en las secciones de referentes de este documento.

Por lo anterior, la propuesta de modificación del plan de estudios de la Licenciatura en Ingeniería de Software contempla cuatro áreas de competencia:

- *Desarrollo de software.*
- *Mantenimiento de Software.*
- *Administración de los Procesos de Software.*
- *Innovación en Ingeniería de Software.*





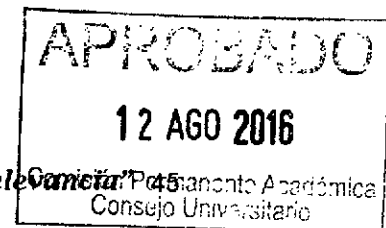
3. INTEGRACIÓN DE LOS EJES DEL MEFI

El MEFI promueve la Formación Integral del estudiante por medio de la interacción de seis ejes: Responsabilidad Social, Flexibilidad, Innovación, Internacionalización, Educación Centrada en el Aprendizaje (ECA) y Educación Basada en Competencias (EBC).

En esta sección se describe la manera en que se aborda cada uno de estos ejes en el plan de estudios de LIS.

Tabla 13. Ejes del MEFI.

EJES DEL MEFI	Estrategias y acciones
ECA	<ul style="list-style-type: none"> ▪ Fomenta mayor participación del estudiante en las sesiones de clase, considerándolo como alumno activo, y no como un alumno pasivo que sólo recibe información del profesor. ▪ Considera al estudiante como protagonista en el proceso educativo, permitiendo que él mismo se haga responsable de su propio aprendizaje. ▪ Reduce las horas presenciales, proporcionando espacio para que el alumno dirija por sí mismo la construcción de su conocimiento, ayudándolo a aprender. ▪ Reconoce el esfuerzo del estudiante con actividades de aprendizaje dentro y fuera del aula; entre esas actividades incluye escenarios reales como las prácticas profesionales y el servicio social, donde estas últimas sean acreditadas. ▪ Enfatiza la enseñanza centrada en el aprendizaje significativo, y promueve el trabajo en equipos interdisciplinarios.
EBC	<ul style="list-style-type: none"> ▪ Establece perfiles de ingreso y egreso basados en el modelo por competencias. ▪ Se incrementan y/o establecen diversas estrategias de evaluación en las asignaturas del programa de LIS, tales como: el desarrollo de proyectos, estudios de caso, y portafolio de evidencias, entre otras. ▪ Se incorpora en dicho programa las competencias genéricas establecidas en el MEFI, y se agregan esquemas de acreditación que consideran el nivel de dominio en el que un estudiante ha alcanzado la competencia de una asignatura. ▪ Se fomenta en las sesiones de clase espacios para que los alumnos conozcan las competencias de su asignatura en curso, de tal modo que el estudiante identifique de qué forma contribuye en su formación profesional.





EJES DEL MEFI	Estrategias y acciones
Flexibilidad	<ul style="list-style-type: none"> Se mantiene la organización y administración de asignaturas, de tal modo que permita a los estudiantes seleccionar su propia carga académica, con ayuda de su tutor, de acuerdo a sus necesidades, y sin omitir los requisitos académicos. Se considera en el plan de estudios asignaturas optativas. Se considera en el plan de estudios asignaturas libres. Permite cursar asignaturas de otras instituciones de educación superior locales o nacionales reconocidas. Se considera un máximo de catorce semestres para concluir el PE, de modo que el alumno pueda seleccionar las asignaturas a cursar en cada semestre.
Innovación	<ul style="list-style-type: none"> Las asignaturas estarán mediadas a través de entornos tecnológicos con el uso de la plataforma. Incorpora estándares de evaluación basados en el Modelo por Competencias. Diversifica las estrategias de enseñanza aprendizaje utilizadas en las asignaturas: presencial, mixta y no presencial. Incorpora estrategias de aprendizaje variadas indicadas en los programas de estudio. Fomenta el uso de software específico para la resolución de problemas de alguna área definida. Promueve el contacto de profesores de la licenciatura con asociaciones académicas, tanto nacionales como internacionales.
Responsabilidad Social	<ul style="list-style-type: none"> Incorpora las asignaturas de Responsabilidad Social Universitaria, Cultura Maya y Taller de Emprendedores en el plan de estudios, lo cual pretende que el egresado promueva el bienestar social a través del desarrollo sustentable y fomente la cultura emprendedora, respectivamente. Mantiene el Servicio Social como una actividad obligatoria, para promover la participación en proyectos que generen un impacto en beneficio de la comunidad.
Internacionalización	<ul style="list-style-type: none"> Establece como mínimo el nivel B1 de dominio del idioma inglés como requisito de permanencia. Se incrementa el manejo de referencias en inglés en cada asignatura. Apoya la movilidad estudiantil nacional e internacional, por medio de intercambios académicos. Incorpora referentes internacionales de la IEEE y la ACM en el diseño de la malla curricular de LIS.

APR
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

4. OBJETIVO GENERAL DEL PLAN DE ESTUDIOS

Formar profesionales en el proceso de desarrollo y la evolución de sistemas de software, de pequeña a gran escala, que solucionen problemas en diferentes áreas de aplicación, utilizando las herramientas adecuadas para optimizar los recursos de tiempo y costo, con apego a la ética profesional.



5. PERFIL DE INGRESO

Para esta propuesta de modificación, se reestructura la lista de las competencias deseables del perfil de ingreso, según el acuerdo 444 de la SEP por el que se establecen las competencias que constituyen el marco curricular común del Sistema Nacional de Bachillerato (SNB).

Matemáticas

1. Construye e interpreta modelos matemáticos mediante la aplicación de procedimientos aritméticos, algebraicos, geométricos y variacionales, para la comprensión y análisis de situaciones reales, hipotéticas o formales.
2. Formula y resuelve problemas matemáticos, aplicando diferentes enfoques.
3. Explica e interpreta los resultados obtenidos mediante procedimientos matemáticos y los contrasta con modelos establecidos o situaciones reales.
4. Argumenta la solución obtenida de un problema, con métodos numéricos, gráficos, analíticos o variacionales, mediante el lenguaje verbal, matemático y el uso de las tecnologías de la información y la comunicación.
5. Analiza las relaciones entre dos o más variables de un proceso social o natural para determinar o estimar su comportamiento.
6. Cuantifica, representa y contrasta experimental o matemáticamente las magnitudes del espacio y las propiedades físicas de los objetos que lo rodean.
7. Elige un enfoque determinista o uno aleatorio para el estudio de un proceso o fenómeno, y argumenta su pertinencia.
8. Interpreta tablas, gráficas, mapas, diagramas y textos con símbolos matemáticos y científicos.

Ciencias Experimentales

1. Establece la interrelación entre la ciencia, la tecnología, la sociedad y el ambiente en contextos históricos y sociales específicos.
2. Fundamenta opiniones sobre los impactos de la ciencia y la tecnología en su vida cotidiana, asumiendo consideraciones éticas.
3. Identifica problemas, formula preguntas de carácter científico y plantea las hipótesis necesarias para responderlas.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

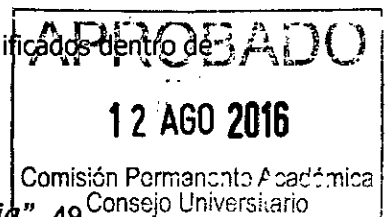




4. Obtiene, registra y sistematiza la información para responder a preguntas de carácter científico, consultando fuentes relevantes y realizando experimentos pertinentes.
5. Contrasta los resultados obtenidos en una investigación o experimento con hipótesis previas y comunica sus conclusiones.
6. Valora las preconcepciones personales o comunes sobre diversos fenómenos naturales a partir de evidencias científicas.
7. Hace explícitas las nociones científicas que sustentan los procesos para la solución de problemas cotidianos.
8. Explica el funcionamiento de máquinas de uso común a partir de nociones científicas.
9. Diseña modelos o prototipos para resolver problemas, satisfacer necesidades o demostrar principios científicos.
10. Relaciona las expresiones simbólicas de un fenómeno de la naturaleza y los rasgos observables a simple vista o mediante instrumentos o modelos científicos.
11. Analiza las leyes generales que rigen el funcionamiento del medio físico y valora las acciones humanas de impacto ambiental.
12. Relaciona los niveles de organización química, biológica, física y ecológica de los sistemas vivos.
13. Aplica normas de seguridad en el manejo de sustancias, instrumentos y equipo en la realización de actividades de su vida cotidiana.

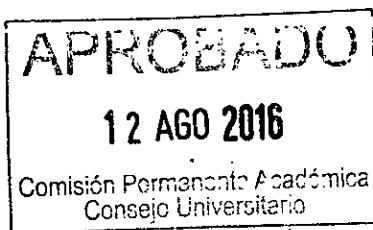
Ciencias Sociales

1. Valora las diferencias sociales, políticas, económicas, étnicas, culturales y de género y las desigualdades que inducen.
2. Analiza con visión emprendedora los factores y elementos fundamentales que intervienen en la productividad y competitividad de una organización y su relación con el entorno socioeconómico.
3. Evalúa las funciones de las leyes y su transformación en el tiempo.
4. Analiza las funciones de las instituciones del Estado Mexicano y la manera en que impactan su vida.
5. Valora distintas prácticas sociales mediante el reconocimiento de sus significados dentro de un sistema cultural, con una actitud de respeto.



Comunicación

1. Identifica, ordena e interpreta las ideas, datos y conceptos explícitos e implícitos en un texto, considerando el contexto en el que se generó y en el que se recibe.
2. Evalúa un texto mediante la comparación de su contenido con el de otros, en función de sus conocimientos previos y nuevos.
3. Plantea supuestos sobre los fenómenos naturales y culturales de su entorno con base en la consulta de diversas fuentes.
4. Produce textos con base en el uso normativo de la lengua, considerando la intención y situación comunicativa.
5. Expresa ideas y conceptos en composiciones coherentes y creativas, con introducciones, desarrollo y conclusiones claras.
6. Argumenta un punto de vista en público de manera precisa, coherente y creativa.
7. Valora el pensamiento lógico en el proceso comunicativo en su vida cotidiana y académica.
8. Identifica e interpreta la idea general y posible desarrollo de un mensaje oral o escrito en una segunda lengua, recurriendo a conocimientos previos, elementos no verbales y contexto cultural.
9. Se comunica en una lengua extranjera mediante un discurso lógico, oral o escrito, congruente con la situación comunicativa.
10. Utiliza las tecnologías de la información y comunicación para investigar, resolver problemas, producir materiales y transmitir información.
11. Analiza y compara el origen, desarrollo y diversidad de los sistemas y medios de comunicación.



6. PERFIL DE EGRESO

6.1. Áreas de competencia

El Licenciado en Ingeniería de Software aplica técnicas, herramientas, métodos y procedimientos para desarrollar y mantener software de aplicación que resuelvan problemas en diferentes áreas en una organización; también, será capaz de administrar los procesos de desarrollo, mantenimiento, calidad, y configuración del software, así como realizar investigación sobre los procesos de desarrollo de software, con el propósito de mejorar la práctica profesional.

Por lo anterior, se establecen las siguientes áreas de competencia:

- Desarrollo de software
- Mantenimiento de Software
- Administración de los Procesos de Software
- Innovación en Ingeniería de Software

A continuación se describen las competencias de egreso de cada área.



6.2. Competencias de egreso

Tabla 14. Competencias de egreso.

Desarrollo de Software	Mantenimiento de Software
<p>Competencia de egreso: Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.</p>	<p>Competencia de egreso: Mantiene productos de software heredados en diferentes contextos de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.</p>
Administración de los Procesos de Software	Innovación en Ingeniería de Software
<p>Competencia de egreso: Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.</p>	<p>Competencia de egreso: Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, mediante fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.</p>

12 AGO 2016

Comisión Permanente Académica
5 Consejo Universitario



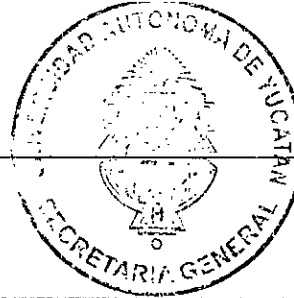
6.3. Desagregado de saberes

Tabla 15. Desagregado de saberes de la competencia de egreso Desarrollo de Software.

Desarrollo de Software		
Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.		
Saber hacer	Saber conocer	Saber ser
Modela el proceso de negocio de la organización, en función de su dominio de aplicación, empleando notaciones en el área de modelado del negocio.	Identifica las características de las notaciones de modelado del negocio para una organización, de acuerdo con las técnicas adecuadas de análisis de sistemas.	Comunica sus ideas en forma oral y escrita mediante el uso correcto del idioma español.
Aplica el proceso de requisitos en el desarrollo de productos de software que satisfagan las necesidades de la organización, empleando notaciones y/o estándares del área de requisitos.	Explica las técnicas utilizadas en el proceso de requisitos en el desarrollo de software, considerando el contexto del proyecto y los estándares establecidos.	Incorpora las tecnologías de información de manera pertinente y responsable en todos los ámbitos del quehacer profesional.
Modela una solución eficiente del problema planteado, a partir de la especificación de los requisitos, empleando metodologías de diseño de software.	Describe las metodologías de diseño en el proceso de desarrollo de software, de acuerdo con el dominio de aplicación y los paradigmas vigentes.	Manifiesta interés en la actualización de sus conocimientos y habilidades en las áreas relacionadas con la Ingeniería de Software de forma autónoma y permanente.
Modela una solución eficiente del problema de software planteado, considerando los atributos de calidad del producto y aplicando estilos y patrones arquitectónicos.	Identifica los estilos y patrones arquitectónicos de la fase de diseño, de acuerdo a los estándares establecidos.	Manifiesta objetividad, integridad y comportamiento ético en los diversos campos del quehacer profesional.
Modela el esquema de persistencia de datos de un producto de software, aplicando los métodos y técnicas adecuadas para representación de la información.	Describe los modelos de representación de información, utilizados en el diseño de software, de acuerdo con el ámbito de tratamiento de la información.	Manifiesta un pensamiento crítico en su quehacer profesional con atención y respeto a las ideas de otros.
Modela la interacción del usuario de un producto de software, considerando los atributos de usabilidad y experiencia de usuario, aplicando los métodos y técnicas de Interacción Humano-Computadora.	Describe los métodos y técnicas de Interacción Humano Computadora, considerando los atributos de usabilidad y experiencia de usuario.	Interactúa con otros profesionales en ambientes multidisciplinarios de manera respetuosa.
Construye productos de software en diferentes dominios de aplicación, empleando distintos paradigmas de programación.	Describe los diferentes métodos de representación de algoritmos de la fase de construcción de software, de manera clara y precisa.	Incorpora la creatividad en su práctica profesional de manera continua y permanente.
Utiliza bibliotecas, marcos de trabajo y componentes de software en un entorno de desarrollo, optimizando el proceso de implementación.	Describe los diferentes paradigmas de programación involucrados en el proceso de desarrollo de software, de manera clara y precisa.	Respeto la confidencialidad de la información proporcionada por el cliente, de acuerdo a los principios éticos de la disciplina.
Utiliza herramientas informáticas de soporte en el proceso de desarrollo de software, para la optimización de las actividades de las fases de dicho proceso.	Describe correctamente los diferentes marcos de trabajo involucrados en el proceso de desarrollo de software.	Demuestra una actitud objetiva y profesional en la toma de decisiones, ante diversas situaciones laborales.

APROBADO
12. AGO 2016
Comisión Permanente Académica
Consejo Universitario





Desarrollo de Software		
Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.		
Saber hacer	Saber conocer	Saber ser
<p>Evalúa la calidad del producto de software durante su desarrollo, aplicando diferentes técnicas de verificación de software.</p> <p>Aplica el proceso de validación al producto de software, considerando que los requisitos previamente establecidos satisfagan las necesidades de la organización.</p> <p>Implanta el producto de software verificado y validado en su ambiente de operación, considerando los estándares, políticas e infraestructura de la organización.</p>	<p>Identifica las herramientas informáticas idóneas que brindan soporte a los procesos de desarrollo de software, de acuerdo a la actividad y al desarrollo tecnológico.</p> <p>Describe las diferentes estrategias y técnicas de verificación utilizadas en el proceso de desarrollo de software, de manera clara y precisa.</p> <p>Describe las diferentes estrategias y técnicas de validación involucradas en el proceso de desarrollo de software, de manera clara y precisa.</p> <p>Describe los tipos de capacitación y documentación requeridos para la liberación de un producto de software, de acuerdo con el contexto del proyecto de desarrollo.</p>	<p>Manifiesta autocontrol ante situaciones bajo presión, asumiendo responsablemente las tareas que le corresponden.</p> <p>Mantiene una actitud perseverante en la resolución de problemas propios de su disciplina.</p> <p>Enfrenta situaciones innovadoras en su ámbito profesional, de manera responsable y competente.</p> <p>Manifiesta una actitud emprendedora en su práctica profesional que le permita el desarrollo de proyectos innovadores.</p>

Tabla 16. Desagregado de saberes de la competencia de egreso Mantenimiento de Software.

Mantenimiento de Software		
Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.		
Saber hacer	Saber conocer	Saber ser
<p>Aplica el proceso de mantenimiento en la evolución de los productos de software, acorde a las necesidades de cambio de la organización.</p> <p>Aplica técnicas de mantenimiento al producto heredado garantizando la continuidad de su operación.</p> <p>Modela el proceso de negocio de la organización, en función de su dominio de aplicación, empleando notaciones en el área de modelado del negocio.</p> <p>Aplica el proceso de requisitos en el mantenimiento de productos de software que satisfagan las necesidades de la organización,</p>	<p>Describe el proceso de mantenimiento de los productos de software en diferentes dominios de aplicación, de acuerdo con el marco de referencia propio de la disciplina.</p> <p>Describe los fundamentos de mantenimiento en un producto de software, de acuerdo con el marco de referencia propio de la disciplina.</p> <p>Describe las técnicas utilizadas en el mantenimiento de un producto heredado, en función del contexto del proyecto y considerando los estándares establecidos.</p> <p>Identifica las características de las notaciones de modelado del negocio, para una organización, de acuerdo con</p>	<p>Comunica sus ideas en forma oral y escrita mediante el uso correcto del idioma español.</p> <p>Incorpora las tecnologías de información de manera pertinente y responsable en todos los ámbitos del quehacer profesional.</p> <p>Manifiesta interés en la actualización de sus conocimientos y habilidades en las áreas relacionadas con la Ingeniería de Software de forma autónoma y permanente.</p> <p>Manifiesta objetividad, integridad y comportamiento ético en los diversos campos del quehacer profesional.</p>

AGOSTADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario





Mantenimiento de Software

Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

Saber hacer	Saber conocer	Saber ser
<p>considerando la viabilidad en cuanto al tiempo y costo.</p> <p>Modela una solución eficiente del problema planteado para el mantenimiento, a partir de la especificación de los requisitos, empleando metodologías de diseño de software.</p> <p>Modela una solución eficiente del problema planteado para el mantenimiento de software, considerando los atributos de calidad del producto y aplicando estilos y patrones arquitectónicos.</p> <p>Modela el esquema de persistencia de datos para el mantenimiento de un producto de software, aplicando los métodos y técnicas adecuadas para representación de la información.</p> <p>Modela la interacción del usuario para el mantenimiento de un producto de software, considerando los atributos de usabilidad y experiencia de usuario, aplicando los métodos y técnicas de Interacción Humano-Computadora.</p> <p>Mantiene productos de software en diferentes dominios de aplicación, empleando distintos paradigmas de programación.</p>	<p>las técnicas adecuadas de análisis de sistemas.</p> <p>Explica las técnicas utilizadas en el proceso de requisitos en el desarrollo de software, considerando el contexto de proyecto y los estándares establecidos.</p> <p>Describe las metodologías de diseño en el proceso de mantenimiento de software, de acuerdo con el dominio de aplicación y los paradigmas vigentes.</p> <p>Identifica los estilos y patrones arquitectónicos de la fase de diseño, de acuerdo a los estándares establecidos.</p> <p>Describe los modelos de representación de información, utilizados en el diseño de software, de acuerdo con el ámbito de tratamiento de la información.</p> <p>Describe los métodos y técnicas de Interacción Humano Computadora, utilizados durante el proceso de mantenimiento de software, considerando los atributos de usabilidad y experiencia de usuario.</p> <p>Describe los diferentes métodos de representación de algoritmos de la fase de construcción de software, de manera clara y precisa.</p> <p>Describe los diferentes paradigmas de programación involucrados en el proceso de desarrollo de software, de manera clara y precisa.</p> <p>Describe correctamente los diferentes marcos de trabajo involucrados en el proceso de mantenimiento de software.</p> <p>Describe las diferentes estrategias y técnicas de verificación utilizadas en el proceso de mantenimiento de software, de manera clara y precisa.</p>	<p>Manifiesta un pensamiento crítico en su quehacer profesional con atención y respeto a las ideas de otros.</p> <p>Interactúa con otros profesionales en ambientes multidisciplinarios de manera respetuosa.</p> <p>Incorpora la creatividad en su práctica profesional de manera continua y permanente.</p> <p>Respeto la confidencialidad de la información proporcionada por el cliente, de acuerdo a los principios éticos de la disciplina.</p> <p>Demuestra una actitud objetiva y profesional en la toma de decisiones, ante diversas situaciones laborales.</p> <p>Manifiesta autocontrol ante situaciones bajo presión, asumiendo responsablemente las tareas que le corresponden.</p> <p>Mantiene una actitud perseverante en la resolución de problemas propios de su disciplina.</p> <p>Enfrenta situaciones innovadoras en su ámbito profesional, de manera responsable y competente.</p> <p>Manifiesta una actitud emprendedora en su práctica profesional que le permita el desarrollo de proyectos innovadores.</p>
<p>Utiliza bibliotecas, marcos de trabajo y componentes de software en un entorno de desarrollo, optimizando el proceso de mantenimiento.</p> <p>Evalúa la calidad del producto de software durante su mantenimiento, aplicando diferentes técnicas de verificación de software.</p> <p>Aplica el proceso de validación al producto de software, considerando que los requisitos previamente establecidos para el mantenimiento, satisfagan las necesidades de la organización.</p> <p>Implanta, como resultado del mantenimiento, el producto de software verificado y validado en su ambiente de operación, considerando</p>		

APR
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

Mantenimiento de Software		
Saber hacer	Saber conocer	Saber ser
<p>Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.</p> <p>los estándares, políticas y la infraestructura de la organización.</p> <p>Utiliza herramientas informáticas de soporte en el proceso de mantenimiento de software, para la optimización de las actividades de las fases de dicho proceso.</p>	<p>Describe las diferentes estrategias y técnicas de validación involucradas en el proceso de mantenimiento de software de manera clara y precisa.</p> <p>Describe los tipos de capacitación y documentación requeridos para la liberación de un producto de software, de acuerdo con el contexto del proyecto de mantenimiento.</p> <p>Identifica las herramientas informáticas idóneas que brindan soporte a los procesos de mantenimiento de software, de acuerdo a la actividad y al desarrollo tecnológico.</p>	



Tabla 17. Desagregado de saberes de la competencia de egreso Administración de los Procesos de Software.

Administración de los Procesos de Software		
Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.		
Saber hacer	Saber conocer	Saber ser
<p>Estima el tamaño, esfuerzo, duración y costo de un producto de software en las fases iniciales del proyecto, aplicando distintos métodos y técnicas de estimación.</p> <p>Evalúa la viabilidad de un proyecto de software que satisfaga una necesidad identificada, con base en los estudios técnico, económico y operacional.</p> <p>Planifica las actividades asociadas a las fases de desarrollo del proyecto de software, considerando las características del producto, del equipo de desarrollo y del cliente.</p> <p>Controla los procesos involucrados en el desarrollo de software, aplicando técnicas de monitoreo y de administración de riesgos.</p>	<p>Explica los métodos y técnicas de estimación utilizados en la práctica profesional, de forma correcta.</p> <p>Describe los métodos de evaluación asociados a la viabilidad económica, técnica y operacional, comúnmente utilizados en el ámbito de la ingeniería.</p> <p>Describe los principales métodos de planificación de las actividades asociadas con las fases de desarrollo del proyecto de software, considerando las características del producto.</p> <p>Describe los modelos, técnicas y herramientas utilizados para el control de un proyecto de software, de acuerdo con el marco de referencia propio de la ingeniería de software.</p>	<p>Comunica sus ideas en forma oral y escrita mediante el uso correcto del idioma español.</p> <p>Incorpora las tecnologías de información de manera pertinente y responsable en todos los ámbitos del quehacer profesional.</p> <p>Manifiesta interés en la actualización de sus conocimientos y habilidades en las áreas relacionadas con la Ingeniería de Software de forma autónoma y permanente.</p> <p>Manifiesta objetividad, integridad y comportamiento ético en los diversos campos del quehacer profesional.</p>

RECIBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Administración de los Procesos de Software

Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.

Saber hacer	Saber conocer	Saber ser
<p>Administra los procesos involucrados en la gestión de la calidad del software, con base en los factores de calidad acordados por el cliente.</p> <p>Gestiona las actividades vinculadas con la identificación, organización y control sobre las modificaciones de los artefactos desarrollados, utilizando diferentes herramientas.</p> <p>Utiliza herramientas informáticas de soporte en la administración de los procesos de calidad, desarrollo y configuración de software, para la optimización de las actividades de las fases de dicho proceso</p>	<p>Describe los modelos y estándares involucrados en el aseguramiento de la calidad de los procesos y métodos de la ingeniería de software, de acuerdo al contexto del problema.</p> <p>Describe los modelos, técnicas y herramientas utilizados en gestión de la configuración de un proyecto de software, de acuerdo con el marco de referencia propio de la ingeniería de software.</p> <p>Identifica las herramientas informáticas idóneas que brindan soporte a los procesos de administración de software, de acuerdo a la actividad y al desarrollo tecnológico.</p>	<p>Manifiesta un pensamiento crítico en su quehacer profesional con atención y respeto a las ideas de otros.</p> <p>Interactúa con otros profesionales en ambientes multidisciplinarios de manera respetuosa.</p> <p>Incorpora la creatividad en su práctica profesional de manera continua y permanente</p>

Respeto la confidencialidad de la información proporcionada por el cliente, de acuerdo a los principios éticos de la disciplina.

Demuestra una actitud objetiva y profesional en la toma de decisiones, ante diversas situaciones laborales.

Manifiesta autocontrol ante situaciones bajo presión, asumiendo responsablemente las tareas que le corresponden.

Mantiene una actitud perseverante en la resolución de problemas propios de su disciplina.

Enfrenta situaciones innovadoras en su ámbito profesional, de manera responsable y competente.

Manifiesta una actitud emprendedora en su práctica profesional que le permita el desarrollo de proyectos innovadores.

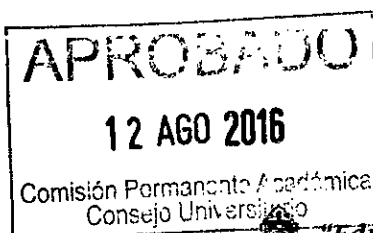
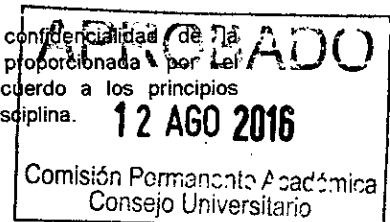


Tabla 18. Desagregado de saberes de la competencia de egreso Innovación en Ingeniería de Software.

Innovación en Ingeniería de Software		
Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, mediante fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.		
Saber hacer	Saber conocer	Saber ser
Elabora propuestas de mejora en los procesos de administración, desarrollo y mantenimiento de software en una organización, utilizando adecuadamente los fundamentos matemáticos.	Describe los fundamentos matemáticos en el ámbito de la innovación tecnológica, de acuerdo con el marco de referencia propio de la Ingeniería de Software.	Comunica sus ideas en forma oral y escrita mediante el uso correcto del idioma español.
Elabora propuestas de mejora en los procesos de administración, desarrollo y mantenimiento de software en una organización, utilizando adecuadamente los fundamentos ingenieriles.	Describe los fundamentos ingenieriles en el ámbito de la innovación tecnológica, de acuerdo con el marco de referencia propio de la Ingeniería de Software.	Incorpora las tecnologías de información de manera pertinente y responsable en todos los ámbitos del quehacer profesional.
Elabora propuestas de mejora en administración, desarrollo y mantenimiento de software en una organización, utilizando adecuadamente los fundamentos de las ciencias computacionales.	Describe los fundamentos de las ciencias computacionales en el ámbito de la innovación tecnológica, de acuerdo con el marco de referencia propio de la Ingeniería de Software.	Manifiesta interés en la actualización de sus conocimientos y habilidades en las áreas relacionadas con la Ingeniería de Software de forma autónoma y permanente.
Utiliza los modelos, técnicas y herramientas de la Ingeniería de Software, para la generación de evidencia en los procesos de software, mediante un enfoque disciplinado, sistemático y cuantificable.	Describe los modelos, técnicas y herramientas utilizados en los procesos de software, de acuerdo con el marco de referencia propio de la Ingeniería de Software.	Manifiesta objetividad, integridad y comportamiento ético en los diversos campos del quehacer profesional.
Aplica el método experimental en los procesos de Ingeniería de Software, para la innovación en las actividades de las fases de dicho proceso.		Manifiesta un pensamiento crítico en su quehacer profesional con atención y respeto a las ideas de otros.
		Interactúa con otros profesionales en ambientes multidisciplinares de manera respetuosa.
		Incorpora la creatividad en su práctica profesional de manera continua y permanente.
		Respeto la confidencialidad de la información proporcionada por el cliente, de acuerdo a los principios éticos de la disciplina.



Innovación en Ingeniería de Software

Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, mediante fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.

Saber hacer	Saber conocer	Saber ser
-------------	---------------	-----------

Demuestra una actitud objetiva y profesional en la toma de decisiones, ante diversas situaciones laborales.

Manifiesta autocontrol ante situaciones bajo presión, asumiendo responsablemente las tareas que le corresponden.

Mantiene una actitud perseverante en la resolución de problemas propios de su disciplina.

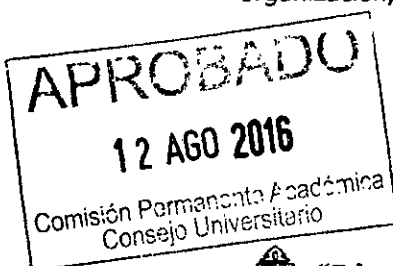
Enfrenta situaciones innovadoras en su ámbito profesional, de manera responsable y competente.

Manifiesta una actitud emprendedora en su práctica profesional que le permita el desarrollo de proyectos innovadores.

6.4. Competencias disciplinares

Las competencias disciplinares que representan la integración dinámica de conocimientos, habilidades, actitudes y valores que desarrolla el estudiante de Ingeniería de Software son:

1. Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
2. Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos, y del Cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
3. Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
4. Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



7. ESTRUCTURA CURRICULAR

7.1. Características del Plan de Estudios

El plan de estudios se encuentra organizado en tres tipos de actividades curriculares: Asignaturas (obligatorias, optativos y libres), Servicio Social y Prácticas Profesionales (ver Tabla 19). La unidad de medida para cada una de las actividades curriculares es el crédito. Los créditos de que consta el plan de estudios en su conjunto, se encuentran distribuidos de la siguiente manera:

Tabla 19. Plan de estudios por actividades.

Actividades	Créditos	Porcentaje (%)	Horas
Asignaturas Obligatorias	268		4288
Servicio Social	12	80.00	480
Prácticas Profesionales	8		320
Asignaturas Optativas ^a	54	15.00	864
Asignaturas Libres ^b	18	5.00	288
Total	360	100.00	6240

*Se espera que el alumno curse al menos el porcentaje de créditos establecido en el modelo.

Para cursar el plan de estudios con una dedicación de tiempo completo, se estima que en tiempo se requiere de nueve semestres (cuatro años y medio); sin embargo, si el alumno por diferentes situaciones necesita de mayor tiempo para concluir el plan de estudios, dispone de hasta catorce semestres (siete años) para concluirlo.

7.2. Organización del Plan de Estudios

Las asignaturas del plan de estudios se ofrecen bajo la modalidad mixta y de acuerdo a su forma de elección pueden ser:

Asignaturas obligatorias

Las asignaturas obligatorias son aquellas que resultan indispensables para el logro del perfil de egreso. El plan de estudios se encuentra integrado por 38 asignaturas obligatorias, que contribuyen al logro de las competencias genéricas, disciplinares y específicas.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

Facultad de Matemáticas



Tabla 20. Asignaturas obligatorias.

Clave	Nombre de la Asignatura	Horas Presenciales	Horas No Presenciales	Créditos
MATLIS168-API	Administración de Proyectos I	72	40	7
MATLIS169-AP2	Administración de Proyectos II	64	32	6
MATLIS161-AI	Álgebra Intermedia	64	64	8
MATLIS163-AL	Álgebra Lineal	72	56	8
MATLIS162-AS	Álgebra Superior	72	56	8
MATLIS161-A	Algoritmia	72	40	7
MATLIS163-AC	Arquitectura y Organización de Computadoras	72	56	8
MATLIS165-AS	Arquitecturas de Software	72	40	7
MATLIS166-AC	Aseguramiento de la Calidad del Software	72	40	7
MATLIS162-CD	Cálculo Diferencial	72	56	8
MATLIS163-CI	Cálculo Integral	72	56	8
MATLIS165-CS	Construcción de Software	72	40	7
MATLIS162-CM	Cultura Maya	48	48	6
MATLIS165-DAW	Desarrollo de Aplicaciones Web	72	40	7
MATLIS165-DBD	Diseño de Bases de Datos	72	40	7
MATLIS164-DS	Diseño de Software	72	40	7
MATLIS164-ED	Estructuras de Datos	72	56	8
MATLIS167-EIS	Experimentación en Ingeniería de Software	72	40	7
MATLIS161-FIS	Fundamentos de Ingeniería de Software	64	32	6
MATLIS161-GA	Geometría Analítica	64	64	8
MATLIS165-IE	Inferencia Estadística	72	40	7
MATLIS167-IT	Innovación Tecnológica	64	32	6
MATLIS166-IHC	Interacción Humano Computadora	72	40	7
MATLIS168-MS	Mantenimiento de Software	72	40	7
MATLIS162-MD	Matemáticas Discretas	72	40	7
MATLIS166-MS	Métricas de Software	72	40	7
MATLIS164-P	Probabilidad	72	40	7
MATLIS162-PE	Programación Estructurada	72	40	7
MATLIS163-POO	Programación Orientada a Objetos	72	56	8
MATLIS167-RC	Redes y Seguridad de computadoras	72	40	7
MATLIS166-RS	Requisitos de Software	72	40	7
MATLIS161-RSU	Responsabilidad Social Universitaria	48	48	6
MATLIS168-SD	Sistemas Distribuidos	72	40	7
MATLIS164-SE	Sistemas Operativos	72	40	7
MATLIS166-TE	Taller de Emprendedores	48	48	6
MATLIS163-TC	Teoría de la Computación	72	56	8

APROBADO

12 AGO 2015

Comisión Permanente Académica
Consejo Universitario



Clave	Nombre de la Asignatura	Horas Presenciales	Horas No Presenciales	Créditos
MATLIS164-TLP	Teoría de Lenguajes de Programación	64	32	6
MATLIS167-VVS	Verificación y Validación de Software	64	32	6

A continuación se presentan las asignaturas compartidas con los planes de estudio alineados al MEFI en la Facultad.

LA: Licenciatura en Actuaría.

LIC: Licenciatura en Ingeniería en Computación.

LEM: Licenciatura en Enseñanza de las Matemáticas.

LCC: Licenciatura en Ciencias de la Computación



Tabla 21. Asignaturas obligatorias compartidas con otros planes de estudio.

Clave	Asignatura	LCC	LIC	LEM	LA
MATLIS-161-AI	Álgebra Intermedia	X	X	X	X
MATLIS-163-AL	Álgebra Lineal	X	X		
MATLIS-162-AS	Algebra Superior	X			
MATLIS-161-A	Algoritmia	X			
MATLIS-163-AC	Arquitectura y Organización de Computadoras	X			
MATLIS-162-CD	Cálculo Diferencial	X	X		
MATLIS-163-CI	Cálculo Integral	X	X		
MATLIS-162-CM	Cultura Maya	X	X	X	X
MATLIS-161-GA	Geometría Analítica	X	X	X	X
MATLIS-162-MD	Matemáticas Discretas	X			
MATLIS-162-PE	Programación Estructurada	X			
MATLIS-163-POO	Programación Orientada a Objetos	X			
MATLIS-161-RSU	Responsabilidad Social Universitaria	X	X	X	X
MATLIS-168-SD	Sistemas Distribuidos	X			
MATLIS-163-TC	Teoría de la Computación	X			
MATLIS-164-TLP	Teoría de Lenguajes de Programación	X			

Asignaturas Optativas

Las asignaturas optativas se encuentran organizadas en áreas de concentración y en total tendrán un valor al menos de 54 créditos en el plan de estudios. Dichas asignaturas permiten al estudiante participar en la construcción de su perfil de egreso; podrán comenzar a cursarse una vez aprobados 180 créditos y su valor en créditos se determina en el programa de estudios. Un área de concentración estará integrado por bloques de asignaturas optativas que complementan la formación profesional brindando posibilidades de especialización en un área particular de la disciplina Ingeniería de Software— o de profundización en un dominio de interés.

APROBADO
 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario





Considerando los intereses individuales para la formación de los estudiantes, las asignaturas optativas podrían cursarse en otras dependencias de la UADY o incluso en otras instituciones de educación superior nacionales o extranjeras.

Asignaturas libres

Las asignaturas libres son aquellas que el estudiante debe cursar para complementar su formación integral; en total tendrán un valor al menos de 18 créditos en el plan de estudios, podrán ser cursadas a partir del ingreso del estudiante y su valor en créditos se determina en el programa de estudios.

Flexibilidad

El Modelo Educativo para la Formación Integral reconoce las diferencias individuales entre los estudiantes, por lo que la Licenciatura en Ingeniería de Software permitirá al estudiante configurar el conjunto de asignaturas a cursar en cada semestre, con base en:

- a) los requisitos académicos de cada asignatura,
- b) la oferta académica semestral,
- c) los intereses personales.

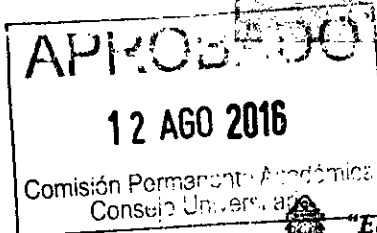
Es importante considerar que un alumno debe aprobar entre 37 y 43 créditos cada semestre para concluir el plan de estudios en 9 semestres.

Requisitos académicos

El desarrollo de las competencias de egreso se logra con la integración de las competencias genéricas, disciplinares y específicas declaradas en las asignaturas obligatorias. En este sentido, hay algunas asignaturas que contribuyen al desarrollo de competencias que el estudiante requiere movilizar para el desarrollo de competencias de mayor complejidad en otras asignaturas; por esta razón se han identificado requisitos académicos para cursar determinadas asignaturas del plan de estudios. La Tabla 22 presenta la lista de las asignaturas obligatorias con requisitos académicos (seriación) identificados.

Tabla 22. Requisitos académicos.

Asignatura	Antecedente Académico
	Cálculo Diferencial
	Probabilidad
	Inferencia Estadística
	Administración de Proyectos I



8. MALLA CURRICULAR

La malla curricular ilustra la secuencia sugerida en semestres de las asignaturas obligatorias del Plan de Estudios. La información proporcionada en esta malla se complementa con los requisitos académicos previamente descritos en este documento.

El número total de asignaturas libres y optativas estará en función del valor en créditos que la Dependencia, la UADY, u otra Institución, le haya asignado a dichas asignaturas.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

Facultad de Matemáticas



MALLA CURRICULAR
 CAMPUS DE CIENCIAS EXACTAS E INGENIERÍAS
 FACULTAD DE MATEMÁTICAS
 LICENCIATURA EN INGENIERÍA DE SOFTWARE
 MODALIDAD: MIXTA

PRIMER SEMESTRE	SEGUNDO SEMESTRE	TERCER SEMESTRE	CUARTO SEMESTRE	QUINTO SEMESTRE	SEXTO SEMESTRE	SÉPTIMO SEMESTRE	OCTAVO SEMESTRE	NOVENO SEMESTRE
MATLIS161-AS Álgebra Intermedia C HP HNP HT 8 64 64 128	MATLIS162-AA Álgebra Superior C HP HNP HT 8 72 56 128	MATLIS163-AL Álgebra Lineal C HP HNP HT 8 72 56 128	MATLIS164-P Probabilidad C HP HNP HT 7 72 40 112	MATLIS165-IE Inferencia Estadística C HP HNP HT 7 72 40 112	MATLIS166-MS Métricas de Software C HP HNP HT 7 72 40 112	MATLIS167-EIS Experimentación en Ingeniería de Software C HP HNP HT 7 72 40 112	MATLIS168-AP1 Administración de Proyectos I C HP HNP HT 7 72 40 112	MATLIS169-AP2 Administración de Proyectos II C HP HNP HT 6 64 32 96
MATLIS161-GA Geometría Analítica C HP HNP HT 8 64 64 128	MATLIS162-CD Cálculo Diferencial C HP HNP HT 8 72 56 128	MATLIS163-CI Cálculo Integral C HP HNP HT 8 72 56 128	MATLIS164-DS Diseño de Software C HP HNP HT 7 72 40 112	MATLIS165-AS Arquitecturas de Software C HP HNP HT 7 72 40 112	MATLIS166-AC Aseguramiento de la Calidad del Software C HP HNP HT 7 72 40 112	MATLIS167-VVS Verificación y Validación de Software C HP HNP HT 6 64 32 96	MATLIS168-MS Mantenimiento de Software C HP HNP HT 7 72 40 112	
MATLIS161-A Algoritmia C HP HNP HT 7 72 40 112	MATLIS162-PE Programación Estructurada C HP HNP HT 7 72 40 112	MATLIS163-POO Programación Orientada a Objetos C HP HNP HT 8 72 56 128	MATLIS164-ED Estructuras de Datos C HP HNP HT 8 72 56 128	MATLIS165-CS Construcción de Software C HP HNP HT 7 72 40 112	MATLIS166-RS Requisitos de Software C HP HNP HT 7 72 40 112	MATLIS167-RC Redes y Seguridad de Computadoras C HP HNP HT 7 72 40 112	MATLIS168-SD Sistemas Distribuidos C HP HNP HT 7 72 40 112	
MATLIS161-FIS Fundamentos de Ingeniería de Software C HP HNP HT 6 64 32 96	MATLIS162-MD Matemáticas Discretas C HP HNP HT 7 72 40 112	MATLIS163-TC Teoría de la Computación C HP HNP HT 8 72 56 128	MATLIS164-SO Sistemas Operativos C HP HNP HT 7 72 40 112	MATLIS165-DBD Diseño de Bases de Datos C HP HNP HT 7 72 40 112	MATLIS166-IHC Interacción Humano Computadora C HP HNP HT 7 72 40 112	MATLIS167-IT Innovación Tecnológica C HP HNP HT 6 64 32 96		
MATLIS161-RSU Responsabilidad Social Universitaria C HP HNP HT 6 48 48 96	MATLIS162-CM Cultura Maya C HP HNP HT 6 48 48 96	MATLIS163-AC Arquitectura y Organización de Computadoras C HP HNP HT 8 72 56 128	MATLIS164-TLP Teoría de Lenguajes de Programación C HP HNP HT 6 64 32 96	MATLIS165-DAW Desarrollo de Aplicaciones Web C HP HNP HT 7 72 40 112				



Niveles de Formación			
			Institucional
			Disciplinaria
			Especialidad

Ejemplo de clave: **MATLIS165-FIS**
 MAT Facultad de Matemáticas 16 Año de aprobación
 LIS Licenciatura en Ingeniería en Software 5 Semestre FIS Iniciales asignatura

Asignaturas Libres (HT: 288)		C
		18
	Prácticas Profesionales (Requisito: 180 Créditos - HT: 320)	8
	Taller de Emprendedores (HP: 48 - HNP: 48 - HT: 96)	6
	Asignaturas Optativas (HT: 864)	54
	Servicio Social (Requisito: 252 Créditos - HT: 480)	12

Semestre	Total	HP	HNP	HT
1	312	248	64	560
2	336	240	96	576
3	360	280	80	640
4	352	208	144	560
5	360	200	160	560
6	288	160	128	448
7	272	144	128	416
8	216	120	96	336
9	64	32	32	96

Total de horas presenciales: 2608
 Total de horas no presenciales: 1680
 Total de horas del plan de estudios: 6240
 Total de créditos del plan de estudios: 360

→ Indica requisito académico previo

	Créditos	Porcentaje	Horas
Obligatorias	268		4288
Prácticas Profesionales	8	80%	320
Servicio Social	12		480
Optativas	54	15%	864
Libres	18	5%	288
Total	360	100%	6240



12 ABO 2016
 Comisión Permanente Académica
 Consejo Universitario

8.1. Asignaturas optativas

Las asignaturas optativas se agrupan en áreas de concentración, las cuales promueven competencias que permiten profundizar en un área de especialización de la propia disciplina, o en su caso, en un dominio de aplicación.

Debido a la flexibilidad considerada para completar la formación del Ingeniero de Software mediante las asignaturas optativas, el alumno podrá cursar las asignaturas de las áreas de concentración ofrecidas por la dependencia de la UADY, o en su caso, podrá elaborar una propuesta de asignaturas ofrecidas por otra dependencia, e inclusive por otra institución de educación superior; en cualquiera de los casos, las asignaturas que integran las áreas de concentración a cursar por el alumno, deberán ser aprobadas por la Secretaría Académica.

Cabe mencionar, que los Cuerpos Académicos responsables de Líneas de Generación y/o Aplicación del Conocimiento (LGAC) vinculados con las áreas de conocimiento que configuran el perfil profesional del Ingeniero de Software, podrán proponer nuevas asignaturas optativas —que integren un área de concentración— como es el caso de las áreas de concentración denominadas: *Desarrollo de Aplicaciones Web, Videojuegos, Bases de Datos, Mejora de Procesos, Redes de Computadoras e Informática Educativa.*

Tabla 23. Asignaturas optativas.

Área de Concentración	Asignaturas
Desarrollo de Aplicaciones Web	Desarrollo de aplicaciones Web mediante tecnología J2EE Desarrollo de aplicaciones Web dinámicas con Ajax Ingeniería Web
Videojuegos	Gráficas por computadora Física para videojuegos I Introducción a los Videojuegos Inteligencia Artificial para video juegos
Bases de Datos	Bases de Datos Distribuidas Programación SQL Avanzado Minería de Datos Manejo de Datos No Estructurados
Mejora de Procesos	Proceso de Desarrollo de Software Personal (PSP) Proceso de Desarrollo de Software en Equipo (TSP) Modelos de Gestión de Servicios de Tecnologías de la Información Gestión de Servicios de Tecnologías de la Información con ITIL Metodologías Ágiles
Redes de Computadoras	Ingeniería de Software Empírico Conceptos y protocolos de enrutamiento Conmutación y conexión inalámbrica LAN Administración de Servidores UNIX
Informática Educativa	Introducción a la Informática Educativa Análisis de Aprendizaje Ingeniería de Software Educativo Ambientes de Aprendizaje Mejorados con Tecnología



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



9. ESQUEMA DE CONSISTENCIA

9.1. Matriz de consistencia de las asignaturas en relación con las competencias de egreso

Tabla 24. Matriz de consistencia por competencias de egreso.

Desarrollo de Software	Mantenimiento de Software	Administración de los Procesos de Software	Innovación en Ingeniería de Software
		Administración de Proyectos I Administración de Proyectos II	
		Álgebra Intermedia	
		Álgebra Lineal	
		Álgebra Superior	
		Algoritmia	
		Arquitectura y Organización de Computadoras	
Arquitecturas de Software			
Aseguramiento de la Calidad del Software			
		Cálculo Diferencial	
		Cálculo Integral	
Construcción de Software			
		Cultura Maya	
Desarrollo de Aplicaciones Web			
Diseño de Bases de Datos			
Diseño de Software			
Estructuras de Datos			
			Experimentación en Ingeniería de Software
		Fundamentos de Ingeniería de Software	

12 AGO 2016
 Comisión de Planeación
 C-1





Geometría Analítica

Inferencia Estadística

Desarrollo de Software	Mantenimiento de Software	Administración de los Procesos de Software	Innovación en Ingeniería de Software
-------------------------------	----------------------------------	---	---

Innovación Tecnológica

Interacción Humano Computadora

Mantenimiento de Software

Matemáticas Discretas

Métricas de Software

Probabilidad

Programación Estructurada

Programación Orientada a Objetos

Redes y Seguridad de Computadoras

Requisitos de Software

Responsabilidad Social Universitaria

Sistemas Distribuidos

Sistemas Operativos

Taller de Emprendedores

Teoría de la Computación

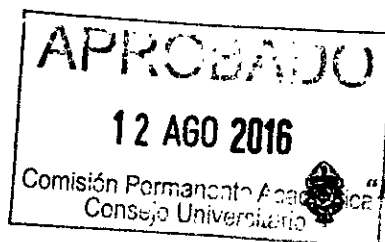
Teoría de Lenguajes de Programación

Verificación y Validación de Software

9.2. Esquema de consistencia por competencia de egreso

Tabla 25. Esquema de consistencia por competencias de egreso.

Competencia de egreso	Asignaturas	Competencias de las asignaturas
Desarrollo de Software	Álgebra Intermedia	Interpreta relaciones entre el lenguaje natural y simbólico en diversos contextos, empleando expresiones algebraicas.
	Álgebra Lineal	Aplica los principios y metodologías del Álgebra Lineal para el análisis, interpretación y solución de problemas, de manera clara, eficiente y fluida
	Álgebra Superior	Resuelve problemas matemáticos e interdisciplinarios aplicando los conceptos y propiedades de lógica, conjuntos, relaciones, funciones, cardinalidad y cálculo combinatorio.
	Algoritmia	Desarrolla algoritmos para la solución de problemas computacionales, en forma secuencial y lógica.
	Arquitectura y Organización de Computadoras	Implementa programas de cómputo usando lenguajes de bajo nivel, considerando la arquitectura de hardware.
	Arquitecturas de Software	Diseña una arquitectura para un sistema de software específico, considerando los atributos de calidad requeridos por la disciplina.
	Aseguramiento de la Calidad del Software	Aplica la teoría relacionada con el aseguramiento de la calidad del software en los procesos de desarrollo de software de manera continua, crítica y flexible.
	Cálculo Diferencial	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Diferencial de una variable y de herramientas computacionales de forma eficiente
	Cálculo Integral	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Integral de una variable y de herramientas computacionales de forma eficiente.
	Construcción de Software	Desarrolla aplicaciones utilizando prácticas y técnicas de construcción de software, considerando los estándares de codificación.



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Cultura Maya	Establece propuestas de solución a las problemáticas actuales de la sociedad, desde la realidad de la cultura maya, promoviendo la revaloración de la misma bajo los principios de multiculturalidad e interculturalidad.
	Desarrollo de Aplicaciones Web	Desarrolla aplicaciones Web con acceso a base de datos, de manera eficiente y creativa, utilizando las tecnologías más recientes en el área.
	Diseño de Bases de Datos	Diseña la estructura lógica de una base de datos, para la solución de un problema del mundo real, utilizando los modelos y técnicas apropiadas.
	Diseño de Software	Diseña productos de software utilizando técnicas y marcos de referencia de la disciplina que aseguren los atributos de calidad requeridos.
	Estructuras de Datos	Emplea las estructuras de datos lineales y no lineales en el desarrollo de software de aplicación de pequeña a mediana escala, para el manejo eficiente de la información.
	Fundamentos de Ingeniería de Software	Analiza las características principales de los métodos, técnicas, procedimientos y buenas prácticas, utilizados en los procesos de desarrollo y gestión del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.
	Geometría Analítica	Resuelve problemas en diversos contextos, a partir de la aplicación de propiedades trigonométricas y geométricas fundamentales de lugares geométricos en el plano, utilizando los métodos adecuados de manera eficiente.
	Interacción Humano Computadora	Desarrolla sistemas de software que proporcionen interacciones adecuadas para el usuario, considerando metodologías y principios básicos de diseño de interfaces en sistemas interactivos.
	Matemáticas Discretas	Representa procesos y estructuras computacionales mediante el uso de la teoría algebraica, recurrencias y grafos.



APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Métricas de Software	Evalúa proyectos de software de manera sistemática, disciplinada y cuantificable, aplicando métricas, estándares, metodologías y herramientas de medición.
	Probabilidad	Resuelve problemas de naturaleza aleatoria presentes en las diferentes áreas del conocimiento, utilizando la teoría básica de la probabilidad.
	Programación Estructurada	Desarrolla aplicaciones computacionales eficientes, fundamentado en las metodologías del paradigma de la programación estructurada.
	Programación Orientada a Objetos	Desarrolla aplicaciones computacionales eficientes, fundamentado en el paradigma de la programación orientada a objetos.
	Redes y Seguridad de Computadoras	Implementa mecanismos de seguridad en el procesamiento, almacenamiento e intercambio de información en los componentes de un sistema, interconectados a través de redes de comunicación.
	Requisitos de Software	Aplica los procesos de desarrollo y administración de requisitos de software en un problema del mundo real, utilizando los métodos, técnicas y herramientas de Ingeniería de Software.
	Responsabilidad Social Universitaria	Practica la responsabilidad social universitaria, en forma individual y colaborativa, como interrogación crítica de los impactos de la formación universitaria humanística y profesional mediante el uso de herramientas de investigación de RSU en la misma universidad, y evaluada a la luz del contexto sistémico económico, social y medioambiental global, a fin de querer ser una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad.
	Sistemas Distribuidos	Desarrolla aplicaciones eficientes en ambientes de procesamiento compartido, acordes con los modelos y arquitecturas de sistemas distribuidos.
	Sistemas Operativos	Analiza la operación de los diferentes componentes de los administradores de recursos de un sistema operativo,

APROBADO

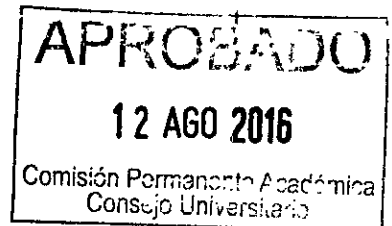
12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Taller de Emprendedores	considerando el marco teórico de las ciencias computacionales. Diseña un proyecto tecnológico orientado a la resolución de un problema del entorno, con responsabilidad, innovación, iniciativa y sentido de la ética.
	Teoría de la Computación	Diseña algoritmos computacionales que reconocen los elementos básicos de un lenguaje de programación, con base en los modelos formales de la teoría computacional.
	Teoría de Lenguajes de Programación Verificación y Validación de Software	Desarrolla software utilizando diversos paradigmas de programación, con base en la eficiencia y pertinencia del lenguaje. Implementa métodos y técnicas de verificación y validación, contribuyendo al aseguramiento de la calidad de un producto de software construido.

Competencia de egreso	Asignaturas	Competencias de las asignaturas
Mantenimiento de Software	Álgebra Intermedia	Interpreta relaciones entre el lenguaje natural y simbólico en diversos contextos, empleando expresiones algebraicas.
	Álgebra Lineal	Aplica los principios y metodologías del Álgebra Lineal para el análisis, interpretación y solución de problemas, de manera clara, eficiente y fluida
	Álgebra Superior	Resuelve problemas matemáticos e interdisciplinarios aplicando los conceptos y propiedades de lógica, conjuntos, relaciones, funciones, cardinalidad y cálculo combinatorio.
	Algoritmia	Desarrolla algoritmos para la solución de problemas computacionales, en forma secuencial y lógica.
	Arquitectura y Organización de Computadoras Arquitecturas de Software	Implementa programas de cómputo usando lenguajes de bajo nivel, considerando la arquitectura de hardware. Diseña una arquitectura para un sistema de software específico, considerando los atributos de calidad requeridos por la disciplina.



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Aseguramiento de la Calidad del Software	Aplica la teoría relacionada con el aseguramiento de la calidad del software en los procesos de desarrollo de software de manera continua, crítica y flexible.
	Cálculo Diferencial	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Diferencial de una variable y de herramientas computacionales de forma eficiente
	Cálculo Integral	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Integral de una variable y de herramientas computacionales de forma eficiente.
	Construcción de Software	Desarrolla aplicaciones utilizando prácticas y técnicas de construcción de software, considerando los estándares de codificación.
	Cultura Maya	Establece propuestas de solución a las problemáticas actuales de la sociedad, desde la realidad de la cultura maya, promoviendo la revaloración de la misma bajo los principios de multiculturalidad e interculturalidad.
	Desarrollo de Aplicaciones Web	Desarrolla aplicaciones Web con acceso a base de datos, de manera eficiente y creativa, utilizando las tecnologías más recientes en el área.
	Diseño de Bases de Datos	Diseña la estructura lógica de una base de datos, para la solución de un problema del mundo real, utilizando los modelos y técnicas apropiadas.
	Diseño de Software	Diseña productos de software utilizando técnicas y marcos de referencia de la disciplina que aseguren los atributos de calidad requeridos.
	Estructuras de Datos	Emplea las estructuras de datos lineales y no lineales en el desarrollo de software de aplicación de pequeña a mediana escala, para el manejo eficiente de la información.
	Fundamentos de Ingeniería de Software	Analiza las características principales de los métodos, técnicas, procedimientos y buenas prácticas, utilizados en los procesos de desarrollo y gestión del software, de

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Geometría Analítica	acuerdo con el cuerpo de conocimientos reconocido por la disciplina. Resuelve problemas en diversos contextos, a partir de la aplicación de propiedades trigonométricas y geométricas fundamentales de lugares geométricos en el plano, utilizando los métodos adecuados de manera eficiente.
	Interacción Humano Computadora	Desarrolla sistemas de software que proporcionen interacciones adecuadas para el usuario, considerando metodologías y principios básicos de diseño de interfaces en sistemas interactivos.
	Mantenimiento de Software	Realiza el mantenimiento y actualización de sistemas de software, utilizando metodologías que aseguren altos estándares de calidad.
	Matemáticas Discretas	Representa procesos y estructuras computacionales mediante el uso de la teoría algebraica, recurrencias y grafos.
	Métricas de Software	Evalúa proyectos de software de manera sistemática, disciplinada y cuantificable, aplicando métricas, estándares, metodologías y herramientas de medición.
	Probabilidad	Resuelve problemas de naturaleza aleatoria presentes en las diferentes áreas del conocimiento, utilizando la teoría básica de la probabilidad.
	Programación Estructurada	Desarrolla aplicaciones computacionales eficientes, fundamentado en las metodologías del paradigma de la programación estructurada.
	Programación Orientada a Objetos	Desarrolla aplicaciones computacionales eficientes, fundamentado en el paradigma de la programación orientada a objetos.
	Redes y Seguridad de Computadoras	Implementa mecanismos de seguridad en el procesamiento, almacenamiento e intercambio de información en los componentes de un sistema, interconectados a través de redes de comunicación.
	Requisitos de Software	Aplica los procesos de desarrollo y de administración de requisitos de software en un problema del mundo real, utilizando los



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Responsabilidad Social Universitaria	<p>métodos, técnicas y herramientas de ingeniería de software.</p> <p>Practica la responsabilidad social universitaria, en forma individual y colaborativa, como interrogación crítica de los impactos de la formación universitaria humanística y profesional mediante el uso de herramientas de investigación de RSU en la misma universidad, y evaluada a la luz del contexto sistémico económico, social y medioambiental global, a fin de querer ser una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad.</p>
	Sistemas Distribuidos	Desarrolla aplicaciones eficientes en ambientes de procesamiento compartido, acordes con los modelos y arquitecturas de sistemas distribuidos.
	Sistemas Operativos	Analiza la operación de los diferentes componentes de los administradores de recursos de un sistema operativo, considerando el marco teórico de las ciencias computacionales.
	Taller de Emprendedores	Diseña un proyecto tecnológico orientado a la resolución de un problema del entorno, con responsabilidad, innovación, iniciativa y sentido de la ética.
	Teoría de la Computación	Diseña algoritmos computacionales que reconocen los elementos básicos de un lenguaje de programación, con base en los modelos formales de la teoría computacional.
	Teoría de Lenguajes de Programación Verificación y Validación de Software	<p>Desarrolla software utilizando diversos paradigmas de programación, con base en la eficiencia y pertinencia del lenguaje.</p> <p>Implementa métodos y técnicas de verificación y validación, contribuyendo al aseguramiento de la calidad de un producto de software construido.</p>

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
Administración de los Procesos de Software	Administración de Proyectos I	Elabora el plan para la administración de un proyecto de software, mediante un enfoque sistémico, disciplinado y cuantificable.
	Administración de Proyectos II	Implementa el plan para la administración de un proyecto de software, utilizando los métodos, técnicas y herramientas apropiadas de administración de proyectos.
	Álgebra Intermedia	Interpreta relaciones entre el lenguaje natural y simbólico en diversos contextos, empleando expresiones algebraicas.
	Álgebra Lineal	Aplica los principios y metodologías del Álgebra Lineal para el análisis, interpretación y solución de problemas, de manera clara, eficiente y fluida
	Álgebra Superior	Resuelve problemas matemáticos e interdisciplinarios aplicando los conceptos y propiedades de lógica, conjuntos, relaciones, funciones, cardinalidad y cálculo combinatorio.
	Algoritmia	Desarrolla algoritmos para la solución de problemas computacionales, en forma secuencial y lógica.
	Arquitectura y Organización de Computadoras	Implementa programas de cómputo usando lenguajes de bajo nivel, considerando la arquitectura de hardware.
	Aseguramiento de la Calidad del Software	Aplica la teoría relacionada con el aseguramiento de la calidad del software en los procesos de desarrollo de software de manera continua, crítica y flexible.
	Cálculo Diferencial	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Diferencial de una variable y de herramientas computacionales de forma eficiente
	Cálculo Integral	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Integral de una variable y de herramientas computacionales de forma eficiente.
Cultura Maya	Establece propuestas de solución a las problemáticas actuales de la sociedad, desde la realidad de la cultura maya, promoviendo la revaloración de la misma bajo los principios de multiculturalidad e interculturalidad.	



APROBADO
 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Fundamentos de Ingeniería de Software	Analiza las características principales de los métodos, técnicas, procedimientos y buenas prácticas, utilizados en los procesos de desarrollo y gestión del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.
	Geometría Analítica	Resuelve problemas en diversos contextos, a partir de la aplicación de propiedades trigonométricas y geométricas fundamentales de lugares geométricos en el plano, utilizando los métodos adecuados de manera eficiente.
	Matemáticas Discretas	Representa procesos y estructuras computacionales mediante el uso de la teoría algebraica, recurrencias y grafos.
	Métricas de Software	Evalúa proyectos de software de manera sistemática, disciplinada y cuantificable, aplicando métricas, estándares, metodologías y herramientas de medición.
	Probabilidad	Resuelve problemas de naturaleza aleatoria presentes en las diferentes áreas del conocimiento, utilizando la teoría básica de la probabilidad.
	Programación Estructurada	Desarrolla aplicaciones computacionales eficientes, fundamentado en las metodologías del paradigma de la programación estructurada.
	Programación Orientada a Objetos	Desarrolla aplicaciones computacionales eficientes, fundamentado en el paradigma de la programación orientada a objetos.
	Redes y Seguridad de Computadoras	Implementa mecanismos de seguridad en el procesamiento, almacenamiento e intercambio de información en los componentes de un sistema, interconectados a través de redes de comunicación.
	Responsabilidad Social Universitaria	Practica la responsabilidad social universitaria, en forma individual y colaborativa, como interrogación crítica de los impactos de la formación universitaria humanística y profesional mediante el uso de herramientas de investigación de RSU en la misma universidad, y evaluada a la luz del contexto sistémico económico, social y medioambiental global, a fin de querer ser

APPROVED
 12 AGO 2016
 Comisión Permanente Académica
 Consejo de Facultad



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Taller de Emprendedores	una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad. Diseña un proyecto tecnológico orientado a la resolución de un problema del entorno, con responsabilidad, innovación, iniciativa y sentido de la ética.
	Teoría de la Computación	Diseña algoritmos computacionales que reconocen los elementos básicos de un lenguaje de programación, con base en los modelos formales de la teoría computacional.
	Verificación y Validación de Software	Implementa métodos y técnicas de verificación y validación, contribuyendo al aseguramiento de la calidad de un producto de software construido.

Competencia de egreso	Asignaturas	Competencias de las asignaturas
Innovación en Ingeniería de Software	Álgebra Intermedia	Interpreta relaciones entre el lenguaje natural y simbólico en diversos contextos, empleando expresiones algebraicas.
	Álgebra Lineal	Aplica los principios y metodologías del Álgebra Lineal para el análisis, interpretación y solución de problemas, de manera clara, eficiente y fluida
	Álgebra Superior	Resuelve problemas matemáticos e interdisciplinarios aplicando los conceptos y propiedades de lógica, conjuntos, relaciones, funciones, cardinalidad y cálculo combinatorio.
	Algoritmia	Desarrolla algoritmos para la solución de problemas computacionales, en forma secuencial y lógica.
	Arquitectura y Organización de Computadoras Cálculo Diferencial	Implementa programas de cómputo usando lenguajes de bajo nivel, considerando la arquitectura de hardware. Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Diferencial de una variable y de herramientas computacionales de forma eficiente



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Cálculo Integral	Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Integral de una variable y de herramientas computacionales de forma eficiente.
	Cultura Maya	Establece propuestas de solución a las problemáticas actuales de la sociedad, desde la realidad de la cultura maya, promoviendo la revaloración de la misma bajo los principios de multiculturalidad e interculturalidad.
	Experimentación en Ingeniería de Software	Aplica el proceso experimental en estudios empíricos sobre fenómenos de la Ingeniería de Software, utilizando los métodos, técnicas y herramientas apropiadas del diseño y análisis de experimentos.
	Fundamentos de Ingeniería de Software	Analiza las características principales de los métodos, técnicas, procedimientos y buenas prácticas, utilizados en los procesos de desarrollo y gestión del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.
	Geometría Analítica	Resuelve problemas en diversos contextos, a partir de la aplicación de propiedades trigonométricas y geométricas fundamentales de lugares geométricos en el plano, utilizando los métodos adecuados de manera eficiente.
	Inferencia Estadística	Aplica la metodología de estimación y pruebas de hipótesis para uno o dos parámetros en problemas prácticos, relacionados con la investigación en Ingeniería de Software.
	Innovación Tecnológica	Realiza proyectos de Ingeniería de Software, promoviendo la innovación tecnológica y el desarrollo sostenible,
	Matemáticas Discretas	Representa procesos y estructuras computacionales mediante el uso de la teoría algebraica, recurrencias y grafos.
	Métricas de Software	Evalúa proyectos de software de manera sistemática, disciplinada y cuantificable, aplicando métricas, estándares, metodologías y herramientas de medición.
	Probabilidad	Resuelve problemas de naturaleza aleatoria presentes en las diferentes áreas del

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Competencia de egreso	Asignaturas	Competencias de las asignaturas
	Programación Estructurada	conocimiento, utilizando la teoría básica de la probabilidad. Desarrolla aplicaciones computacionales eficientes, fundamentado en las metodologías del paradigma de la programación estructurada.
	Programación Orientada a Objetos Redes y Seguridad de Computadoras	Desarrolla aplicaciones computacionales eficientes, fundamentado en el paradigma de la programación orientada a objetos. Implementa mecanismos de seguridad en el procesamiento, almacenamiento e intercambio de información en los componentes de un sistema, interconectados a través de redes de comunicación.
	Responsabilidad Social Universitaria	Practica la responsabilidad social universitaria, en forma individual y colaborativa, como interrogación crítica de los impactos de la formación universitaria humanística y profesional mediante el uso de herramientas de investigación de RSU en la misma universidad, y evaluada a la luz del contexto sistémico económico, social y medioambiental global, a fin de querer ser una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad.
	Taller de Emprendedores	Diseña un proyecto tecnológico orientado a la resolución de un problema del entorno, con responsabilidad, innovación, iniciativa y sentido de la ética.
	Teoría de la Computación	Diseña algoritmos computacionales que reconocen los elementos básicos de un lenguaje de programación, con base en los modelos formales de la teoría computacional.



APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



9.3. Matriz de las competencias genéricas por asignatura.

Tabla 26. Matriz de competencias genéricas.

Competencias genéricas	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Administración de Proyectos I	X									X	X			X	X	X							
Administración de Proyectos II	X		X						X	X	X			X									
Álgebra Intermedia				X		X	X	X							X				X				
Álgebra Lineal	X		X				X	X											X				
Álgebra Superior					X			X					X		X				X				
Algoritmos			X			X		X				X			X		X						
Apertura y Organización de Computadores		X	X				X						X	X	X								
Arquitecturas de Software	X									X	X						X						
Aseguramiento de la Calidad del Software	X					X					X					X	X	X					
Cálculo Diferencial	X		X	X			X	X								X							
Cálculo Integral	X		X	X			X	X								X							
Construcción de Software	X			X				X								X	X	X					
Cultura Maya	X		X	X	X						X			X				X			X	X	X
Desarrollo de Aplicaciones Web	X									X	X						X		X				
Diseño de Bases de Datos	X					X	X	X						X		X							
Diseño de Software	X		X								X					X			X				
Estructuras de Datos			X			X	X	X			X							X	X				
Experimentación en Ingeniería de Software	X			X	X		X							X	X								
Fundamentos de Ingeniería de Software	X		X	X							X					X							
Geometría Analítica	X			X	X	X		X										X					

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Inferencia Estadística		X		X		X		X		X				
Innovación Tecnológica	X		X			X	X	X			X			
Interacción Humano Computadora	X		X	X				X		X				X
Mantenimiento de Software	X							X	X				X	X
Matemáticas Discretas		X			X	X				X				X
Métricas de Software	X	X	X	X				X					X	
Probabilidad			X	X	X	X		X						
Programación Estructurada				X	X			X	X	X				X
Programación Orientada a Objetos	X			X				X	X					X
Redes y Seguridad de Computadoras	X	X		X				X					X	X
Requisitos de Software	X			X	X			X					X	X
Responsabilidad Social Universitaria			X		X	X	X							X X
Sistemas Distribuidos		X		X				X					X	X X
Sistemas Operativos	X				X	X				X			X	
Taller de Emprendedores	X		X		X	X							X	X
Teoría de la Computación		X	X		X			X		X				X
Teoría de Lenguajes de Programación	X	X		X	X					X				X
Verificación y Validación de Software	X	X	X	X									X	X



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



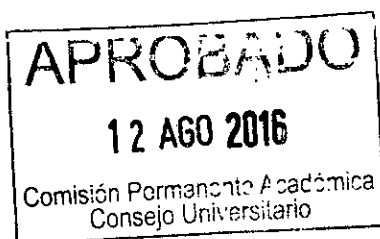
10. PROGRAMAS DE ESTUDIO

Los programas de estudio permiten identificar las competencias genéricas, disciplinarias y específicas que se van a desarrollar con cada asignatura, así como la competencia de egreso a la que contribuyen; dichos programas servirán de insumo para que el profesorado elabore la planeación didáctica de cada una de las asignaturas.

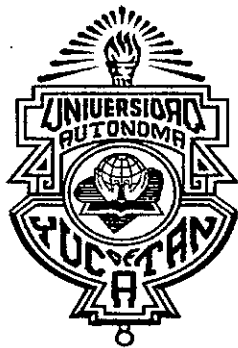
De acuerdo con la Guía para la Elaboración de Planes y Programas de Estudio del Programa Institucional de Habilitación en el MEFI, los elementos incorporados a cada programa son:

1. Datos generales de identificación.
2. Intencionalidad formativa de la asignatura.
3. Relación con otras asignaturas para el logro de las competencias de egreso.
4. Competencia de la asignatura.
5. Competencias genéricas, disciplinares y específicas a las que contribuye la asignatura.
6. Contenidos esenciales para el desarrollo de la competencia de la asignatura.
7. Estrategias de enseñanza y aprendizaje.
8. Estrategias generales de evaluación.
9. Referencias.
10. Perfil deseable del profesor.

Los programas de estudio de las treinta y siete asignaturas obligatorias se presentan a continuación.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Administración de Proyectos I

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Administración de Proyectos I		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Octavo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos previos	Ninguno		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Esta asignatura permitirá al estudiante de la Licenciatura en Ingeniería de Software, planear el desarrollo de proyectos de software mediante un enfoque sistémico, disciplinado y cuantificable.

El propósito de Administración de Proyectos I es aportar los conceptos metodológicos, las técnicas y las herramientas para iniciar, planear, ejecutar, controlar y cerrar un proyecto de desarrollo de software.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Administración de Proyectos I se relaciona con Administración de Proyectos II, Métricas de Software y Aseguramiento de la Calidad de Software; ya que contribuyen a la competencia de egreso:

- Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.

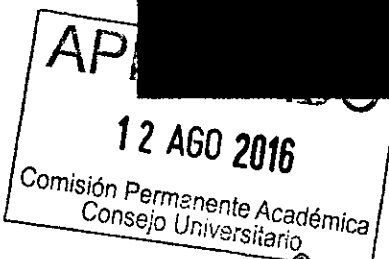
4. COMPETENCIA DE LA ASIGNATURA

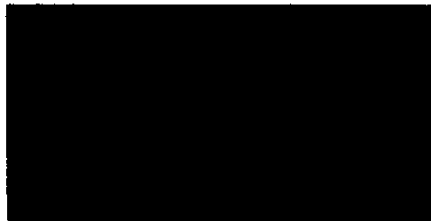
Elabora el plan para la administración de un proyecto de software, mediante un enfoque sistémico, disciplinado y cuantificable.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera





transparente y ética.

- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.



Disciplinares

- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

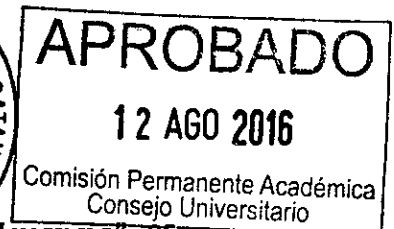


Específicas

- Determina el tamaño, esfuerzo, duración y costo de un producto de software, aplicando distintos métodos y técnicas de estimación.
- Evalúa la viabilidad de un proyecto de software, considerando sus análisis técnico, económico y operacional.
- Planifica las actividades asociadas a las fases de desarrollo del proyecto de software, considerando las características del producto, del equipo de desarrollo y del cliente.
- Elabora un Plan de Desarrollo de Software, empleando métodos, técnicas y herramientas de administración de proyectos.
- Desarrolla un Plan de Administración de la Configuración de Software, utilizando métodos, técnicas y herramientas de administración de la configuración.
- Desarrolla un Plan de Administración de Riesgos, mediante el uso de métodos, técnicas y herramientas de administración de riesgos.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción a la Administración de Proyectos de Software
- Enfoques para la Administración de Proyectos de software
- Estimación de recursos
- Evaluación de Proyectos de Software
- Planificación de actividades
- Administración de la Configuración del Software



- Administración de Riesgos

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Aprendizaje basado en problemas
- Juego de roles
- Aprendizaje cooperativo

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso – 70%

- Debates
- Resolución de casos
- Resolución de situaciones problema
- Elaboración de reportes

Evaluación de producto – 30%

- Desarrollo de proyectos

9. REFERENCIAS

1. Ahmed, A. (2012). *Software Project Management: A Process-Driven Approach*. USA: CRC Press.
2. Brechner, E. (2015). *Agile Project Management with Kanban*. USA: Microsoft Press.
3. Futrell, R.T., Shafer, D. F., & Shafer L. I. (2002). *Quality software Project Management*. USA: Prentice Hall PTR.
4. Larson, E.W. y Gray, C. F. (2014). *Project Management: The Managerial Process* (6ª ed.). USA: Mc Graw Hill Education.
5. McManus, J. (2011). *Risk Management in Software Development Projects*. USA: Routledge.
6. Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge* (5ª ed.). USA: Project Management Institute.
7. Rubin K. S. (2013). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. USA: Addison-Wesley.

APROBADO

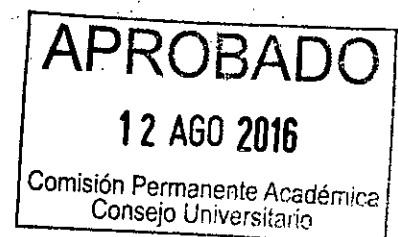
12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Administración de Proyectos II

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Administración de Proyectos II		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Noveno semestre		
e. Duración total en horas	96	Horas presenciales 64	Horas no presenciales 32
f. Créditos	6		
g. Requisitos académicos previos	Administración de Proyectos I		

APR
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de la Administración de Proyectos II es importante para la formación de los estudiantes de Ingeniería de Software, ya que les permitirá implementar estrategias para el desarrollo de proyectos de software; considerando la administración del tiempo y de los recursos humanos así como los acuerdos de propiedad intelectual. Lo anterior, para resolver problemas relacionados con la administración de proyectos de software y de acuerdo a los estándares internacionales.

El propósito de esta asignatura es aportar los métodos, técnicas y herramientas relacionadas con la Administración de Proyectos, para su aplicación en la ejecución, mantenimiento y cierre de los mismos.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Administración de Proyectos II se relaciona con las asignaturas Administración de Proyectos I, Construcción de Software, Mantenimiento de Software, Innovación Tecnológica, Responsabilidad Social Universitaria y Taller de Emprendedores; ya que contribuyen al logro de la competencia de egreso:

- Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.

4. COMPETENCIA DE LA ASIGNATURA

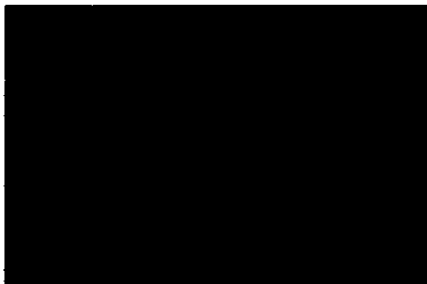
Implementa el plan para la administración de un proyecto de software, utilizando los métodos, técnicas y herramientas apropiadas para el desarrollo y cierre del mismo.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

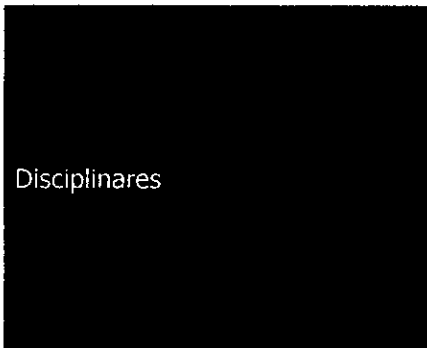
Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Interviene con iniciativa y espíritu emprendedor en su ejercicio profesional y personal de forma autónoma y permanente.





- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas a nivel superior.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



- Implementa las actividades de seguimiento, control, administración de riesgos y administración de la configuración durante el desarrollo y cierre de un proyecto de software.
- Diseña estrategias de control de cambios en proyectos de desarrollo de software, para la mejora de las etapas planeación, ejecución, mantenimiento y cierre de los mismos.
- Explica con precisión las características de un contrato legal apropiado que formalice el proceso de desarrollo de software, con apego a la ética profesional.
- Aplica las técnicas para la administración del personal involucrado en el desarrollo de un proyecto de software, utilizando de manera eficiente los recursos disponibles.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Administración de los procesos de desarrollo del proyecto
- Gestión del tiempo de las actividades de desarrollo del proyecto
- Aspectos legales y tipos de contratos

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



- Administración de los recursos humanos
- Seguimiento y control de riesgos
- Administración del cierre de un proyecto

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Aprendizaje autónomo y reflexivo
- Aprendizaje cooperativo
- Aprendizaje mediado por las TIC

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

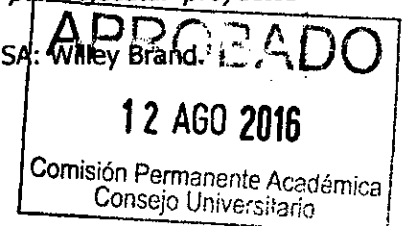
- Pruebas de desempeño
- Desarrollo de proyectos
- Resolución de casos
- Elaboración de reportes
- Resolución de situaciones problema

Evaluación de producto – 30%

- Proyecto integrador

9. REFERENCIAS

1. Alaimo, D. M. y Salías, M. (2015). *Proyectos Ágiles con Scrum: Flexibilidad, aprendizaje, innovación y colaboración en contextos complejos*. Buenos Aires: Kleer.
2. Kerzner, H. R. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. (11th. ed.). New York: Wiley.
3. Lledo, P. (2015). *Administración de Proyectos: El ABC para un Director de proyectos exitoso*. USA: Wiley.
4. Martel, A. (2016). *Gestión práctica de proyectos con Scrum*. (3^a. ed.). México: Antonio Martel Rodriguez (Editor).
5. Olvera, H. (2014). *El Director de Proyectos Práctico: Una receta para ejecutar proyectos exitosos*. México: Amazon Digital Services, Inc.
6. Portny, S. E. (2013). *Project Management For Dummies*. (4th ed.). USA: Wiley Brand.



7. Project Management Institute. (2012). *A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide*. (5th. ed.). USA: Project Management Institute.
8. Rivera, F. y Hernández, G. (2010) *Administración de proyectos: Guía para el aprendizaje*. México: Prentice Hall.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado.
- Mínimo tres años de experiencia profesional.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que se va a impartir.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE

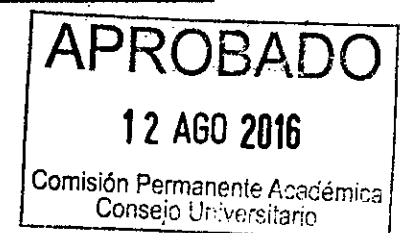


Álgebra Intermedia

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Álgebra Intermedia		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Primer semestre		
e. Duración total en horas	128	Horas presenciales 64	Horas no presenciales 64
f. Créditos	8		



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de Álgebra Intermedia contribuye a la formación del estudiante de la Licenciatura en Ingeniería de Software, pues aporta conceptos, ideas, algoritmos, técnicas y métodos de Álgebra con enfoque calculativo, para que posteriormente pueda estudiar con más soltura conceptos abstractos de Álgebra.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Álgebra Intermedia se relaciona con las asignaturas: Álgebra Superior, Álgebra Lineal, Matemáticas Discretas, Cálculo Diferencial, Cálculo Integral y Cálculo Vectorial; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Interpreta relaciones entre el lenguaje natural y simbólico en diversos contextos, empleando expresiones algebraicas.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia;
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente;
- Trabaja bajo presión de manera eficaz y eficientemente.
- Toma decisiones en su práctica profesional y personal de manera responsable
- Desarrolla su pensamiento en intervenciones profesionales y en su vida personal con pertinencia.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.

Específicas

- Realiza las operaciones matemáticas con polinomios de manera adecuada.
- Calcula el resultado de los productos notables de manera clara.
- Descompone expresiones algebraicas en factores hasta su mínima expresión.
- Resuelve ecuaciones y sistemas de ecuaciones polinómicas de primer y segundo grado a través de métodos algebraicos.
- Realiza las operaciones básicas en expresiones racionales de manera adecuada.
- Realiza las operaciones básicas en los números complejos, considerando sus propiedades.
- Representa problemas matemáticos en forma algebraica, mediante el uso de ecuaciones exponenciales y logarítmicas en forma adecuada.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Operaciones con polinomios
- Productos notables y factorización
- Expresiones racionales
- Números complejos
- Ecuaciones cuadráticas
- Sistemas de dos ecuaciones cuadráticas o lineales con dos incógnitas
- Ecuaciones exponenciales y logarítmicas

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Prácticas supervisadas
- Aprendizaje colaborativo
- Debates
- Aprendizaje autónomo y reflexivo



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



- Lluvia de ideas

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso -60%

- Pruebas de desempeño
- Prácticas supervisadas

Evaluación de producto -40%

- Elaboración de proyecto
- Pruebas de desempeño

9. REFERENCIAS

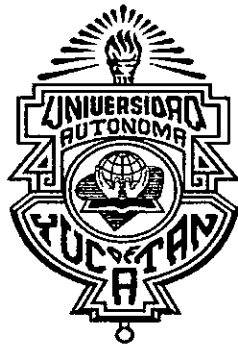
1. Andreescu, T. & Andrica, D. (2014). *Complex Numbers from A to... Z*. Boston MA: Birkhäuser.
2. Churchill, R. V. & Brown, J. W. (2008). *Complex Variables and applications*. (7th ed.). McGraw-Hill.
3. De Oteyza, E. (2010). *Álgebra*. México: Prentice Hall.
4. Fuller, G. (2000). *Álgebra Elemental*. México: Compañía Editorial Continental. (Clásico)
5. Gustafson, D. (2008). *Álgebra Intermedia*. México: Internacional Thomson Editores.
6. Kaufmann, J. y Schwitters, K. (2011). *Álgebra Intermedia*. México: Internacional Thomson Editores.
7. Lehmann, C. H. (2011). *Álgebra*. México: Limusa.
8. Pinzón, J. y Rosas, C. (2009). *Temas de Álgebra*. México: Ediciones UADY.
9. Swokowski, E. y Cole, J. (2009). *Álgebra y Trigonometría con Geometría Analítica*. México: Cengage Learning Editores S.A. de C.V.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o carrera afín, preferentemente con posgrado en Matemáticas.
- Mínimo dos años de experiencia profesional.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Álgebra Lineal

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Álgebra Lineal		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Tercer semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La comprensión y el desarrollo de la ciencia y la tecnología requiere de herramientas matemáticas como el Álgebra Lineal; esta asignatura provee conceptos, ideas, algoritmos, técnicas y métodos indispensables para la resolución de problemas relacionados con la ciencia y la ingeniería.

Álgebra Lineal proporciona al estudiante las herramientas matemáticas, para analizar y encontrar la solución de los sistemas de ecuaciones lineales que comúnmente aparecen en la modelación de fenómenos en las ciencias y la ingeniería.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Álgebra Lineal se relaciona con las asignaturas Álgebra Intermedia, Estructuras de Datos, Probabilidad e Inferencia Estadística; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Aplica los principios y metodologías del Álgebra Lineal para el análisis, interpretación y solución de problemas de manera clara, eficiente y fluida.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal de forma autónoma y permanente.
- Desarrolla su pensamiento, en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja bajo presión de manera eficaz y eficiente.
- Se comunica en español en forma oral y escrita en sus

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



[Redacted]

Disciplinares

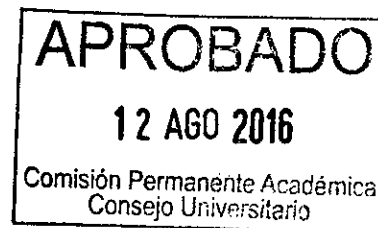
Específicas

intervenciones profesionales y en su vida personal utilizando correctamente el idioma.

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Aplica las matrices, sus propiedades y operaciones para la solución de problemas matemáticos.
- Calcula correctamente la inversa de una matriz, utilizando el determinante y sus propiedades.
- Resuelve sistemas de ecuaciones lineales resultantes de la modelación de problemas en diferentes áreas, utilizando los métodos de Gauss, Gauss-Jordan y matriz inversa.
- Utiliza correctamente los conceptos de espacio vectorial y álgebra de vectores, para la construcción de la base de un espacio y la determinación de su dimensión.
- Aplica correctamente el producto escalar de un espacio vectorial en la determinación de la ortogonalidad de dos vectores y la proyección ortogonal de un vector en otro.
- Utiliza correctamente las transformaciones lineales y sus propiedades en la representación de matrices de reflexión, dilatación, contracción y rotación.
- Utiliza correctamente valores y vectores propios en la determinación de matrices diagonalizables.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Matrices y vectores
- Determinantes y sus propiedades
- Resolución de sistemas de ecuaciones lineales
- Espacios vectoriales sobre los números reales
- Producto escalar y proyecciones ortogonales
- Transformaciones lineales y cambios de base
- Valores y vectores propios



7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Exposición
- Debates
- Lluvia de ideas
- Búsqueda de información

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Pruebas de desempeño
- Ensayos

Evaluación de producto – 40%

- Elaboración de proyectos

9. REFERENCIAS

1. Couch, LW. (2012). *Digital and Analog Communication Systems*. (8th ed.). Prentice Hall.
2. Howard, A. (2010). *Elementary Linear Álgebra* (10th ed.). Wiley.
3. Hill, D.R. (2003). *Linear Álgebra labs with Matlab*. (3rd ed.). Pearson. (Clásico)
4. León, S. (2009). *Linear Álgebra with Applications*. (8th ed.). Pearson.
5. Meye, C.D. (2001). *Matrix Analysis and Applied Linear Álgebra*. USA: SIAM. (Clásico)
6. Larson, R., & Falvo, D. (2009). *Elementary Linear Álgebra Enhanced Edition*. (6th ed.). Brooks Cole.
7. Grossman, S. (1994). *Elementary Linear Álgebra*. (5th ed.). Brooks Cole. (Clásico)
8. Strang, G. (2009). *Introduction to Linear Álgebra*. (4th ed.). Wellesley Cambridge Press.
9. Strang, G. (2005). *Linear Álgebra and its Applications*. (4th ed.). Brooks Cole. (Clásico)

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o carrera afín, preferentemente con posgrado en Matemáticas.
- Mínimo un año de experiencia profesional e investigación en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE

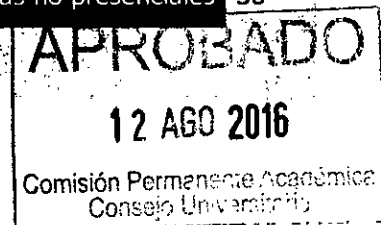


Álgebra Superior

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Álgebra Superior		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Segundo semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de Álgebra Superior es importante para la formación de los estudiantes de la Licenciatura en Ingeniería de Software, ya que les permitirá adquirir las herramientas algebraicas básicas para aplicar diferentes formas de razonamiento al reconocer, definir y resolver problemas, contribuyendo a la comprensión y utilización del lenguaje matemático.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Álgebra Superior se relaciona con las asignaturas Álgebra Lineal, Matemáticas Discretas, Cálculo Diferencial, Cálculo Integral, Cálculo Vectorial, Probabilidad e Inferencia Estadística; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Resuelve problemas matemáticos e interdisciplinarios aplicando los conceptos y propiedades de lógica, conjuntos, relaciones, funciones, cardinalidad y cálculo combinatorio.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Utiliza habilidades de investigación en sus intervenciones profesionales con rigor científico.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica y reflexiva.
- Responde a nuevas situaciones en su práctica profesional y en su vida personal con flexibilidad en contextos locales, nacionales e internacionales.
- Toma decisiones en su práctica profesional y personal de manera responsable.
- Trabaja bajo presión de manera eficaz y eficiente.

Disciplinares

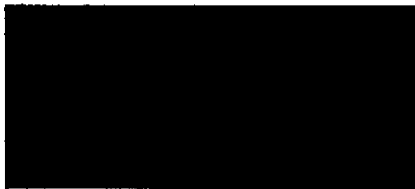
- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



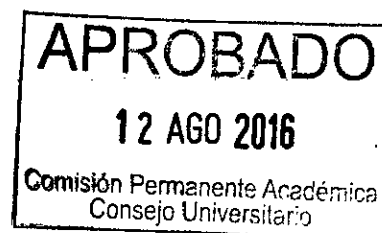


Específicas

- se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Determina la negación de una proposición de manera adecuada.
- Transforma proposiciones del tipo "p implica q" y "p si y sólo si q" a proposiciones equivalentes.
- Determina el valor de verdad de proposiciones con cuantificadores y conectores lógicos, con base en las reglas de la lógica matemática.
- Demuestra la veracidad de proposiciones matemáticas, aplicando los diferentes métodos de demostración.
- Identifica los elementos que pertenecen a un conjunto, con base en su definición.
- Determina conjuntos a partir de otros, utilizando las operaciones de unión e intersección, diferencia, producto cartesiano de dos conjuntos y conjunto potencia.
- Identifica las relaciones de equivalencia entre dos conjuntos, con base en las propiedades reflexiva, transitiva y simétrica.
- Determina las clases de equivalencia entre dos conjuntos, con base en una relación de equivalencia.
- Identifica si una relación entre dos conjuntos es una función, con base en su definición y propiedades.
- Determina una función resultante, aplicando la composición de funciones.
- Clasifica las funciones en inyectiva, suprayectiva, biyectiva e invertible, con base en sus propiedades.
- Construye funciones biyectivas entre conjuntos de la misma cardinalidad.
- Resuelve problemas de conteo, aplicando las técnicas básicas.
- Obtiene la potencia positiva de un binomio, aplicando el triángulo de Pascal.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Lógica matemática.
- Métodos de demostración.
- Conjuntos y operaciones entre conjuntos.



- Relaciones y funciones.
- Cardinalidad.
- Cálculo Combinatorio.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas.
- Aprendizaje colaborativo.
- Prácticas supervisadas.
- Uso de software matemático.
- Realización de proyectos.
- Aprendizaje autónomo y reflexivo.
- Preguntas dirigidas.

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño.
- Actividades de prácticas supervisadas.
- Resolución de ejercicios y problemas.

Evaluación de producto – 30%

- Elaboración de proyectos.
- Investigaciones documentales.
- Portafolio de evidencias.

9. REFERENCIAS

1. Bravo, Rincón & Rincón (2006). *Álgebra Superior*. México: UNAM.
2. Cárdenas, Raggi & Tomas (1990). *Álgebra Superior*. México: Trillas. (Clásico).
3. Devlin, K. (2004). *Sets, Functions and Logic, an introduction to abstract Mathematics*. (3rd.ed.). USA. Chapman & Hall / CRC Mathematics. (Clásico).
4. Gómez Laveaga, C. (2014). *Álgebra Superior*. México: UNAM.
5. Grimaldi, R. P. (1997). *Matemáticas Discretas y Combinatoria. Una introducción con aplicaciones*. (3a. ed.). México: Addison – Wesley Iberoamericana. (Clásico).
6. Kurosh, A. G. (1977). *Curso de Álgebra Superior*. Moscú: MIR. (Clásico).
7. Lehmann, C. (1993). *Álgebra*. México: Limusa. (Clásico).
8. Mattson, H.F. Jr. (1993). *Discrete Mathematics with Applications*. Canadá: John Wiley & SonsInc. (Clásico).
9. Pinzón, A. (1975). *Conjuntos y Estructuras*. México: Harla. (Clásico).
10. Rosen, K. (1999). *Discrete Mathematics and its Applications*. USA. (3rd. Edition). Mc GrawHill.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



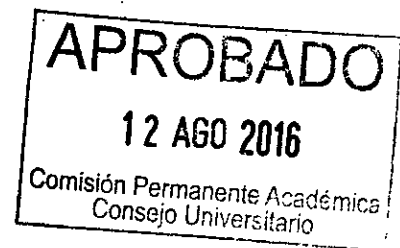
(Clásico).

11. Scheinerman, E. (2001). *Matemáticas Discretas*. USA: Thomson Learning. (Clásico).

12. Suples, P. & Hill, S. (2004). *Introducción a la Lógica Matemática*. México: Reverté. (Clásico).

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o carrera afín, preferentemente con posgrado en Matemáticas.
- Mínimo dos años de experiencia profesional en investigación o trabajo en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Algoritmia

Tipo de asignatura

Obligatoria

Modalidad de la asignatura

Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Algoritmia		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Primer semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La Algoritmia es importante para el estudiante, pues le permite analizar un problema y plantear soluciones mediante la descripción ordenada, precisa y finita de una secuencia de instrucciones.

Esta asignatura aporta al estudiante, los elementos básicos para desarrollar un pensamiento lógico y ordenado; y diseñar estrategias en la solución de problemas, facilitando el proceso de la escritura de programas en cualquier lenguaje de programación.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Algoritmia se relaciona con las asignaturas Fundamentos de Ingeniería de Software, Programación Estructurada, Programación Orientada a Objetos, Teoría de la Computación, Arquitectura y Organización de Computadoras, Diseño de Software, Estructuras de Datos, Sistemas Operativos, Teoría de Lenguajes de Programación, Arquitecturas de Softwares, Construcción de Software, Diseño de Bases de Datos, Desarrollo de Aplicaciones Web, Métricas de Software, Aseguramiento de la Calidad del Software, Requisitos de Software, Interacción Humano Computadora, Experimentación en Ingeniería de Software, Verificación y Validación de Software, Redes y Seguridad de Computadoras, Innovación Tecnológica, Administración de Proyectos I, Mantenimiento de Software, Sistemas Distribuidos y Administración de Proyectos II; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

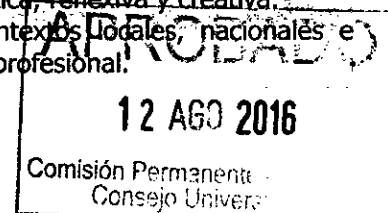
4. COMPETENCIA DE LA ASIGNATURA

Desarrolla algoritmos para la solución de problemas computacionales, en forma secuencial y lógica.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Resuelve problemas en contextos locales, nacionales e internacionales, de manera profesional.



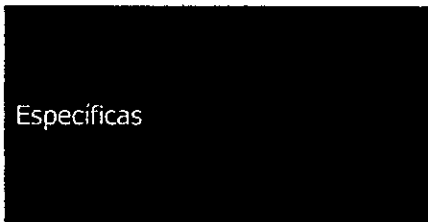


- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.



Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



Específicas

- Describe el proceso de solución de un problema, de forma clara y ordenada.
- Diseña la solución computacional de un problema mediante diagramas de flujo, respetando sus reglas de elaboración.
- Diseña la solución computacional de un problema mediante pseudocódigo, respetando sus reglas de escritura.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción a la algoritmia
- Diseño de algoritmos con diagramas de flujo
- Diseño de algoritmos con pseudocódigo

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Simulación
- Resolución de problemas y ejercicios
- Aprendizaje mediado por las TIC

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 80%

- Resolución de problemas y ejercicios.
- Pruebas de desempeño
- Resolución de casos



Evaluación de producto – 20%

- Portafolio de evidencias
- Desarrollo de proyectos

9. REFERENCIAS

1. Cairó, O. (2006). *Fundamentos de la Programación Piensa en C*. México: Pearson Prentice Hall.
2. Corona, M. & Ancona, M. (2011). *Diseño de Algoritmo y su codificación en lenguaje C*. México: Mc Graw Hill.
3. García, B. (2014). *Fundamentos de Programación*. España: Lulu Press Inc.
4. Joyanes, L. (2008). *Fundamentos de programación algoritmos, estructuras de datos y objetos*. Madrid. Mc Graw Hill.
5. Joyanes, L. (2013). *Fundamentos generales de programación*. México: Mc Graw Hill.
6. Mendez, A. (2013). *Diseño de Algoritmos y su programación en C*. México: AlfaOmega.
7. Sznajdleder, P. (2012). *Algoritmos a fondo con implementación en C y JAVA*. Argentina: Alfaomega.
8. Villalobos, M. (2008). *Fundamentos de programación C# más de 100 algoritmos codificados*. Perú: Macro.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado.
- Mínimo un año de experiencia profesional en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Arquitectura y Organización de Computadoras

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Arquitectura y Organización de Computadoras		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Tercer semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Arquitectura y Organización de Computadoras se enfoca a los aspectos del diseño y organización de la Unidad Central de Procesamiento (CPU) y su integración con los demás componentes que conforman una computadora. La arquitectura del procesador se relaciona con el software computacional, ya que debe cooperar con el sistema operativo y software de base.

Esta asignatura aporta al estudiante un panorama de la arquitectura, la organización y operación de una máquina computacional de propósito general; así como de los diversos dispositivos periféricos que interaccionan y se interconectan al CPU.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Arquitectura y Organización de Computadoras se relaciona con las asignaturas Teoría de la Computación, Estructuras de Datos, Teoría de Lenguajes de Programación, Sistemas Operativos y Sistemas Distribuidos; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

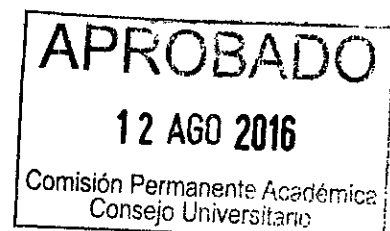
4. COMPETENCIA DE LA ASIGNATURA

Implementa programas de cómputo usando lenguajes de bajo nivel, considerando la arquitectura de hardware.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en inglés de manera oral y escrita, en la interacción con otros de forma adecuada
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.

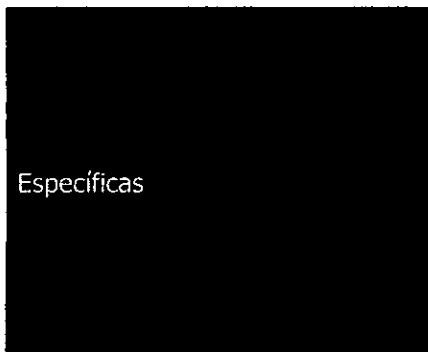




- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.
- Toma decisiones en su práctica profesional y personal, de manera responsable. Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética



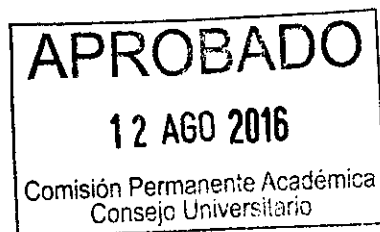
- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



- Identifica claramente los componentes que conforman una computadora.
- Describe con precisión la estructura y organización de la Unidad Central de Procesamiento.
- Describe con precisión la estructura y organización del sistema de memoria de una computadora.
- Describe claramente los componentes de las arquitecturas de cómputo que ejecutan instrucciones en paralelo.
- Implementa algoritmos computacionales, usando lenguajes de bajo nivel.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de la arquitectura de computadoras.
- Arquitectura y organización de la Unidad Central de Procesamiento.
- Instrucciones y operaciones en el lenguaje máquina.
- Organización y arquitectura del sistema de memoria.
- Sistema de Entrada/Salida.
- Procesamiento secuencial y paralelo.





7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de ejercicios y problemas
- Aprendizaje basado en problemas
- Aprendizaje orientado a proyectos
- Prácticas de laboratorio
- Aprendizaje colaborativo

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Prácticas supervisadas

Evaluación de producto – 30%

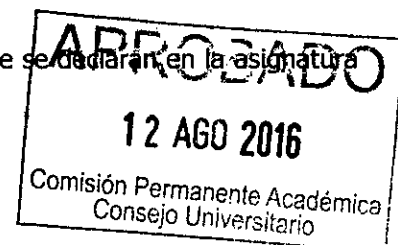
- Elaboración de proyecto

9. REFERENCIAS

1. Hennessy, J. & Patterson, D. (2011). *Computer Organization and Design*. Edition: The Hardware/Software Interface (4th ed.). Morgan Kaufmann.
2. Hennessy, J. & Patterson, D. (2011). *Computer Architecture: A Quantitative Approach* (5th ed.). Morgan Kaufmann.
3. Stallings, W. (2005). *Organización y Arquitectura de Computadores*. (7ª ed). Madrid: Pearson.
4. Harris, D., & Harris, S. (2012). *Digital Design and Computer Architecture* (2nd ed.). Morgan Kaufmann.
5. Null, L. (2010). *Essentials of Computer Organization and Architecture*. EUA: Jones & Bartlett Learning.
6. Hamacher, C., Vranesic, Z., & Zaky, S. (2011). *Computer Organization and Embedded Systems*. McGraw-Hill Science/Engineering/Math.
7. Stokes, J. (2006). *Inside the machine: An illustrated introduction to Microprocessors and Computer Architecture*. No Starch Press. (Clásico)

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software carrera afin, preferentemente con posgrado.
- Mínimo un año de experiencia profesional, en investigación en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declararán en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Arquitecturas de Software

Tipo de asignatura

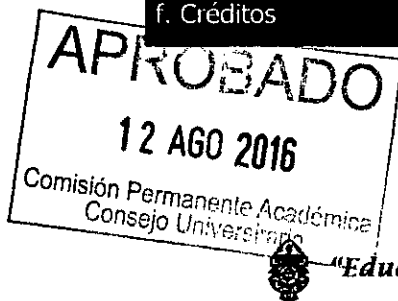
Obligatoria

Modalidad de la asignatura

Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Arquitecturas de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Quinto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Hoy en día la complejidad y tamaño de los sistemas de software ha ido en aumento, por lo que el reto en el diseño de software de éstos ya no son los algoritmos y las estructuras de datos, sino se centra en las interacciones con sistemas existentes, protocolos de comunicación, sincronización en el acceso a la información, incremento en el número de usuarios, entre otros. Para esto es necesario definir una arquitectura de software que permita dar solución a estos requerimientos y garantice el óptimo funcionamiento del sistema.

El propósito de la asignatura es proveer al estudiante los métodos, técnicas y herramientas para la definición, aplicación y evaluación de una arquitectura que cumpla con los requerimientos funcionales y no funcionales de un sistema de software.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Arquitecturas de Software se relaciona con las asignaturas Construcción de Software, Diseño de Software, Mantenimiento de Software y Requisitos de Software; ya que contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos heredados, en diferentes dominios de aplicación, aplicando técnicas, herramientas, métodos y procedimientos, mediante un enfoque sistemático, disciplinado y cuantificable.

4. COMPETENCIA DE LA ASIGNATURA

Diseña una arquitectura para un sistema de software específico, considerando los atributos de calidad requeridos por la disciplina.



5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Describe los atributos de calidad de la arquitectura de software, de acuerdo con los marcos de referencia propios de la disciplina.
- Diseña arquitecturas de software, con los requerimientos funcionales y no funcionales de un sistema, y de acuerdo a los atributos de calidad propios de la disciplina.
- Diseña arquitecturas de software, mediante la integración de componentes y minimizando su dependencia.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de Arquitectura de Software
- Principales patrones arquitectónicos
- Marcos de trabajo
- Documentación de la arquitectura

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Aprendizaje basado en problemas
- Prácticas en laboratorio
- Aprendizaje cooperativo

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70 %

- Prácticas de laboratorio
- Pruebas de desempeño
- Resolución de casos
- Elaboración de reportes

Evaluación de producto – 30 %

- Desarrollo de proyectos

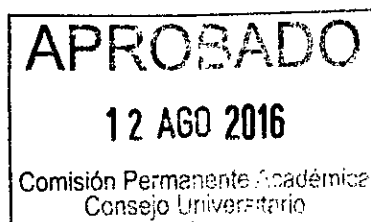


9. REFERENCIAS

1. Bass, L., Clements, P. & Kazman, R. (2003). Software architecture in practice. Boston: (2nd ed.). Addison-Wesley.
2. Clements, P. (2003). Documenting software architectures: views and beyond. Boston: Addison-Wesley.
3. Clements, P., Kazman, R., & Klein, M. (2002). Evaluating software architectures: methods and case studies. Boston: Addison-Wesley.
4. Gorton, I. (2006). Essential software architecture. Berlin: Springer
5. Schmidt, D. C.; Kircher, M., Jain, P., Buschmann, F. & Henney, K. (2008). Pattern-oriented software architecture. New York: Wiley.
6. Shaw, M. & Garlan, D. (1996). Software Architecture: Perspectives on an Emerging Discipline. USA: Prentice Hall.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado en el área.
- Mínimo tres años de experiencia profesional de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Aseguramiento de la Calidad del Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

Nombre de la asignatura	Aseguramiento de la Calidad del Software		
Tipo	Obligatoria		
Modalidad	Mixta		
Ubicación sugerida	Sexto semestre		
Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
Créditos	7		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Aseguramiento de la Calidad es importante para el Ingeniero de Software, porque le permite desarrollar un plan de aseguramiento de la calidad en los procesos de desarrollo del software.

Esta asignatura provee al estudiante estándares, métodos, técnicas y herramientas que se utilizan en el aseguramiento de la calidad del software.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Aseguramiento de la Calidad del Software se relaciona con las asignaturas Fundamentos de Ingeniería de Software, Requisitos de Software, Arquitecturas de Software, Diseño de Software, Diseño de Base de Datos, Construcción de Software, Programación Orientada a Objetos, Programación Estructurada, Desarrollo de Aplicaciones Web, Métricas de Software, Verificación y Validación de Software, Mantenimiento de Software, Administración de Proyecto I y Administración de Proyectos II; ya que contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.
- Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.

4. COMPETENCIA DE LA ASIGNATURA

Aplica la teoría relacionada con el aseguramiento de la calidad del software en los procesos de desarrollo de software de manera continua, crítica y flexible.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

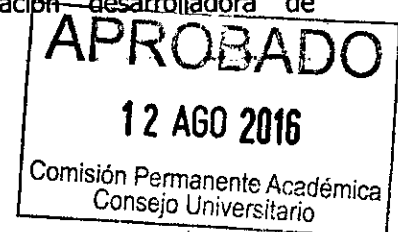
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Analiza conceptos y modelos de aseguramiento de la calidad del software, de manera fundamentada.
- Analiza el proceso de administración de la calidad del software, de manera fundamentada.
- Identifica la estructura de un plan de aseguramiento de la calidad de software, de manera precisa y ordena.
- Aplica los conceptos y modelos de aseguramiento de la calidad en los procesos de desarrollo de software en una organización, de manera crítica y flexible.
- Aplica los procesos de administración de la calidad del software en el desarrollo de software en una organización, de manera crítica y flexible.
- Selecciona las pruebas de software, de acuerdo a los recursos de una organización desarrolladora de software.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de Aseguramiento de la Calidad del Software
- Proceso de Administración de la Calidad del Software
- Plan de aseguramiento de la Calidad del Software
- Pruebas en el ciclo de vida
- Metodología de pruebas de software
- Administración del proceso de prueba

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Escenarios reales de aprendizaje
- Interrogatorio
- Tormenta de ideas
- Aprendizaje colaborativo
- Investigaciones bibliográficas
- Seminarios
- Proyecto integrador

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso 60%

- Seminario
- Investigaciones bibliográficas

Evaluación de producto 40%

- Proyecto Integrador

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



9. REFERENCIAS

- 1 Buerstein, I. (2003). *Practical Software Testing: A Process-Oriented Approach*. USA: Springer Verlag. (Clásico)
- 2 Craig, R. D., Jaskiel S. P. (2002). *Systematic Software Testing*. England: Artech House. (Clásico)
- 3 Galin, D. (2004). *Software Quality Assurance: From Theory to Implementation*. England: Pearson Addison Wesley. (Clásico)
- 4 Godbole, N. (2004). *Software Quality Assurance: Principles and Practice*. U.K.: Alpha Science International. (Clásico)
- 5 Jones, C., & Bonsignour, O. (2011). *The Economics of Software Quality*. USA: Addison-Wesley Professional.
- 6 Kaner, C., Bach, J., & Pettichord, B. (2002). *Lessons Learned In Software Testing: A context-Driven Approach*. USA: Wiley John + Sons. (Clásico)
- 7 Lewis, W. E. (2009). *Software Testing and Continuous Quality Improvement*. USA: Auerbach.
- 8 Page, A., Johnston, K., & Rollison, B. (2008). *How We Test Software at Microsoft*. USA: Microsoft Press
- 9 Radice, R. A. (2004). *High Quality Low Cost Software Inspections*. USA: Paradoxicon. (Clásico)
- 10 Regan, G. O. (2002). *A practical Approach to Software Quality*. USA: Springer. (Clásico)
- 11 Weinberg, G. (1997). *Quality Software Management: Vol. 4 Anticipating Change*. USA: Dorset House Publishing Company, Incorporated. (Clásico)
- 12 Weinberg, G. (2008). *Perfect Software and other illusions about testing*. USA: Dorset House Publishing Co Inc.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de software o carrera afín, preferentemente con estudios de postgrado.
- Mínimo dos años de experiencia profesional de trabajo en el área de calidad o gestión de proyectos software.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Cálculo Diferencial

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Cálculo Diferencial		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Segundo semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio del Cálculo Diferencial contribuye a la resolución de problemas en diferentes contextos, mediante el análisis de las funciones que modelan los fenómenos y la aplicación de la diferencial de la función.

En particular, el Cálculo Diferencial es importante para la formación de los estudiantes, ya que les permitirá obtener herramientas matemáticas para la modelación computacional de problemas reales.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Cálculo Diferencial se relaciona con las asignaturas Álgebra Intermedia, Geometría Analítica, Cálculo Integral, Probabilidad e Inferencia Estadística; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Diferencial de una variable y de herramientas computacionales de forma eficiente.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Toma decisiones en su práctica profesional y personal.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



[Redacted]

manera responsable.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.

Específicas

- Resuelve problemas planteados en la ingeniería y las ciencias, aplicando las herramientas matemáticas y computacionales de límites y continuidad.
- Utiliza la noción de derivada en contextos matemáticos y de aplicaciones básicas, mediante procedimientos aritméticos, algebraicos y geométricos.
- Modela situaciones de variación y cambio en contextos matemáticos, utilizando la diferenciación y funciones.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Límite y continuidad de funciones reales de variable real.
- La derivada y las principales técnicas de derivación.
- Optimización y aplicaciones de la derivada

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de problemas y ejercicios
- Aprendizaje basado en problemas
- Proyectos
- Simulación
- Aprendizaje cooperativo
- Aprendizaje autónomo y reflexivo
- Uso de graficadores

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

A Evaluación de proceso - 60%

- Pruebas de desempeño
- Críticas

12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Evaluación de producto – 40%

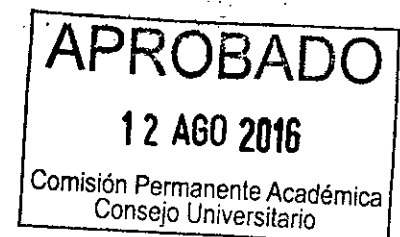
- Elaboración de proyecto
- Portafolio de evidencias

9. REFERENCIAS

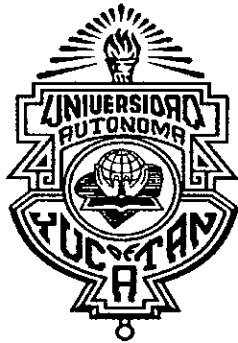
1. Banner, A. (2007). *The calculus lifesaver: All the tools you need to excel at calculus*. Princeton University Press.
2. Larson, R., & Edwards, H. (2009). *Calculus* (9th ed.). Brooks Cole.
3. Purcell E. (2007). *Cálculo Diferencial e Integral* (9ª ed.). México: Pearson Educación de México.
4. Stewart, J. (2006). *Cálculo Diferencial e Integral* (2ª ed.). México: Thomson Corporation.
5. Stewart, J. (2007). *Calculus* (6th ed.). Brooks Cole.
6. Stewart, J. (2010). *Cálculo de una variable: conceptos y contextos* (4ª ed.). México: Cengage Learning.
7. Stewart, J. (2011). *Single variable calculus: Early Transcendentals*. (7th ed.). Brooks Cole.
8. Salas, S. (2005). *Cálculo*. España: Reverté.
9. Varber, D., Purcell E., & Rigdon, S. (2006). *Calculus*. (9th ed.). USA: Prentice Hall.
10. Zill, D. (2011). *Matemáticas 2 Cálculo Integral*. México: McGraw-Hill | Interamericana.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o carrera afín, preferentemente con posgrado en Matemáticas.
- Mínimo un año de experiencia profesional en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Cálculo Integral

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Cálculo Integral		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Tercer semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Haber acreditado la asignatura Cálculo Diferencial.

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio del Cálculo Integral contribuye a la resolución de problemas en diferentes contextos, mediante el análisis de las funciones que modelan los fenómenos y la aplicación de la integral de la función.

En particular, el Cálculo Integral es importante para la formación de los estudiantes, ya que les permitirá obtener herramientas matemáticas para la modelación computacional de problemas reales.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Cálculo Integral se relaciona con las asignaturas Álgebra Intermedia, Geometría Analítica, Cálculo Integral, Probabilidad e Inferencia Estadística; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

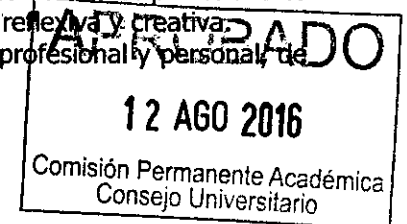
4. COMPETENCIA DE LA ASIGNATURA

Resuelve problemas matemáticos en el ámbito de la ingeniería haciendo uso del Cálculo Integral de una variable y de herramientas computacionales de forma eficiente.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Toma decisiones en su práctica profesional y personal, de



[Redacted]

manera responsable.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.

Específicas

- Analiza los significados de la integral en diferentes contextos matemáticos, de forma clara y ordenada.
- Resuelve ejercicios y problemas matemáticos, utilizando las propiedades de la integral y herramientas computacionales.
- Resuelve problemas matemáticos en el área de la ingeniería, usando el cálculo integral y herramientas computacionales.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Antiderivadas.
- Suma de Riemann.
- Integral definida
- Teorema fundamental del cálculo.
- Integral indefinida.
- Técnicas de integración.
- Integral impropia.
- Aplicaciones de la integral en física e ingeniería.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de problemas y ejercicios
- Aprendizaje basado en problemas
- Proyectos
- Simulación
- Aprendizaje cooperativo
- Aprendizaje autónomo y reflexivo
- Uso de graficadores

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Pruebas de desempeño
- Resolución de ejercicios y problemas
- Resolución de casos

Evaluación de producto – 40%

- Elaboración de proyecto
- Portafolio de evidencias

9. REFERENCIAS

1. Banner, A. (2007). *The calculus lifesaver: All the tools you need to excel at calculus*. Princeton University Press.
2. Larson, R., & Edwards, H. (2009). *Calculus* (9th ed.). Brooks Cole.
3. Purcell E. (2007). *Cálculo Diferencial e Integral* (9ª ed.). México: Pearson Educación de México.
4. Stewart, J. (2006). *Cálculo Diferencial e Integral* (2ª ed.). México: Thomson Corporation. (Clásico).
5. Stewart, J. (2007). *Calculus* (6th ed.). Brooks Cole.
6. Stewart, J. (2010). *Cálculo de una variable: conceptos y contextos* (4ª ed.). México: Cengage Learning.
7. Stewart, J. (2011). *Single variable calculus: Early Transcendentals*. (7th ed.). Brooks Cole.
8. Salas, S. (2005). *Cálculo*. España: Reverté. (Clásico).
9. Varber, D., Purcell E., & Rigdon, S. (2006). *Calculus*. (9th ed). USA: Prentice Hall. (Clásico).
10. Zill, D. (2011). *Matemáticas 2 Cálculo Integral*. México: McGraw-Hill | Interamericana.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o carrera afín, preferentemente con posgrado en Matemáticas.
- Mínimo un año de experiencia profesional en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Construcción de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Construcción de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Quinto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Construcción de Software es importante para el Ingeniero de Software porque le permitirá crear código simple y legible.

Esta asignatura aporta al estudiante la teoría básica relacionada con la fase de construcción del software para llevar un control riguroso en el desarrollo de software.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Construcción de Software se relaciona con las asignaturas Fundamentos de Ingeniería de Software, Arquitecturas de Software, Diseño de Software, Diseño de Base de Datos, Programación Orientada a Objetos, Programación Estructurada, Algoritmia, Desarrollo de Aplicaciones Web, Métricas de Software, Aseguramiento de la Calidad del Software, Requisitos de Software, Verificación y Validación de Software, Mantenimiento de Software, Administración de Proyecto I y Administración de Proyectos II; ya que en conjunto contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla aplicaciones utilizando prácticas y técnicas de construcción de software, considerando los estándares de codificación.

5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.

Disciplinares

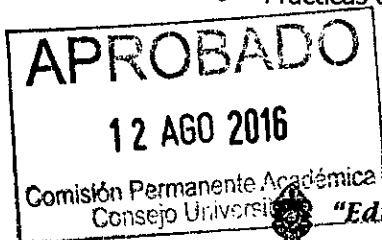
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Aplica prácticas de construcción de software durante la escritura del código, de manera simple y legible.
- Aplica técnicas de refactorización durante la escritura del código, para la mejora de la simplicidad y la legibilidad.
- Utiliza herramientas de construcción de software durante la escritura del código, de manera ágil.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de construcción
- Prácticas de construcción



- Principios de refactorización
- Técnicas de refactorización
- Herramientas para la construcción

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Escenarios reales de aprendizaje
- Interrogatorio
- Tormenta de ideas
- Demostración
- Investigaciones bibliográficas
- Aprendizaje colaborativo
- Seminarios
- Resolución de problemas
- Análisis de códigos

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Seminarios
- Investigaciones bibliográficas

Evaluación de producto – 40%

- Proyecto Integrador

9. REFERENCIAS

1. Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2002). *Documenting software architectures: views and beyond*. USA: Pearson Education. (Clásico)
2. Maguire, S. (1993). *Writing Solid Code: Microsoft's Techniques for Developing Bug-Free C Programs (Microsoft Programming Series)*. USA: Microsoft Press. (Clásico)
3. Martin, F., Brant K., & John B. (1999). *Refactoring: improving the design of existing code*. USA: Addison Weley. (Clásico)
4. McConnell, S. (2004). *Code complete*. USA: Microsoft Press. (Clásico)



APROBADO

12 AGO 2016

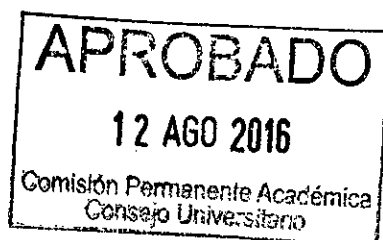
Comisión Permanente Académica
Consejo Universitario



5. Mili, H., Mili, A., Yacoub, S., & Addy, E. (2001). *Reuse-based software engineering: techniques, organization, and controls*. USA: Wiley-Interscience. (Clásico)
6. Sommerville, I. (2011). *Software Engineering*. USA: Addison-Wesley

10 PERFIL DESEABLE DEL PROFESOR

- Ingeniero en software o afín, preferentemente con estudios de postgrado.
- Mínimo un año de experiencia profesional de trabajo en el área de construcción de software.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Cultura Maya

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Cultura Maya		
b. Clasificación	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Segundo semestre		
e. Duración total en horas	96	Horas presenciales 48	Horas no presenciales 48
f. Créditos	6		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La asignatura "Cultura maya" para estudiantes universitarios permite un acercamiento a la cultura de la península de Yucatán, mediante los diferentes elementos que la caracterizan, asimismo permite comprender por qué es importante "RECONOCER Y VALORAR LA CULTURA MAYA" dentro del contexto universitario conformado por una sociedad multicultural. Por otra parte, permitirá obtener los conocimientos básicos sobre los elementos que conforman la cultura maya y en particular la identidad del maya contemporáneo. De la misma manera promueve valorar y respetar la diversidad cultural en el plano social e institucional, así como desarrollar un pensamiento crítico, reflexivo y creativo. El enfoque de la asignatura considera la investigación y análisis crítico de los temas que servirán de guía para la construcción del aprendizaje del estudiante y su difusión.

Que los estudiantes comprendan el concepto de identidad a través de la cultura maya y de los diversos elementos que la conforman y que han contribuido a su evolución y manifestación actual, lo que permitirá reflexionar y aportar desde su disciplina, los conocimientos necesarios para la revaloración y conformación del ser maya contemporáneo.

3. RELACION CON OTROS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

La asignatura Cultura Maya, al ser una asignatura institucional obligatoria tiene una relación transversal con las competencias de egreso de los programas educativos de la universidad a nivel licenciatura.

4. COMPETENCIA DE LA ASIGNATURA

Establece propuestas de solución a las problemáticas actuales de la sociedad, desde la realidad de la cultura maya, promoviendo la revaloración de la misma bajo los principios de multiculturalidad e interculturalidad.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.





- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.
- Valora la diversidad y multiculturalidad en su quehacer cotidiano, bajo los criterios de la ética.
- Aprecia las diversas manifestaciones artísticas y culturales en su quehacer cotidiano, de manera positiva y respetuosa.
- Valora la cultura maya en su quehacer cotidiano, de manera positiva y respetuosa.

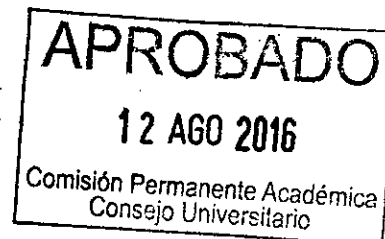
Disciplinares

- No aplica



Específicas

- Reconoce su identidad cultural en prácticas sociales y contextos diversos como sujeto y parte de una cultura.
- Explica la situación actual de la cultura maya tomando como referencia su historia y su lengua, con una visión crítica de la realidad
- Explica la cosmovisión de la cultura maya con las implicaciones en la vida, religión, arte, arquitectura, ciencia y lengua, tomando como referencia la relación hombre-naturaleza, y una visión crítica de la situación actual de la humanidad.
- Explica las aportaciones de la cultura maya en las innovaciones científicas y tecnológicas, desde una visión crítica, fomentando la revaloración de los conocimientos ancestrales mayas
- Explica el valor de la cultura maya con referencia a la identidad del ser maya contemporáneo y las diversas manifestaciones de la cultura, con una visión crítica.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- El concepto antropológico de cultura
- Multiculturalidad e interculturalidad
- Identidad cultural
- Área maya en Mesoamérica y área maya peninsular
- Historia breve de la civilización maya
- Lengua Maya y sus variantes
- Centros ceremoniales y principales asentamientos
- El origen del hombre a través de la literatura maya
- La Milpa y el Maíz como fundamento de la cosmovisión
- Casa Maya
- Las Matemáticas, la Ingeniería y la Arquitectura
- La Medicina
- La Astronomía y los Calendarios
- Identidad del ser maya yucateco contemporáneo
- Vida cotidiana, acciones actuales
- Manifestaciones culturales contemporáneas

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Elaboración de organizadores gráficos
- Análisis de conceptos mediante ejemplos prácticos de la disciplina (estudios de caso)
- Aprendizaje en escenarios reales
- Aprendizaje colaborativo
- Aprendizaje autónomo y reflexivo
- Investigación documental haciendo uso de las TIC's
- Elaboración de objetos de aprendizaje
- Entrevistas a expertos
- Documentación audiovisual de algún elemento cultural contemporáneo

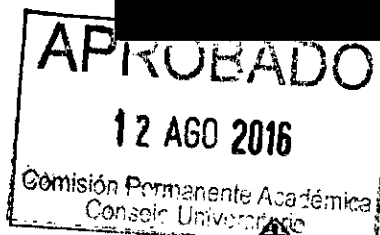
8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Elaboración de proyectos de integración
- Reportes de investigación documental
- Elaboración de ensayos

Evaluación de producto – 40%

- Presentación del proyecto "Ser maya yucateco contemporáneo"
- Portafolio de evidencias

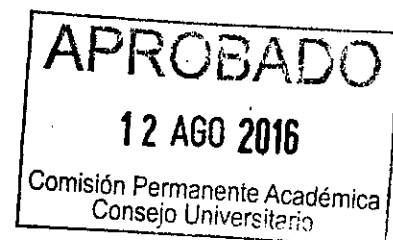


9. REFERENCIAS

1. Ancona, E. (1978). *Historia de Yucatán*. Yucatán, México: Universidad Autónoma de Yucatán (13)
2. Canto, A.L.C. (2005). *El diseño en la arquitectura prehispánica maya: la geometría y la astronomía como parte fundamental en el proceso arquitectónico*. Tesis de maestría. Universidad Autónoma de Yucatán. Facultad de Arquitectura (29)
3. Casares, O. (2004). *Astronomía en el área maya*. Mérida, Yucatán, México: UADY (37)
4. Chávez, C.M. (s/f). *Medicina maya en el Yucatán colonial (siglos XVI-XVIII)*. Tesis de doctorado. UNAM, Facultad de Filosofía y Letras (35)
5. González, N. y Mas, J. (2003). El nuevo concepto de cultura: la nueva visión del mundo desde la perspectiva del otro. *Pensar Iberoamérica, revista de cultura*. Organización de Estados Iberoamericanos para la Educación, la ciencia y la cultura. Disponible en internet: <http://www.oei.es/pensariberoamerica/colaboraciones11.htm> (2)
6. Kirchof, P. (1960). Mesoamérica. *Suplemento de la revista Tlatoani 3*. Escuela Nacional de Antropología e Historia. México (41)
7. Libros del Chilam balam (25)
8. Ramundo, P.S. (2004). *El concepto antropológico de cultura*. Argentina: IDIP (1)
9. Rodríguez, I.E. (2005). *Estudio del comportamiento estructural de la vivienda maya tesis de licenciatura*. México. Universidad Autónoma de Yucatán. Facultad de Ingeniería (28)
10. Ruz, M.H. (2006). *Mayas: primera parte. Pueblos indígenas del México Contemporáneo*. México: CDI:PNUD (19)
11. Sam Colop, L. E. (2008). *Popol Wuj Cholsamaj*. Guatemala (21)
12. Staines, L. (2004). Pintura mural maya. *Revista Digital Universitaria* [en línea]. 10 de agosto de 2004, Vol. 5, No. 7. [Consultada: 11 de octubre de 2011]. Disponible en Internet: <<http://www.revista.unam.mx/vol.5/num7/art40/art40.htm>>ISSN: 1607-6079. (18)
13. Trejo, S. (Editora, 2000) *Arquitectura e ideología de los antiguos mayas: Memoria de la Segunda Mesa Redonda de Palenque 1997*. México: CONACULTA: INAH (31)

10. PERFIL DESEABLE DEL PROFESOR

- Identificarse con la cultura maya y con la filosofía universitaria
- Amplio conocimiento de la historia y cultura maya
- Originario del área maya peninsular y haber radicado los últimos tres años en el mismo
- Conocimiento de conceptos básicos de la lengua maya
- Diplomado en Humanidades Mayas o afín.
- Licenciados del área del campus de ciencias sociales o bien, profesor del área disciplinar del programa educativo, que desarrolle investigación o actividades en el tema de la cultura maya.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Desarrollo de Aplicaciones Web

Tipo de asignatura

Obligatoria

Modalidad de la asignatura

Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Desarrollo de Aplicaciones Web		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Quinto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La World Wide Web ha venido a significar un cambio en numerosos ámbitos. Hoy en día, las tecnologías involucradas en el desarrollo de documentos y aplicaciones web han facilitado el modo de crear productos que pueden ser accedidos desde dispositivos con diferentes características, como tamaños de pantalla y velocidad de procesamiento, entre otros.

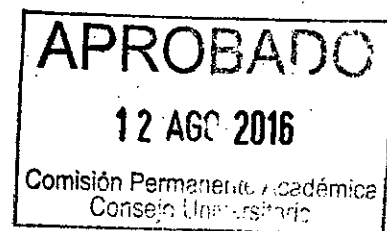
Una aplicación Web es aquella que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es una de las razones más importantes de su popularidad.

El propósito de esta asignatura consiste en ofrecer a los participantes las bases teóricas y prácticas respecto al proceso de desarrollo de aplicaciones web, siguiendo una metodología apegada a los estándares actuales, integrando diferentes tecnologías, lenguajes de programación y bases de datos, entre otras.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

La asignatura Desarrollo de Aplicaciones Web se relaciona con las asignaturas: Algoritmia, Programación Estructurada, Programación Orientada a Objetos, Arquitecturas de Software, Diseño de Software, Mantenimiento de Software, Administración de Proyectos, Estructuras de Datos y Redes y Seguridad de Computadoras; ya que contribuye al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.



4. COMPETENCIA DE LA ASIGNATURA

Desarrolla aplicaciones Web con acceso a base de datos, de manera eficiente y creativa, utilizando las tecnologías más recientes en el área.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

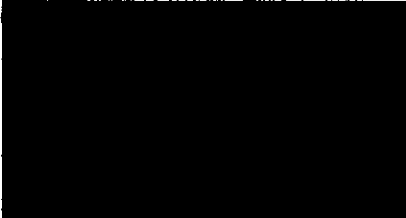
- Identifica las principales características de las tecnologías involucradas en el desarrollo de una aplicación Web, de manera pertinente.
- Elabora páginas Web estáticas utilizando etiquetas HTML y estilos CSS, con apego a los estándares vigentes.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





- Elabora páginas Web dinámicas utilizando el lenguaje JavaScript, con apego a los estándares vigentes.
- Configura un servidor de aplicaciones Web a través de la Internet de manera eficaz y eficiente.
- Elabora aplicaciones Web con acceso a bases de datos considerando esquemas de calidad.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Conceptos preliminares
- Etiquetas HTML
- Estilos CSS
- Lenguaje JavaScript
- Servidor de base de datos
- Servidor de aplicaciones Web
- Lenguajes interpretados del lado del servidor
- Aplicaciones Web con acceso a bases de datos
- Bibliotecas y herramientas para el desarrollo de aplicaciones Web

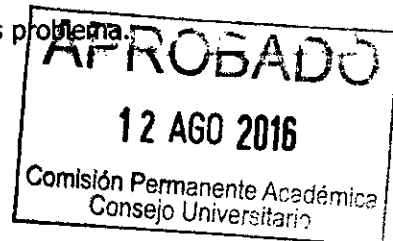
7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de Casos
- Aprendizaje orientado a proyectos.
- Prácticas en laboratorio.
- Resolución de problemas y ejercicios.
- Aprendizaje mediado por las TIC.
- Aprendizaje colaborativo.

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño.
- Resolución de casos.
- Elaboración de reportes.
- Resolución de situaciones problema.



Evaluación de producto – 30%

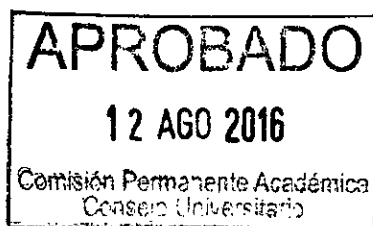
- Desarrollo de proyectos
- Portafolio de evidencias

9. REFERENCIAS

1. Duckett, J. (2014). *JavaScript and JQuery: Interactive Front-End Web Development*. Indiana: Wiley.
2. Frain, B. (2015). *Responsive Web Design with HTML5 and CSS3*. (2a. ed.). United Kingdom: Packt Publishing Ltd.
3. Freeman, E. y Rosbson, E. (2011). *Head First HTML5 Programming. Building Web Apps with JavaScript*. EUA: O'Reilly Media Inc.
4. Myers, M. (2015). *A Smarter Way to Learn JavaScript*. Recuperado de: http://www.cpp.edu/~jcmgarvey/513_2016/ASmarterWaytoLearnJavaScript.pdf
5. Niederst, J. (2012). *Learning Web Design. A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. (4th. ed.). Canadá: O'Reilly Media Inc.
6. Nixon, R. (2015). *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. (4th. ed.). USA: O'Reilly Media Inc.
7. Purewal, S. (2014). *Learning Web App Development*. USA: O'Reilly Media Inc.
8. Welling, L. y Thomson L. (2009). *PHP and MySQL Web Development*. (4th. ed.). USA: Pearson Education.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado en el área.
- Mínimo tres años de experiencia profesional de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir..



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Diseño de Bases de Datos

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Diseño de Bases de Datos		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Quinto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Diseño de Bases de Datos es importante para el estudiante en Ingeniería de Software porque permite diseñar la estructura lógica para registrar, mantener y recuperar la información relevante y persistente en una organización.

Esta asignatura proporciona los métodos y herramientas para el diseño lógico de las estructuras de bases de datos.

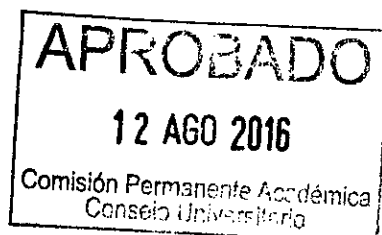
3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Diseño de Bases de Datos se relaciona con las asignaturas Algoritmia, Programación Estructurada, Programación Orientada a Objetos, Estructuras de Datos, Diseño de Software, Construcción de Software y Desarrollo de Aplicaciones Web; ya que contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificado.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Diseña la estructura lógica de una base de datos, para la solución de un problema del mundo real, utilizando los modelos y técnicas apropiadas.



5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

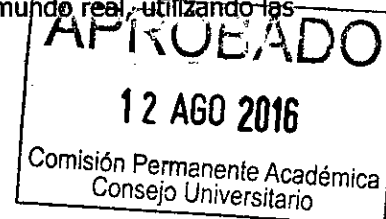
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal utilizando correctamente el idioma.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento, en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Explica los conceptos de bases de datos y el ciclo de desarrollo de sistemas de bases de datos, de manera clara y ordenada.
- Aplica el proceso de diseño lógico de una base de datos, para la solución de un problema del mundo real, utilizando el Modelo de Entidad-Relación.
- Explica los principios de la Teoría Relacional en el modelado de los datos, de manera clara y ordenada.
- Aplica el proceso de diseño lógico de una base de datos, en un problema del mundo real, utilizando las reglas de Normalización.





- Codifica las operaciones de definición y manipulación de datos en un sistema de bases de datos, utilizando el lenguaje estructurado de consultas.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de bases de datos: Conceptos básicos, Modelos de datos, Ciclo de Vida de desarrollo de Sistemas de bases de datos.
- El Modelo de Entidad-Relación
- El Modelo Relacional: Estructura, Integridad, Álgebra Relacional
- Normalización
- Lenguaje Estructurado de Consultas (SQL)

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Proyectos de investigación
- Resolución de problemas y ejercicios
- Prácticas supervisadas
- Aprendizaje orientado a proyectos
- Aprendizaje basado en problemas

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Prácticas supervisadas
- Resolución de ejercicios y problemas

Evaluación de producto – 30%

- Elaboración de proyectos de bases de datos
- Reporte de proyectos de bases de datos
- Investigaciones documentales

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

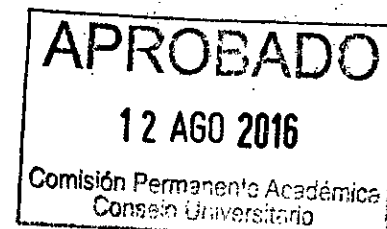


9. REFERENCIAS

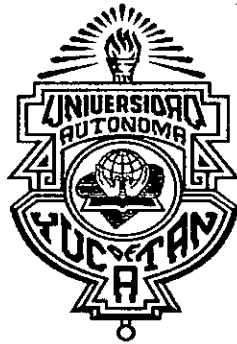
1. Connolly, T. M., & Begg, C. E. (2005). *Sistemas de bases de datos: Un enfoque práctico para diseño, implementación y gestión*. Madrid: Pearson Educación.
2. Date, C. J. (2003). *An introduction to database systems*. Pearson Education.
3. Elmasri, R., & Navathe, S. (2011). *Database systems: Models, languages, design, and application programming*. Boston, MA: Pearson.
4. Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems*. Boston: Pearson.
5. Gillenson, M. L. (2006). *Fundamentals of database management systems: With beginning sql server 2005 programming*. John Wiley & Sons.
6. McLaughlin, M. (2013). *MySQL Workbench: Data Modeling & Development*. Oracle Press.
7. Rob, P., Coronel, C., & Crockett, K. (2008). *Database systems: Design, implementation & management*. London: Cengage Learning.
8. Silberschatz, A., Korth, H., & Shadarshan, S. (2011). *Database Systems Concepts*. McGraw Hill Education.
9. Teorey, T. J., Lightstone, S. S., Nadeau, T., & Jagadish, H. V. (2011). *Database Modeling and Design* (5th ed.). The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado en el área de Tecnologías de Información.
- Mínimo dos años de experiencia profesional de trabajo en el área de programación.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Diseño de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Diseño de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Cuarto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El diseño de software es parte de toda metodología en el desarrollo de sistemas que tiene como fin proveer una solución a un conjunto de requerimientos, que guía a los desarrolladores de software en la construcción del producto.

La asignatura tiene como propósito aportar en la formación de los estudiantes, los principios característicos de la Ingeniería de Software para la especificación de los requerimientos funcionales y no funcionales de un sistema, considerando la factibilidad del mismo y las características de calidad requeridas.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Diseño de Software se relaciona con las asignaturas Construcción de Software, Arquitecturas de Software, Mantenimiento de Software, Requisitos de Software y Programación Orientada a Objetos ya que contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Diseña productos de software utilizando técnicas y marcos de referencia de la disciplina que aseguren los atributos de calidad requeridos.



5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Identifica los conceptos básicos del diseño de software de manera coherente con los marcos de referencia propios de la disciplina.
- Analiza los requerimientos funcionales y no funcionales de los sistemas de software, a través del uso de marcos de referencia de la ingeniería de software.
- Utiliza un lenguaje para el modelado de software enfocado al diseño de productos del mismo, de manera fundamentada.
- Genera el diseño de un software a través del uso de técnicas y patrones básicos que aseguren la calidad del mismo.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al diseño de software.
- Atributos de calidad en el diseño de software.
- Lenguaje para el modelado del diseño.
- Técnicas para alcanzar los atributos de calidad en un diseño de software.
- Procesos para el diseño de sistemas.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Aprendizaje autónomo y reflexivo
- Aprendizaje mediado por las TIC
- Aprendizaje orientado a proyectos.
- Prácticas en laboratorio.

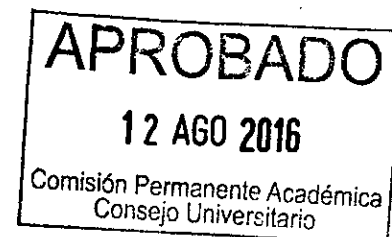
8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Resolución de situaciones problema
- Elaboración de reportes
- Pruebas de desempeño
- Prácticas de laboratorio.

Evaluación de producto – 30%

- Desarrollo de proyectos.

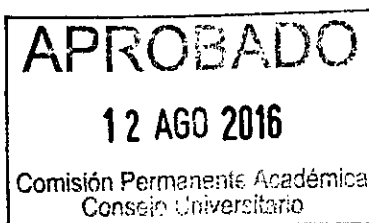


9. REFERENCIAS

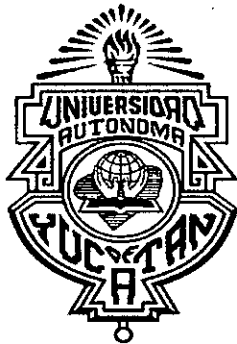
1. Gomaa, H. (2011). *Software modeling and design: UML, use cases, patterns, and software architectures*. USA: Cambridge University Press.
2. Jacobson, I., Booch, G. & Rumbaugh, J. (2008). *El proceso unificado de desarrollo de software*. España: Pearson Educación.
3. Kendall, K., & Kendall, J. (2011). *Análisis y diseño de sistemas*. (8ª. ed.). México, DF: Pearson Educación.
4. Bennett, S., McRobb, S. & Farmer, R. (2007). *Análisis y diseño orientado a objetos de sistemas usando*. Madrid: McGraw-Hill.
5. Zhu, H. (2005). *Software design methodology*. Boston: Elsevier.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado en el área.
- Mínimo tres años de experiencia profesional de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

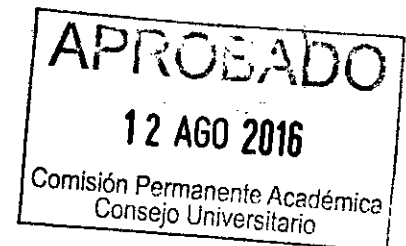


Estructuras de Datos

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Estructuras de Datos		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Cuarto semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		
g. Requisitos académicos	Ninguno		



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de las Estructuras de Datos permite desarrollar programas de cómputo eficientes que utilicen estructuras de datos avanzadas utilizando un lenguaje de programación.

El propósito de esta asignatura es aportar los elementos para aplicar estructuras de datos lineales y no lineales, discernir sobre la mejor estructura de datos para un problema específico y decidir en el uso de los algoritmos de ordenamiento y/o búsqueda más apropiados para resolver un problema determinado.

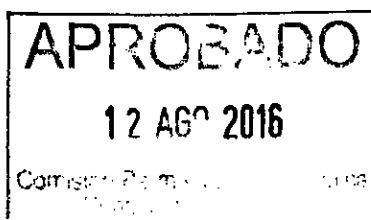
3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Estructuras de Datos se relaciona con las asignaturas Algoritmia, Programación Estructurada, Programación Orientada a Objetos, Diseño de Bases de Datos, Sistemas Operativos, Arquitectura y Organización de Computadoras y Teoría de Lenguajes de Programación; ya que en conjunto contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Emplea las estructuras de datos lineales y no lineales en el desarrollo de software de aplicación de pequeña a mediana escala, para el manejo eficiente de la información.



5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.
- Trabaja bajo presión de manera eficaz y eficientemente.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Aplica conceptos de pilas y colas para el almacenamiento y recuperación de datos, de manera eficiente en programas de software.
- Aplica los conceptos de listas dinámicas para el manejo eficiente de datos en programas de software.
- Ordena datos por medio de algoritmos eficientes en términos de tiempo de ejecución y requerimientos de almacenamiento.



- Aplica algoritmos de dispersión para la búsqueda y recuperación eficiente de información en la memoria principal de una computadora.
- Aplica los conceptos de árboles y grafos para el manejo eficiente de datos en programas de software.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Estructuras de datos básicas: pilas y colas
- Listas dinámicas
- Algoritmos de ordenamiento
- Árboles
- Grafos
- Tablas de dispersión

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Prácticas de laboratorio
- Resolución de ejercicios
- Aprendizaje Cooperativo
- Seminarios
- Investigación bibliográfica
- Desarrollo de proyectos de software

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño.
- Prácticas de software.
- Proyectos de software especiales por tema.

Evaluación de producto – 30%

- Elaboración de proyecto integrador.
- Reporte del proyecto integrador.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



9. REFERENCIAS

1. Cairó, O. y Guardati, S. (2010). *Estructuras de datos*. (3ª ed.). México: Mc. Graw Hill.
2. Canal, J. A. (2011). *Diseño de estructuras de datos*. Barcelona: UOC.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2009). *Introduction to Algorithms*. (3a. ed.). USA: The MIT Press.
4. Abasolo, M. J., & Perales, F. J. (2011). *Introducción a la programación*. Palma: Edicions Universitat de les Illes Balears.
5. Mora, S. L. (2014). *Ejercicios resueltos sobre programación y estructuras de datos*. España: Publicaciones de la Universidad de Alicante.
6. Pous, X. S. (2010). *Estructuras de datos básicas: Secuencias*. Barcelona: UOC.
7. Sedgewick, R. & Wayne, K. (2011). *Algorithms* (4th ed.). USA: Addison-Wesley Educational Publishers Inc. Princeton University.
8. Serna, L. A., Unanue, R. M., & Artacho, M. R. (2011). *Programación y estructuras de datos avanzadas*. Madrid: Centro de Estudios Ramón Areces.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado.
- Mínimo un año de experiencia profesional en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Experimentación en Ingeniería de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Experimentación en Ingeniería de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Séptimo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Inferencia Estadística

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Experimentación en Ingeniería de Software permite identificar y analizar las relaciones entre los diferentes factores o variables inmersos en los fenómenos que se presentan en los procesos de desarrollo de software.

La asignatura ofrece al estudiante herramientas para validar de manera sistemática, disciplinada y cuantificable, propuestas de innovación o mejora a los procesos vinculados con la Ingeniería de Software.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Experimentación en Ingeniería de Software se relaciona con las asignaturas Probabilidad, Inferencia Estadística y Métricas de Software; contribuyendo al logro de la competencia de egreso del área de Innovación en Ingeniería de Software:

- Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, sustentadas en fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.

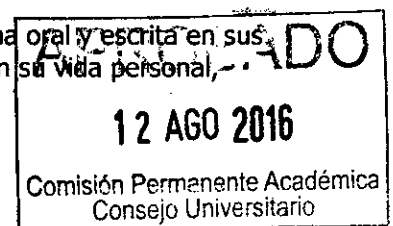
4. COMPETENCIA DE LA ASIGNATURA

Aplica el proceso experimental en estudios empíricos sobre fenómenos de la Ingeniería de Software, utilizando los métodos, técnicas y herramientas apropiadas del diseño y análisis de experimentos.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

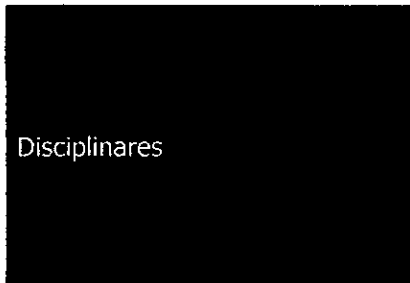
Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal.

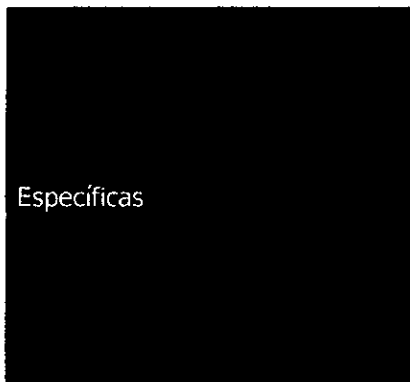




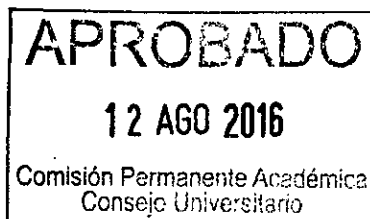
- utilizando correctamente el idioma.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.
- Toma decisiones en su práctica profesional y personal, de manera responsable



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.



- Identifica los elementos básicos del proceso de la experimentación, de acuerdo con los marcos de referencia propio de la disciplina.
- Aplica modelos de regresión lineal simple en el diseño de software, de manera fundamentada.
- Aplica modelos de regresión lineal múltiple en el diseño de software, de manera fundamentada.
- Analiza datos provenientes de diseños de un factor, de forma clara y ordenada.
- Analiza datos provenientes de los diseños factoriales, de manera fundamentada.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- El proceso de la experimentación en Ingeniería de Software
- Modelos de Regresión Lineal Simple
- Modelos de Regresión Lineal Múltiple
- Diseño de un Factor con y sin bloqueo
- Diseño Factorial

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje autónomo y reflexivo
- Aprendizaje colaborativo
- Investigación documental
- Resolución de problemas y ejercicios
- Prácticas en laboratorio
- Estudio de casos
- Seminarios

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso - 70%

- Pruebas de desempeño
- Análisis de casos
- Elaboración de reportes

Evaluación de producto - 30%

- Portafolio de evidencias
- Desarrollo de Proyectos

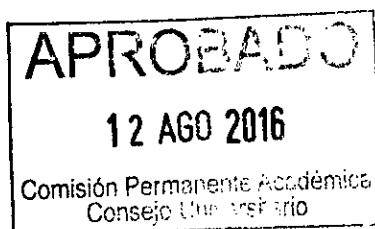


9. REFERENCIAS

1. Genero, M., Cruz-Lemus, J. y Piattini, M. (2014). *Métodos de Investigación en Ingeniería de Software*. Ra-Ma.
2. Juristo, N., & Moreno, A. (2001). *Basics of Software Engineering Experimentation*. Kluwer Academic Publishers, Boston. (Clásico)
3. Kleinbaum, D.G., Kupper L.L., Muller K.E. y Nizam A. (2007). *Applied Regression Analysis and Other Multivariate Methods*. (4ta. ed). Wadsworth Publishing Co Inc. (Clásico)
4. Milton, J. y Arnold J. (2003) *Probabilidad y Estadística con Aplicaciones para Ingeniería y Ciencias Computacionales*. (4ª ed.). McGraw Hill. (Clásico)
5. Montgomery D.C. (2004). *Diseño y Análisis de Experimentos*. (2ª ed.). Limusa. (Clásico)
6. Montgomery, D.C., Peck, E.A. y Vining, G.G. (2012). *Introduction to Linear Regression Analysis*. (5ta ed.). John Wiley & Sons, Inc.
7. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B. & Wesslen, A. (2000). *Experimentation in Software Engineering: An introduction*. Kluwer Academic Publishers. (Clásico)

10. PERFIL DESEABLE DEL PROFESOR

- Profesionista con estudios de posgrado en las áreas de Estadística o Ingeniería de Software, preferentemente con estudios de doctorado.
- Mínimo dos años de experiencia profesional en proyectos de Innovación o de Investigación en el área de la Ingeniería de Software.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Fundamentos de Ingeniería de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Fundamentos de Ingeniería de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Primer Semestre		
e. Duración total en horas	96	Horas presenciales 64	Horas no presenciales 32
f. Créditos	6		
g. Requisitos académicos	Ninguno		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La asignatura es importante para el estudiante de la Ingeniería de Software porque presenta un panorama general de las principales áreas que integran el cuerpo de conocimientos para un Ingeniero de Software.

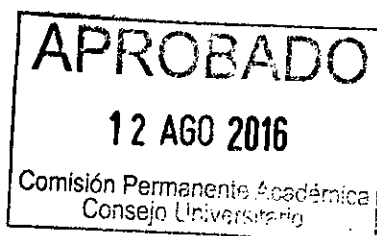
Esta asignatura tiene como propósito aportar en la formación de los estudiantes, el marco histórico, las bases teóricas, así como la identificación de métodos, estrategias y modelos, que servirán de base en las áreas de conocimiento citadas anteriormente.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Fundamentos de Ingeniería de Software se relaciona con las asignaturas: Requisitos de Software, Diseño de Software, Construcción de Software, Mantenimiento de Software, Verificación y Validación de Software, Administración de Proyectos I, Administración de Proyectos II, Aseguramiento de la Calidad del Software y Experimentación en Ingeniería de Software. La asignatura contribuye al desarrollo de todas las competencias de egreso.

4. COMPETENCIA DE LA ASIGNATURA

Analiza las características principales de los métodos, técnicas, procedimientos y buenas prácticas, utilizados en los procesos de desarrollo y gestión del software, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.



5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

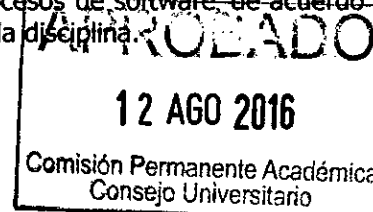
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa
- Toma decisiones en su práctica profesional y personal, de manera responsable.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.

Específicas

- Analiza el desarrollo de la Ingeniería de Software como disciplina profesional de acuerdo con el marco teórico.
- Analiza las características del perfil del ingeniero de software, de acuerdo con modelos curriculares reconocidos.
- Analiza los principales métodos técnicas, procedimientos y buenas prácticas utilizados en las fases de requisitos, diseño, desarrollo, pruebas y mantenimiento, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.
- Discrimina el Modelo de Ciclo de Vida del Software en el inicio de un proyecto, de acuerdo con las características del equipo de desarrollo, de los clientes y de la problemática.
- Analiza los principales métodos técnicas, procedimientos y buenas prácticas utilizados en los procesos de estimación, planificación, seguimiento, control, calidad y configuración, de acuerdo con el cuerpo de conocimientos reconocido por la disciplina.
- Identifica los factores humanos que inciden en el éxito o fracaso de los procesos de software, de acuerdo con el marco teórico de la disciplina.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Evolución de la Ingeniería de Software
- El Perfil del Ingeniero de Software
- Áreas de Desarrollo Software
- Modelos de Ciclo de Vida del Software
- Áreas de Gestión Software
- Factores Humanos en Ingeniería de Software

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje autónomo y reflexivo
- Aprendizaje Colaborativo
- Investigación documental
- Seminarios
- Estudio de casos

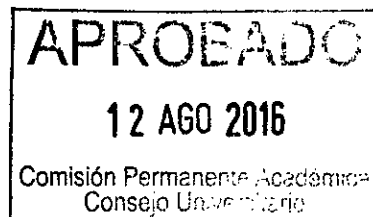
8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Resolución de situaciones problema
- Elaboración de reportes

Evaluación de producto – 30%

- Portafolio de evidencias

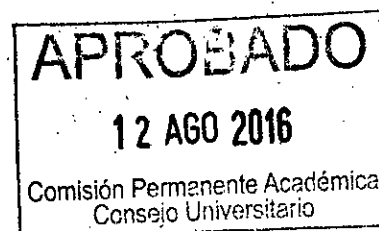


9. REFERENCIAS

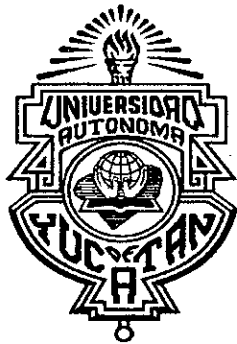
1. Bourque, P. & Firley, R. (2014) *Guide to the Software Engineering Body of Knowledge. SWEBOOK V3.0*. USA: IEEE Computer Society Press.
2. Cuevas G., Amescua, A., Cerrada, J., San-Feliu, T., Calvo-Manzano, J., Arcilla, M. y García, M. (2002) *Gestión del Proceso Software*. España: Editorial CEURA.
3. Pressman, R. & Maxim, B. (2015) *Software Engineering: A Practitioner's Approach*. (8th ed.). USA: Mc Graw Hill.
4. Sánchez, S., Sicilia, M. y Rodríguez, M. (2012) *Ingeniería del Software. Un enfoque desde la guía del SWEBOOK*. España: Alfaomega-Garceta.
5. Sommerville, I. (2011) *Software Engineering*. (9th ed.). USA: Pearson Education/Addison-Wesley.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, o Profesional de TI con posgrado en el área de la Ingeniería de Software.
- Mínimo dos años de experiencia profesional en proyectos de Desarrollo o de Investigación en el área de la Ingeniería de Software.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Geometría Analítica

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Geometría Analítica		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Primer Semestre		
e. Duración total en horas	128	Horas presenciales 64	Horas no presenciales 64
f. Créditos	8		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de la trigonometría y los lugares geométricos básicos, así como las ecuaciones que los representan, es importante para la formación de los estudiantes de Ingeniería de Software, ya que les permitirá tener las bases necesarias para cursar asignaturas de las áreas de Probabilidad, Estadística y Cálculo.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Geometría Analítica se relaciona con las asignaturas Matemáticas Discretas, Cálculo Diferencial, Cálculo Integral y Álgebra Lineal; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

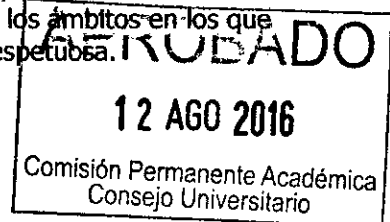
4. COMPETENCIA DE LA ASIGNATURA

Resuelve problemas en diversos contextos, a partir de la aplicación de propiedades trigonométricas y geométricas fundamentales de lugares geométricos en el plano, utilizando los métodos adecuados de manera eficiente.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento, en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.



Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.

Específicas

- Aplica las propiedades fundamentales de las funciones trigonométricas en la resolución de problema.
- Maneja las ecuaciones y gráficas asociadas a líneas rectas de manera fundamentada.
- Maneja las ecuaciones y gráficas asociadas a circunferencias de manera fundamentada.
- Maneja las ecuaciones y gráficas asociadas a las cónicas de manera fundamentada.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Sistema cartesiano y trigonometría
- Funciones y ecuaciones trigonométricas
- Rectas en el plano
- Circunferencia
- Parábola
- Elipse
- Hipérbola

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Elaboración de proyectos
- Aprendizaje basado en problemas
- Aprendizaje colaborativo
- Debates
- Aprendizaje autónomo y reflexivo



8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso -70%

- Pruebas de desempeño
- Investigaciones documentales

Evaluación de producto -30%

- Elaboración de proyectos especiales

9. REFERENCIAS

1. De Oteyza, E. (2005). *Geometría Analítica*. México: Pearson Educación. (Clásico)
2. Gersting, J. (2010). *Technical Calculus with Analytic Geometry*. USA: Dover Publications.
3. Kindle, J. (1991). *Geometría Analítica*. México: McGraw-Hill. (Clásico)
4. Lehmann, C. (2005). *Geometría Analítica*. México: Limusa. (Clásico)
5. Swokowsky, E.; Cole, J. & Romo, R. (2009). *Álgebra y Trigonometría con Geometría Analítica*. México: Cengage Learning Editores S.A. de C.V.
6. Swokowski, E. & Cole, J. (2011). *Precalculus: Functions and Graphs*. Canada: Cengage Learning.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o área afín, preferentemente con posgrado en Matemáticas.
- Mínimo dos años de experiencia profesional.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Inferencia Estadística

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

Nombre de la asignatura	Inferencia Estadística		
Tipo	Obligatoria		
Modalidad	Mixta		
Ubicación sugerida	Quinto semestre		
Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
Créditos	7		
Requisitos académicos	Haber acreditado la asignatura Probabilidad.		

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La Inferencia Estadística es una disciplina que resulta cada vez más necesaria para la formación de los estudiantes de Ingeniería de Software, ya que les permite realizar generalizaciones que contribuyen en la toma de decisiones con base en la información obtenida de una muestra. La teoría de inferencia estadística concatena la teoría de probabilidad con la de estadística. El propósito de esta asignatura es proporcionar las técnicas descriptivas más conocidas, los fundamentos teóricos de la Inferencia Estadística y los procesos para la estimación de parámetros y pruebas de hipótesis estadísticas, que permitan resolver problemas prácticos relacionados con la ingeniería de software.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Inferencia Estadística se relaciona con las asignaturas: Álgebra Intermedia, Álgebra Superior, Álgebra Lineal, Cálculo Diferencial, Cálculo Integral, Probabilidad, Métricas de Software y Experimentación en Ingeniería de Software, contribuyendo al logro de todas las competencias de egreso.

4. COMPETENCIA DE LA ASIGNATURA

Aplica la metodología de estimación y prueba de hipótesis para uno o dos parámetros en problemas prácticos, relacionados con la investigación en ingeniería de software.

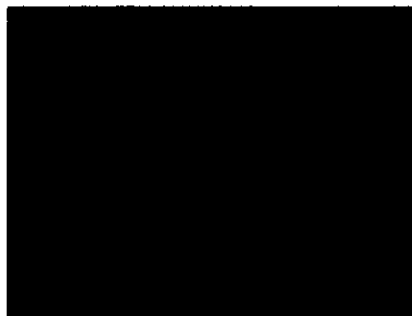
5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Aplica los conocimientos en sus intervenciones.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





- profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.



- Obtiene de manera correcta las características descriptivas de un conjunto de datos, mediante el uso de tablas y gráficas; medidas de tendencia central, dispersión y forma.
- Deduce de manera eficiente las distribuciones muestrales relacionadas con los parámetros más comunes utilizadas en estimación y pruebas de hipótesis.
- Obtiene eficientemente estimadores puntuales relacionados con los parámetros más comunes utilizadas en estimación y pruebas de hipótesis.
- Verifica correctamente las propiedades de un estimador puntual en problemas relacionados con estimación.
- Obtiene correctamente estimadores por intervalos de confianza relacionados con uno o dos parámetros a través de problemas de estimación.
- Interpreta adecuadamente la estimación por intervalos para uno o dos parámetros en problemas prácticos.
- Resuelve problemas prácticos relacionados con la ingeniería de software, a través de la interpretación de los resultados obtenidos en una prueba de hipótesis para uno o dos parámetros.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Análisis Exploratorio de Datos.
- Distribuciones Muestrales.
- Estimación Puntual.
- Estimación por Intervalos.
- Pruebas de Hipótesis.
- Métodos no paramétricos.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

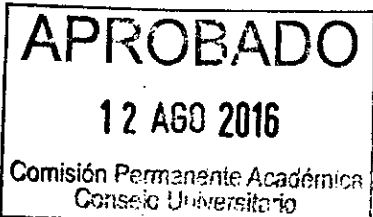
- Resolución de problemas y ejercicios
- Aprendizaje autónomo y reflexivo
- Aprendizaje cooperativo
- Discusión dirigida
- Aprendizaje basado en proyectos problema
- Aprendizaje mediado por las TIC
- Investigación documental

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso – 80%

Evaluación de producto – 20%

- Solución de ejercicios y problemas
- Pruebas de desempeño
- Realización de proyectos prácticos
- Presentación oral ante el grupo
- Revisión de fuentes de información documental
- Prueba de desempeño global

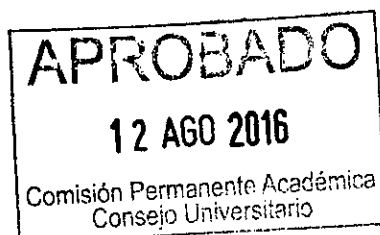


9. REFERENCIAS

1. Conover, W. J. (1999). *Practical nonparametric statistics*. (3ª ed.). New York: John Wiley and Sons. (Clásico).
2. Milton, S.J. & Arnold, J.C. (2004). *Probabilidad y estadística: con aplicaciones para ingeniería y ciencias computacionales*. (4ª ed.) México: McGraw-Hill.
3. Mood, A.M., Graybill, F.A. & Boes, D.C. (1974). *Introduction to the Theory of Statistics*. (3ª ed.). New York: Mc Graw-Hill. (Clásico)
4. Wackerly, D.D., Mendenhall, W. & Scheaffer, R.L. (2010). *Estadística Matemática con Aplicaciones*. (7ª ed.). México: CENGAGE Learning.
5. Walpole, R.E., Myers, R.H., Myers, S.L. & Ye, K. (2012). *Probabilidad y Estadística Para Ingeniería y Ciencia* (9ª ed.). México: Pearson.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o afín, preferentemente con posgrado en el área de Probabilidad y Estadística.
- Mínimo dos años de experiencia profesional, en investigación o trabajo en el área de la estadística.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

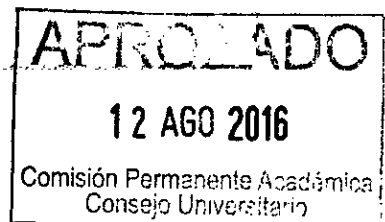


Innovación Tecnológica

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Innovación Tecnológica		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Séptimo semestre		
e. Duración total en horas	96	Horas presenciales 64	Horas no presenciales 32
f. Créditos	6		
g. Requisitos académicos previos	Ninguno		



2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de la Innovación Tecnológica es importante para la formación de los estudiantes de Ingeniería de Software, ya que les permitirá realizar proyectos eficientes y sustentables que promuevan la innovación tecnológica de los procesos de software.

El propósito de esta asignatura es aportar los conceptos y herramientas relacionados con la Innovación Tecnológica, para mejorar los procesos de desarrollo, mantenimiento y administración del software.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Innovación Tecnológica se relaciona con las asignaturas Administración de Proyectos I, Administración de Proyectos II, Experimentación en Ingeniería de Software, Responsabilidad Social Universitaria y Taller de Emprendedores; ya que contribuyen al logro de la competencia de egreso Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Realiza proyectos de ingeniería de software, promoviendo la innovación tecnológica y el desarrollo sostenible.

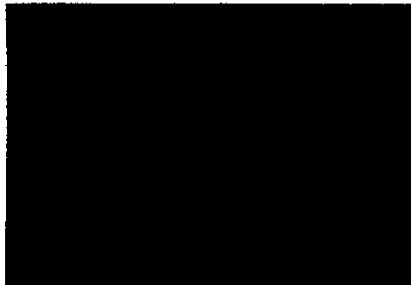
5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Interviene con iniciativa y espíritu emprendedor en su

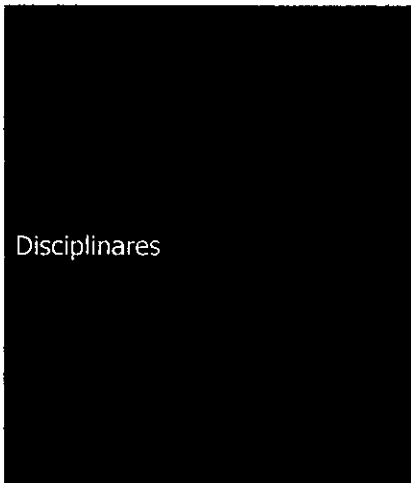
APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





ejercicio profesional y personal de forma autónoma y permanente.

- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.



Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

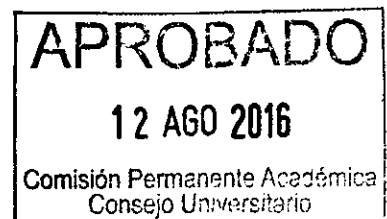


Específicas

- Aplica el proceso de administración estratégica, para la mejora de la competitividad y la toma de decisiones en una organización.
- Diseña estrategias de innovación tecnológica orientadas hacia la sustentabilidad de las organizaciones, para la mejora de los procesos de desarrollo, mantenimiento y administración del software.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Administración estratégica
- Administración de la innovación tecnológica
- Medición del desempeño innovador
- Creación y sostenimiento de culturas organizacionales innovadoras
- Innovación y sustentabilidad como generadores de conocimientos



7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Aprendizaje autónomo y reflexivo
- Aprendizaje cooperativo
- Aprendizaje mediado por las TIC
- Prácticas en laboratorio

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Desarrollo de proyectos
- Resolución de casos
- Elaboración de reportes
- Resolución de situaciones problema

Evaluación de producto – 30%

- Proyecto integrador

9. REFERENCIAS

1. Hill, C. W. y Jones, G. R. (2011). *Administración Estratégica: Un Enfoque Integral*. (9ª ed.). México: CENGAGE Learning.
2. Ahmed, P. K., Shepherd, C. D., Garza, L. R. y Garza, C. R. (2012). *Administración de la Innovación*. México: Pearson.
3. Betz, F. (2011). *Managing Technological Innovation* (3ª ed.). EUA: Wiley.



10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado en el área de las Tecnologías de la Información.
- Mínimo tres años de experiencia profesional o de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Interacción Humano Computadora

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Interacción Humano Computadora		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Sexto Semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La asignatura de Interacción Humano Computadora es importante en la formación del Ingeniero de Software porque contribuye a desarrollar habilidades para el diseño de prototipos de sistemas interactivos considerando los aspectos de usabilidad fundamentales.

Se abordan conceptos y aspectos clave respecto a la interacción humano computadora tales como el diseño centrado en el usuario, el análisis y diseño de interfaces. Estos elementos son imprescindibles porque contribuyen en aspectos esenciales para garantizar la calidad de un producto de software.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

La asignatura de Interacción Humano Computadora se relaciona con las asignaturas Diseño de Software, Desarrollo de Aplicaciones Web, Métricas de Software, Requisitos de Software, Administración de Proyectos I, Administración de Proyectos II, Innovación Tecnológica. En conjunto contribuyen al desarrollo de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla sistemas de software que proporcione interacciones adecuadas para el usuario, considerando metodologías y principios básicos de diseño de interfaces en sistemas interactivos.



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.

Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Utiliza los conceptos de la interacción humano computadora, para el diseño de sistemas interactivos de acuerdo con los marcos de referencia propios de la disciplina.
- Implementa prototipos de sistemas interactivos, considerando los principios básicos del diseño centrado en el usuario de manera fundamentada.
- Construye interfaces de usuario de sistemas interactivos, considerando las guías básicas del diseño de forma efectiva y eficiente.
- Implementa la evaluación de interfaces, considerando los requerimientos no funcionales de un sistema que garanticen la usabilidad del sistema interactivo.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Principios Básicos de Interacción Humano Computadora
- Diseño Centrado en el Usuario
- Análisis y Diseño de Interfaces
- Usabilidad

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje cooperativo
- Resolución de problemas y ejercicios
- Investigación documental
- Proyectos de investigación
- Aprendizaje orientado a proyectos
- Aprendizaje mediado por las TIC
- Aprendizaje autónomo y reflexivo

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Resolución de casos
- Investigación documental
- Resolución de situaciones problema
- Pruebas de desempeño
- Desarrollo de proyectos

Evaluación de producto - 30%

- Desarrollo de proyectos
- Portafolio de evidencias

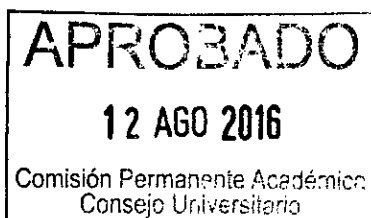


9. REFERENCIAS

1. Baxter, K. y Courage, C. (2005). *Understanding your users: A practical guide to user Requirements Methods Tools and Techniques (interactive technologies)*. USA: Morgan Kaufmann Publishers. (Clásico)
2. Dumas, J. y Redish, J. (1999). *A practical guide to usability testing*. USA: Intellect Books. (Clásico).
3. Goodman, E.; Kuniavsky, M. y Moed, A. (2003). *Observing the user experience: a practitioner's guide to user research*. (2a ed.). USA: Morgan Kaufmann Publishers. (Clásico).
4. Monk, A. (1985). *Fundamentals of human-computer interaction*. USA: Academic Press. (Clásico).
5. Preece, J. y Rogers, Y. (1994). *Human-computer interaction*. UK: Addison-Wesley. (Clásico).
6. Rubin, J. y Chisnell, D. (2008). *Handbook of usability testing: how to plan, design, and conduct effective tests*. (2a ed.). USA: Wiley.
7. Shneiderman, B. y Catherine, P. (2009). *Designing the user interface: strategies for effective human-computer interaction*. (5a ed.). USA: Addison Wesley.

10. PERFIL DESEABLE DEL PROFESOR

- Formación profesional en Ingeniería de Software o afin, preferentemente con estudios de posgrado.
- Mínimo dos años de experiencia profesional de trabajo en el área
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



Licenciatura en Ingeniería de Software



Mantenimiento de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Mantenimiento de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Octavo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El mantenimiento del software constituye una parte fundamental en el desarrollo de sistemas y consiste en la modificación de un producto de software después de su entrega al cliente o usuario, ya sea para corregir defectos, mejorar el rendimiento o adaptarlo a un cambio de entorno. La asignatura tiene como propósito proporcionar los principios específicos de la Ingeniería de Software para realizar el mantenimiento adecuado a un sistema informático, ya sea de tipo correctivo, adaptativo, perfectivo o preventivo.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Mantenimiento de Software se relaciona con las asignaturas Diseño de Software, Administración de Proyectos I y Administración de Proyectos II, Programación Orientada a Objetos, Algoritmia, Estructuras de Datos y Programación Estructurada, ya que contribuyen al logro de la competencia de egreso:

- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Realiza el desarrollo, mantenimiento y actualización de sistemas de software, utilizando metodologías que aseguren altos estándares de calidad.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

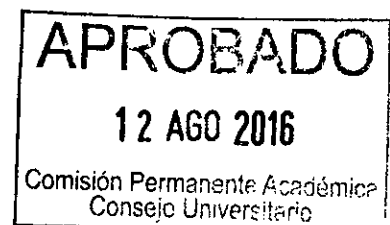
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Trabaja bajo presión de manera eficaz y eficiente.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo

Específicas

- Aplica conceptos y técnicas de reestructuración en el desarrollo de sistemas de software que incrementen la calidad, con apego a los estándares vigentes.
- Elabora un plan de integración de los componentes de un sistema de software, de manera pertinente.
- Elabora un plan de mantenimiento de un sistema de software, expresado con claridad y precisión.
- Repara diversos tipos de errores presentes en un sistema de software, aplicando metodologías de desarrollo de software de manera eficaz y eficiente.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al mantenimiento de software
- Modelos y normas de calidad
- Evaluación y mejora de procesos de desarrollo
- Análisis y reestructuración de sistemas

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de Casos
- Aprendizaje orientado a proyectos.
- Aprendizaje en escenarios reales.
- Prácticas en laboratorio.
- Aprendizaje cooperativo.

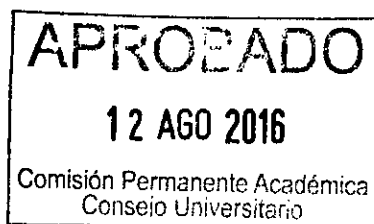
8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Prácticas de laboratorio.
- Pruebas de desempeño.
- Resolución de casos.
- Elaboración de reportes.

Evaluación de producto – 30%

- Desarrollo de proyectos.

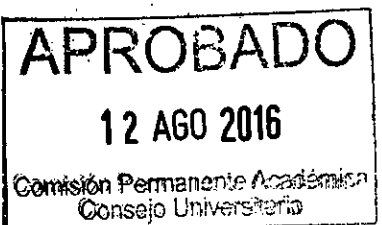


9. REFERENCIAS

1. Alain, A. y Alain, A. (2008). *Software Maintenance Management: Evaluation and Continuous Improvement*. USA: Wiley-IEEE Computer Society Press.
2. Genero, M.; Cruz, J. y Piattini, M. (2014). *Métodos de investigación en ingeniería del software*. España: Alfaomega RA-MA.
3. Kendall, K. y Kendall, J. (2011). *Análisis y diseño de sistemas*. (8ª. ed.). México, DF: Pearson Educación.
4. Piattini, M. y Garzás, J. (2010). *Fábricas de software: experiencias, tecnologías y organización*. España: RA-MA S.A. Editorial y Publicaciones.
5. Piattini, M, García, F., García, I. y Pino, F. (2011). *Calidad de Sistemas de información*. (2a ed.). Alfaomega RA-MA Editorial.
6. Pfleeger, S. y Atlee, J. (2010). *Software engineering: Theory and practice* (4nd. ed.). Upper Saddle River, N.J.: Prentice Hall.
7. Pressman, R. (2005). *Software engineering: A practitioner's approach* (6th. ed.). Boston, Mass: McGraw-Hill.
8. Reifer, D. (2012). *Software Maintenance Success Recipes*. United States of America: CRC Press.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o afín, preferentemente con posgrado en el área.
- Mínimo 3 años de experiencia profesional de trabajo en el área.
- Mínimo 2 años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Matemáticas Discretas

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Matemáticas Discretas		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Segundo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El contexto del desarrollo científico y tecnológico de la humanidad está determinado por la posibilidad de describir las propiedades de los objetos reales, mediante el lenguaje vigoroso de la matemática. Ante esto, la Matemática Discreta permite entender, inferir, aplicar y desarrollar modelos matemáticos tendientes a resolver problemas en el área de las ciencias computacionales.

Matemáticas Discretas aporta al estudiante las bases para el análisis de procesos computacionales, tales como seguridad, encriptación, gráficas de computadora, inteligencia artificial, minería de datos y teoría de lenguajes de programación.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Matemáticas Discretas se relaciona con las asignaturas Programación Estructurada, Programación Orientado a Objetos, Teoría de la Computación, Sistemas Operativos y Estructuras de Datos; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Representa procesos y estructuras computacionales mediante el uso de la teoría algebraica, recurrencias y grafos.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

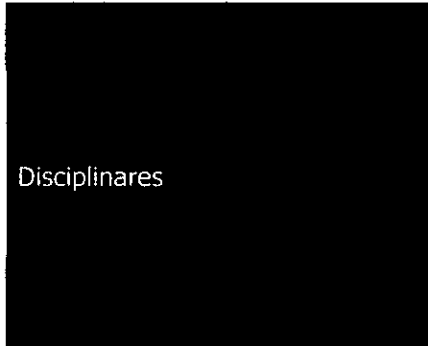
Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.

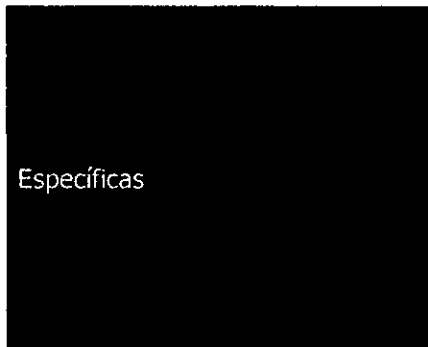




- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.
- Trabaja bajo presión de manera eficaz y eficientemente.



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.



- Representa estructuras computacionales, utilizando la lógica matemática con argumentos congruentes y lógicos.
- Utiliza las funciones recursivas para el procesamiento de estructuras computacionales, de manera fundamentada.
- Utiliza las funciones enteras para el procesamiento de estructuras computacionales, de manera fundamentada.
- Representa estructuras computaciones, utilizando la teoría de grafos de manera fundamentada y eficiente.
- Representa estructuras computacionales, utilizando la teoría de árboles de manera fundamentada y eficiente.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Lógica matemática
- Funciones recursivas
- Funciones enteras
- Teoría de grafos
- Teoría de árboles

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Aprendizaje orientado a proyectos
- Aprendizaje mediado por las TIC
- Aprendizaje autónomo y reflexivo

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



- Aprendizaje cooperativo
- Resolución de problemas y ejercicios
- Investigación documental

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Investigación documental
- Resolución de situaciones problema
- Pruebas de desempeño
- Desarrollo de proyectos

Evaluación de producto – 30%

- Desarrollo de proyectos
- Portafolio de evidencias

9. REFERENCIAS

1. Epp, S. S. (2011). *Discrete Mathematics with Applications*. (4th ed.). USA: Cengage Learning.
2. Gersting, J. L. (2014). *Mathematical Structures for Computer Science*. (7th ed.). USA: W. H. Freeman.
3. Grimaldi, R. P. (2006). *Discrete and Combinatorial Mathematics*. (5th ed.). USA: Pearson, Addison Wesley. (Clásico)
4. Rosen, K. (2012). *Discrete Mathematics and Its Applications*. (7th ed.). USA: McGraw-Hill Education.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado.
- Mínimo un año de experiencia profesional.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que se va a impartir.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Métricas de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Métricas de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Sexto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de las Métricas de Software es importante para la formación del Ingeniero de Software, ya que le ofrecerá un panorama de la medición en el ámbito de la Ingeniería de Software.

El propósito de esta asignatura es aportar una visión general e integradora de la disciplina, tanto desde el punto de vista teórico como práctico. Como resultado del curso, se espera que el alumno analice la relevancia de las métricas de software, identifique sus características principales, sea capaz de aplicarlas para estimar y dar seguimiento puntual del progreso de un proyecto de software; y utilice estándares, metodologías y herramientas de medición.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Métricas de Software se relaciona con las asignaturas Inferencia Estadística, Experimentación en Ingeniería de Software, Administración de Proyectos I, Administración de Proyectos II y Aseguramiento de la Calidad del Software; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Evalúa proyectos de software de manera sistemática, disciplinada y cuantificable, aplicando métricas, estándares, metodologías y herramientas de medición.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus





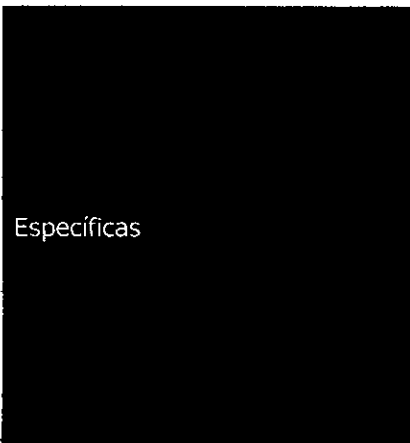
intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.



Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



Específicas

- Aplica de forma correcta los fundamentos de medición en el ciclo de vida del software.
- Aplica un conjunto de métricas significativas, para la mejora de la calidad y la productividad en el desarrollo y mantenimiento del software.
- Predice el esfuerzo, la duración y el costo de los proyectos de software, por medio de técnicas y herramientas de estimación.
- Aplica de forma correcta los fundamentos de la experimentación en la medición del software.
- Utiliza estándares, metodologías y herramientas de medición, para la mejora continua y sistemática de los proyectos de software.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Fundamentos de medición
- Tipos de escalas de medición
- Medición significativa
- Métricas del proceso
- Métricas del proyecto
- Métricas del producto
- Métodos de estimación
- Estimación del tamaño del software
- Estimación del esfuerzo, duración y costo
- Análisis estadístico de la medición del software
- Estándares y metodologías de medición
- Herramientas de medición

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje autónomo y reflexivo
- Aprendizaje cooperativo
- Aprendizaje orientado a proyectos
- Aprendizaje mediado por las TIC
- Estudio de casos
- Investigación documental
- Prácticas en laboratorio
- Resolución de problemas y ejercicios

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 80%

- Pruebas de desempeño
- Resolución de problemas y ejercicios

Evaluación de producto – 20%

- Proyecto integrador

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

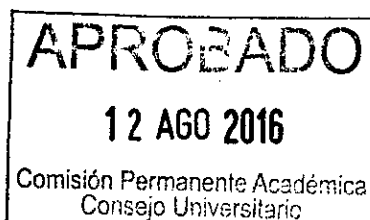


9. REFERENCIAS

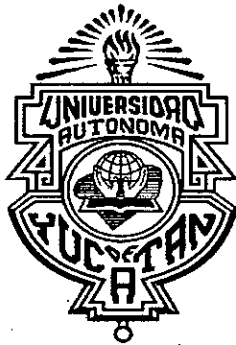
1. Dolado, J. J., Fernández, L. (2000). *Medición para la Gestión en la Ingeniería del Software*. España: Ra-Ma.
2. Fenton, N. E. y Pfleeger, S. L. (1997). *Software Metrics: A Rigorous and Practical Approach* (2ª ed.). Boston, M.A: PWS Publishing Company.
3. Florac, W. A. y Carleton, A. D. (1999). *Measuring the Software Process: Statistical Process Control for Software Improvement*. E. U. A.: Addison-Wesley.
4. Genero, M., Piattini, M. y Calero, C. (Eds.). (2005). *Metrics for Software Conceptual Models*. Londres, GB: Imperial College Press.
5. Garmus, D. y Herron, D. (2001). *Function Point Analysis: Measurement Practices for Successful Software Projects*. Upper Saddle River, N.J: Addison-Wesley.
6. Henderson-Sellers, B. (1996). *Object-Oriented Metrics: Measures of Complexity*. Upper Saddle River, N.J: Prentice-Hall.
7. Laird, L. M., Brennan, M. C. (2006). *Software Measurement and Estimation: A Practical Approach*. E. U. A: IEEE Computer Society-Wiley-Interscience.
8. Lorenz, M. y Kidd, J. (1994). *Object-Oriented Software Metrics: A Practical Guide*. E.U.A: Prentice-Hall.
9. McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. E.U.A: Microsoft Press.
10. Milton, S.J. y Arnold, J.C. (2004). *Probabilidad y Estadística: con Aplicaciones para Ingeniería y Ciencias Computacionales* (4ª ed.). México, D.F: McGraw-Hill Interamericana.
11. Montgomery, D.C. (2006). *Control Estadístico de la Calidad* (3ª ed.). México, D.F.: Limusa Wiley.
12. Parthasarathy, M. A. (2007). *Practical Software Estimation: Function Point Methods for Insourced and Outsourced Projects*. E.U.A: Addison-Wesley.
13. Wackerly, D.D., Mendenhall, W. y Scheaffer, R.L. (2010). *Estadística Matemática con Aplicaciones* (7ª ed.). México, D.F: Cengage Learning.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado.
- Mínimo tres años de experiencia profesional o de trabajo en el área de Ingeniería de Software.
- Mínimo tres años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Probabilidad

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Probabilidad		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Cuarto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de la probabilidad es importante para la formación de los estudiantes de Ingeniería de Software, ya que le aporta los elementos básicos de probabilidad que fundamentan las técnicas para analizar datos, realizar estimaciones y predicciones que son utilizadas en el desarrollo, mantenimiento, administración de los procesos de software así como en las propuestas de innovación.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Probabilidad se relaciona con las asignaturas Álgebra Superior, Cálculo Diferencial, Cálculo Integral, Álgebra Lineal, Inferencia Estadística, Métricas de Software y Experimentación en Ingeniería en Software; contribuyendo al logro de todas las competencias de egreso.

4. COMPETENCIA DE LA ASIGNATURA

Resuelve problemas de naturaleza aleatoria presentes en las diferentes áreas del conocimiento, utilizando la teoría básica de la probabilidad.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Gestiona el conocimiento en sus intervenciones profesionales con rigor científico.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

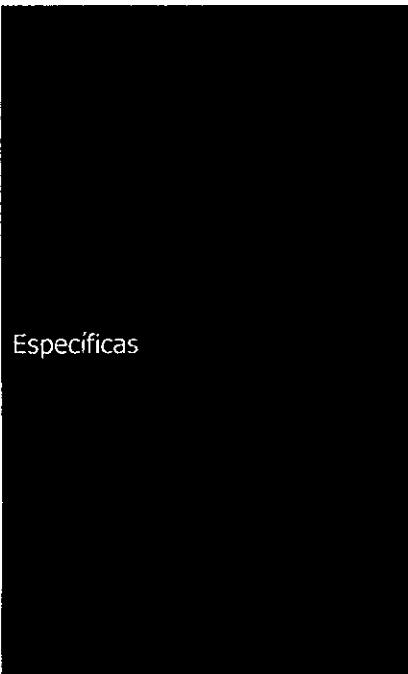




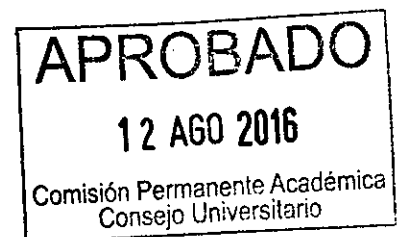
- Desarrolla su pensamiento, en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas a nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y de cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.



- Resuelve problemas relacionados con el cálculo de probabilidades con eficiencia, utilizando las distribuciones de probabilidades discretas y continuas de variables comunes y no comunes.
- Utiliza las propiedades de un modelo de probabilidad en el análisis y solución de problemas de naturaleza aleatoria, en forma eficiente y adecuada.
- Calcula probabilidades de eventos aleatorios de manera eficiente, utilizando los teoremas principales derivados de la probabilidad axiomática.
- Utiliza de manera eficiente las tablas de probabilidades relacionadas con las distribuciones normal, t de student, Chi-Cuadrada, F de Fisher, Binomial y Poisson.
- Determina la distribución de probabilidad y los momentos centrales de una variable aleatoria a través de la función generadora de momentos, de manera fundamentada.
- Determina de manera eficiente la distribución de una función de una y dos variables aleatorias a través de los métodos más comunes.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Conceptos básicos de probabilidad
- Variables aleatorias discretas y continuas
- Familias paramétricas especiales de distribución univariada
- Función generadora de momentos
- Vectores aleatorios
- Funciones de variables aleatorias

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de ejercicios y problemas
- Aprendizaje colaborativo en grupos de discusión
- Aprendizaje autónomo y reflexivo mediante preguntas en clases

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 75%

- Pruebas de desempeño
- Resolución de situaciones problema

Evaluación de producto – 25%

- Pruebas de desempeño global

9. REFERENCIAS

1. Mendenhall, W., & Wackerly, D.D. (2009). *Estadística Matemática con Aplicaciones*. (7ª ed.). México: Cengage Learning.
2. Mendenhall, W.; Beaver, R.; & Beaver, M. (2015). *Introducción a la probabilidad y estadística*. (14a ed.). México: Cengage Learning.
3. Milton, J. & Arnold, J. (2004). *Probabilidad y Estadística con Aplicaciones para Ingeniería y Ciencias Computacionales*. (4a ed.). México: Mc. Graw Hill.
4. Ross, S.M. (2012). *A first course in probability*. (9a ed.). USA: Pearson Education.

APROBADO

12 AGO 2016

Comisión Permanente de
Consejo Universitario



5. Walpole, R.E., Myers, R.H., Myers, S.L., & Ye, K. (2012). *Probability & Statistics for Engineers & Scientists*. (9a ed.). USA: Pearson Education.

10. PERFIL DESEABLE DEL PROFESOR

- Licenciado en Matemáticas o afín con Maestría en Probabilidad y/o Estadística
- Mínimo un año de experiencia profesional
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Prácticas Profesionales

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Presencial

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Prácticas Profesionales		
b. Tipo	Obligatoria		
c. Modalidad	Presencial		
d. Ubicación sugerida	Sexto semestre		
e. Duración total en horas	320	Horas presenciales	320
		Horas no presenciales	0
f. Créditos	8		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Haber obtenido al menos 180 créditos del Plan de Estudios de la Licenciatura en Ingeniería de Software

2. JUSTIFICACIÓN DE LA PRÁCTICA PROFESIONAL DENTRO DEL PE

Las Prácticas Profesionales son importantes para los estudiantes de la Licenciatura en Ingeniería de Software porque representan el escenario idóneo para aplicar las competencias adquiridas vinculándose con el entorno social y productivo.

Las Prácticas Profesionales contribuyen a la formación integral del estudiante a través del ejercicio de contraste entre el conocimiento teórico y la experiencia práctica que la realidad profesional proporciona, permitiéndole intervenir en la solución de problemas o situaciones que el ámbito laboral demanda.

3. COMPETENCIAS DE EGRESO QUE SE FAVORECERÁN CON LA PRÁCTICA

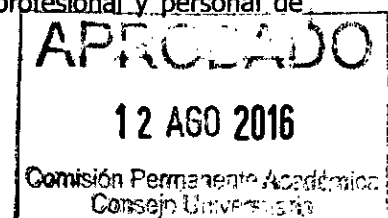
Las Prácticas Profesionales contribuyen al logro de las cuatro competencias de egreso.

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.
- Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.
- Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, mediante fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.

4. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS QUE SE PROMOVERÁN CON LA PRÁCTICA PROFESIONAL

Competencias genéricas:

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Interviene con iniciativa y espíritu emprendedor en su ejercicio profesional y personal de forma autónoma y permanente.



- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve de manera transparente y ética.
- Trabaja bajo presión de manera eficaz y eficientemente.

Competencias Disciplinarias:

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Competencias específicas:

- Depende de la institución donde realizará las prácticas profesionales.

5. ESTRATEGIAS PARA LA GESTIÓN DE LOS ESCENARIOS REALES DE APRENDIZAJE

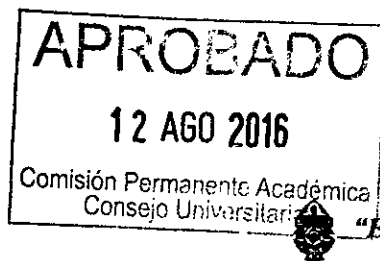
Establecer acuerdos con el sector público y privado para facilitar el acceso de los estudiantes a los escenarios reales de aprendizaje.

6. ESTRATEGIAS DE ACOMPAÑAMIENTO PARA LA MOVILIZACIÓN Y EL DESARROLLO DE COMPETENCIAS

- Plan de trabajo de las actividades a desarrollar.
- Reporte de las actividades realizadas.

7. ESTRATEGIAS GENERALES DE EVALUACIÓN

- Evaluación por la institución receptora.
- Elaboración del reporte final.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Programación Estructurada

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Programación Estructurada		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Segundo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Programación Estructurada es importante para la formación de los estudiantes de Ingeniería de Software, ya que permite desarrollar programas de cómputo básicos utilizando estructuras de secuencia, selección, iteración, así como llamadas y creación de subrutinas, con base el procesamiento algorítmico.

Esta asignatura tiene como propósito, aportar los paradigmas y las técnicas principales de programación para su aplicación en el diseño, desarrollo e implantación de sistemas computacionales.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Programación Estructurada se relaciona con las asignaturas Algoritmia, Fundamentos de Ingeniería de Software, Programación Orientada a Objetos, Teoría de la Computación, Arquitectura y Organización de Computadoras, Diseño de Software, Estructuras de Datos, Sistemas Operativos, Teoría de Lenguajes de Programación, Arquitecturas de Software, Construcción de Software, Diseño de Bases de Datos, Desarrollo de Aplicaciones Web, Métricas de Software, Aseguramiento de la Calidad del Software, Requisitos de Software, Interacción Humano Computadora, Experimentación en Ingeniería de Software, Verificación y Validación de Software, Redes y Seguridad de Computadoras, Innovación Tecnológica, Administración de Proyectos I, Mantenimiento de Software, Sistemas Distribuidos y Administración de Proyectos II; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla aplicaciones computacionales eficientes, fundamentado en las metodologías del paradigma de la programación estructurada.

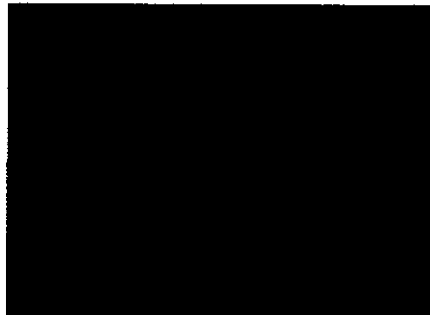
5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

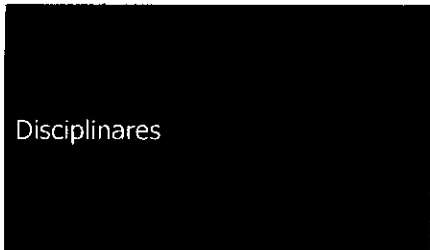
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento en intervenciones profesionales

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario

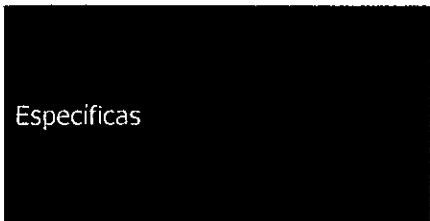




- y personales, de manera crítica, reflexiva y creativa
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Resuelve problemas en contextos locales, nacionales e internacionales, de manera profesional.
- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.
- Trabaja bajo presión de manera eficaz y eficientemente.



- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



- Codifica algoritmos por medio de un lenguaje de programación estructurado, de manera eficiente y eficaz.
- Utiliza sentencias de programación adecuadas, para las aplicaciones computacionales.
- Identifica los diferentes tipos de datos en los lenguajes de programación, para la creación de aplicaciones óptimas.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Variables, operadores y expresiones
- Estructuras de control y arreglos
- Paso parámetros y manejo de funciones
- Alcance de variables
- Cadenas
- Apuntadores y estructuras
- Manejo de memoria dinámica

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Conferencia
- Interrogatorio
- Grupos de discusión
- Resolución de problemas de programación

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



- Trabajo en equipo
- Desarrollo de programas de cómputo
- Investigación bibliográfica

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso – 70%

- Pruebas de desempeño
- Prácticas de programación

Evaluación de producto – 30%

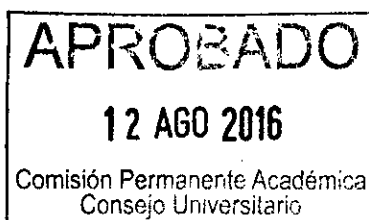
- Pruebas de desempeño
- Proyecto integrador de programación

9. REFERENCIAS

1. Joyanes, L. (2008). *Fundamentos de Programación: algoritmos, estructuras de datos y objetos.* (4ª ed.). España: McGraw Hill
2. Kernighan, B. & Ritchie, D. (2006). *The C Programming Language.* (2nd ed.) China Machine Press. (Clásico)
3. Knuth, D. E. (2011). *The Art of Computer Programming.* (Vols 1-4). Addison-Wesley Professional.
4. Reek, K. A. (1997). *Pointers on C.* USA: Addison-Wesley. (Clásico)
5. Stephen Kochan (2004). *Programming in C.* (3rd ed.). USA: Sams Publishing. (Clásico)

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional, en investigación o trabajo en el área.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Programación Orientada a Objetos

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Programación Orientada a Objetos		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Tercer semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de la Programación Orientada a Objetos es importante para la formación de los estudiantes de Ingeniería de Software, ya que les permitirá desarrollar aplicaciones computacionales que resuelvan problemas de diversos entornos, mediante el uso del paradigma de Programación Orientada a Objetos.

El propósito de esta asignatura es aportar los elementos fundamentales para realizar abstracciones de problemas reales, y desarrollar los modelos y las aplicaciones de software correspondientes en un lenguaje de programación orientado a objetos.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Programación Orientada a Objetos se relaciona con las asignaturas Algoritmia, Fundamentos de Ingeniería de Software, Programación Estructurada, Teoría de la Computación, Arquitectura y Organización de Computadoras, Diseño de Software, Estructuras de Datos, Sistemas Operativos, Teoría de Lenguajes de Programación, Arquitecturas de Software, Construcción de Software, Diseño de Bases de Datos, Desarrollo de Aplicaciones Web, Métricas de Software, Aseguramiento de la Calidad del Software, Requisitos de Software, Interacción Humano Computadora, Experimentación en Ingeniería de Software, Verificación y Validación de Software, Redes y Seguridad de Computadoras, Innovación Tecnológica, Administración de Proyectos I, Mantenimiento de Software, Sistemas Distribuidos y Administración de Proyectos II; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla aplicaciones computacionales eficientes, fundamentado en el paradigma de la programación orientada a objetos.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Aplica los conocimientos en sus intervenciones

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





- profesionales y en su vida personal con pertinencia.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Resuelve problemas en contextos locales, nacionales e internacionales, de manera profesional.
- Trabaja bajo presión de manera eficaz y eficientemente.



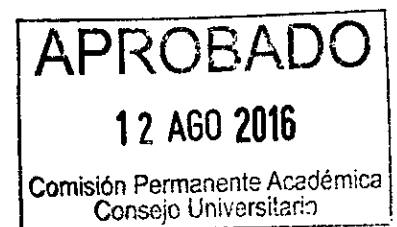
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



- Modela problemas de diversos contextos, aplicando los conceptos básicos del paradigma de Programación Orientada a Objetos.
- Desarrolla programas de software, aplicando de manera óptima los conceptos de extensión y reusabilidad.
- Desarrolla programas de software, mediante el uso de la programación bajo contrato.
- Desarrolla programas de software eficiente, basado en un esquema por capas.
- Desarrolla programas de software tolerantes a fallas, de acuerdo al esquema de manejo de excepciones.
- Implementa la persistencia de datos en programas de cómputo, bajo el paradigma orientado a objetos.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al paradigma de Programación Orientada a Objetos.
- Clases y objetos.
- Extensión y reusabilidad.
- Manejo de errores y excepciones.
- Persistencia.
- Desarrollo por capas.



7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje basado en problemas
- Aprendizaje orientado a proyectos
- Resolución de problemas y ejercicios
- Conferencia
- Interrogatorio.
- Grupos de discusión
- Resolución de problemas de programación en clase y en tareas.
- Aprendizaje colaborativo
- Desarrollo de programas de cómputo
- Investigación bibliográfica

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Prácticas de programación (aplicaciones de software)
- Proyecto de programación
- Pruebas de desempeño

Evaluación de producto – 40%

- Proyecto integrador de programación
- Portafolio de evidencias

9. REFERENCIAS

1. Budd, T. (2002). *An introduction to object-oriented programming* (3a ed.). Boston: Addison-Wesley. (Clásico)
2. Dean, J. y Dean, R. (2009). *Introducción a la programación con Java*. México: McGraw-Hill.
3. Deitel, H. (2014). *Java How To Program (Early Objects)* (10a ed.). Pearson Education Limited.
4. Eckel, B. (2006). *Thinking in Java* (4a ed.). Upper Saddle River, NJ: Prentice Hall. (Clásico)
5. Freeman, E. (2004). *Head First design patterns*. Sebastopol, CA: O'Reilly. (Clásico)
6. Kolling, M. (2012). *Objects First with Java: A Practical Introduction Using BlueJ*. Harlow: Pearson/Education.
7. Sierra, K. y Bates, B. (2005). *Head first Java* (2a ed.). Sebastopol, CA: O'Reilly. (Clásico)
8. Weisfeld, M. (2013). *The object-oriented thought process* (4a ed.). Addison-Wesley Professional.

APROBADO

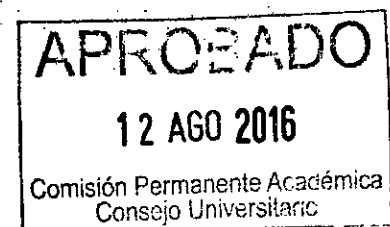
12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

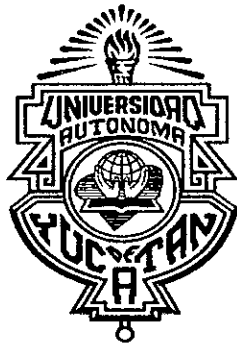


10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afin, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional de trabajo en el área de programación.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Redes y Seguridad de Computadoras

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Redes y Seguridad de Computadoras		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Séptimo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de Redes y Seguridad de Computadoras es importante en la formación del Licenciado en Ingeniería de Software, porque la seguridad es un atributo de calidad de un sistema informático. Esta asignatura permite identificar los riesgos de seguridad de la información en los sistemas que utilizan redes de computadoras como medio de transmisión de datos, reduciendo riesgos e incrementando la calidad del software en cualquiera de las etapas de su ciclo de vida.

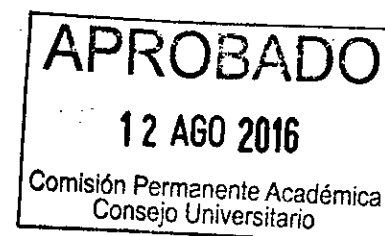
Redes y Seguridad de Computadoras aporta al estudiante las herramientas, técnicas, metodologías y estándares de la seguridad de un sistema informático.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Redes y Seguridad de Computadoras se relaciona con las asignaturas Programación, Sistemas Operativos, Desarrollo Web, Diseño de Bases de Datos, Verificación y Validación de Software, Aseguramiento de la Calidad del Software y Sistemas Distribuidos; ya que contribuyen al logro de las competencias de egreso: Desarrollo de Software, Mantenimiento de Software y Administración de los Procesos de Software.

4. COMPETENCIA DE LA ASIGNATURA

Implementa mecanismos de seguridad en el procesamiento, almacenamiento e intercambio de información en los componentes de un sistema, interconectados a través de redes de comunicación.



**5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE
CONTRIBUYE LA ASIGNATURA**

Genéricas

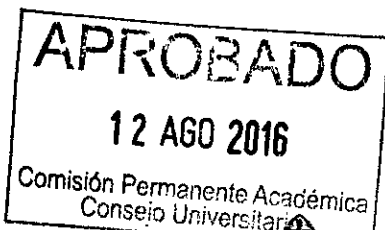
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Describe de forma clara y ordenada los modelos de red OSI y TCP/IP.
- Describe de forma clara y ordenada, los conceptos generales y las características de la evolución de la seguridad informática.
- Utiliza algoritmos de criptografía en los procesos de almacenamiento y comunicación de información, garantizando la confidencialidad e integridad de los datos en un sistema informático.
- Identifica riesgos y amenazas en un sistema informático, a través de un proceso de auditoría de seguridad.



- Selecciona los mecanismos de seguridad informática adecuados, para la reducción de riesgos y amenazas en un sistema informático.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Redes de computadoras
- Seguridad informática
- Seguridad de sistemas operativos
- Seguridad en redes de comunicación
- Seguridad en bases de datos
- Seguridad en aplicaciones
- Criptografía
- Análisis de vulnerabilidades
- Análisis de riesgos
- Principios de codificación segura
- Normas, estándares, procedimientos y buenas prácticas de seguridad informática
- Auditoría y respuesta a incidentes de seguridad

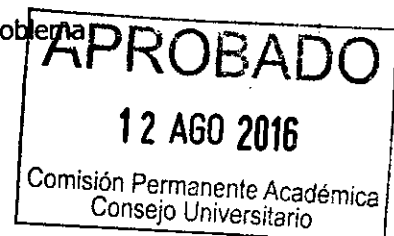
7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de casos
- Desarrollo de proyectos
- Reportes de prácticas
- Resolución de situaciones problema
- Prácticas de laboratorio
- Pruebas de desempeño

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 80%

- Resolución de casos
- Desarrollo de proyectos
- Reportes de prácticas
- Resolución de situaciones problema
- Prácticas de laboratorio



[Redacted]

- Pruebas de desempeño

Evaluación de producto –
20%

- Desarrollo de proyectos
- Pruebas de desempeño

9. REFERENCIAS

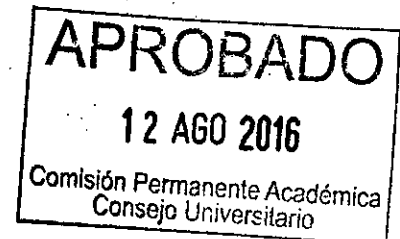
1. Easttom, W. (2016). *Computer Security Fundamentals* (3rd. Ed). United States of America: Pearson IT Certification.
2. Forouzan, B.A. (2010). *TCP/IP Protocol Suite*. New York, NY, USA: McGraw-Hill.
3. Hernandez, S. & Schou, C. (2014). *Information Assurance Handbook: Effective Computer Security and Risk Management Strategies*. United States of America: McGraw-Hill Osborne Media.
4. Kizza, J.M. (2013). *Guide to Computer Network Security*. England: Springer.
5. Konheim, A.G. (2007). *Computer Security and Cryptography*. United States of America: Wiley-Interscience.
6. Kurtz, G.J., & Scambray, J. (2012). *Hacking Exposed 7: Network Security Secrets & Solutions*. United States of America: McGraw-Hill.
7. Liu, V. (2012). *Web Application Security, A Beginner's Guide*. United States of America: McGraw-Hill.
8. Messerschmidt, T. (2016). *Identity and Data Security for Web Development*. United States of America: O'Reilly Media, Inc.
9. Nader, F. M. (2014). *Computer and Communication Networks* (2nd ed.). United States of America: Prentice Hall.
10. Oriyano, S. (2013). *Hacker Techniques, Tools, and Incident Handling*. (2nd ed.). United States of America: Jones & Bartlett Learning.
11. Sima, C., & Liu, V. (2010). *Hacking Exposed Web Applications*. (3rd ed.). United States of America: McGraw-Hill.
12. Stallings, W. (2014). *Data and Computer Communications*. New Jersey, United States of America: Pearson Prentice Hall.
13. Tanenbaum, A.S., & Wetherall, D. J. (2010). *Computer Networks* (5th ed.). United States of America: Prentice Hall.
14. Vacca, J.R. (2012). *Computer and Information Security Handbook*. (2nd ed.). United States of America: Morgan Kaufmann.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

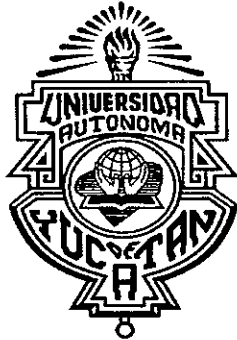


10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado en el área.
- Mínimo tres años de experiencia profesional de trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Requisitos de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Requisitos de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Sexto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Requisitos de Software permitirá a los estudiantes de Ingeniería de Software definir los requisitos de un producto de software de manera sistemática, disciplinada y cuantificable.

El propósito de esta asignatura es aportar los métodos, técnicas y herramientas para la planeación, educación, análisis, especificación, verificación, validación, aceptación y administración de los requisitos de software.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Requisitos de Software se relaciona con las asignaturas Diseño de Software, Aseguramiento de la Calidad del Software e Interacción Humano Computadora; ya que contribuyen al desarrollo de las competencias de egreso:

- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

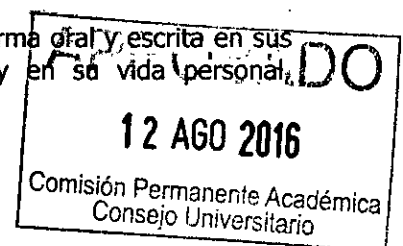
4. COMPETENCIA DE LA ASIGNATURA

Aplica los procesos de desarrollo y administración de requisitos de software en un problema del mundo real, utilizando los métodos, técnicas y herramientas de Ingeniería de Software.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal.





utilizando correctamente el idioma.

- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.



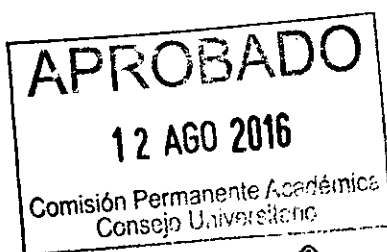
Disciplinares

- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



Específicas

- Aplica técnicas de educación de requisitos de software, de acuerdo a las características del contexto de un problema real.
- Modela el proceso de negocio de la organización, considerando su dominio de aplicación.
- Analiza los requisitos de software de un problema real, determinando la pertinencia de su implementación
- Desarrolla documentos de Especificación de Requisitos de Software, empleando notaciones y/o estándares de Ingeniería de Software.
- Aplica métodos y técnicas de Ingeniería de Software, para el aseguramiento de la calidad de los requisitos de software.
- Aplica métodos y técnicas de Ingeniería de Software, para el control de los cambios en los requisitos de software.
- Utiliza herramientas de Ingeniería de Software, para el desarrollo y administración del proceso de requisitos de software.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al desarrollo de requisitos de software
- El Proceso de Desarrollo de Requisitos
- Educación de requisitos de software
- Análisis y negociación de requisitos de software
- Especificación de requisitos de software
- Verificación y validación de requisitos de software
- Gestión de requisitos de software

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Aprendizaje basado en problemas
- Juego de roles
- Aprendizaje cooperativo

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso – 70%

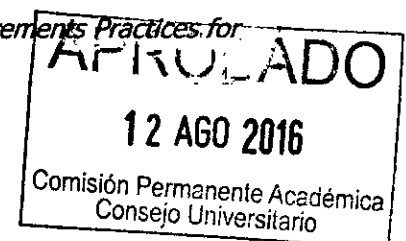
- Debates
- Resolución de casos
- Resolución de situaciones problema
- Elaboración de reportes

Evaluación de producto – 30%

- Desarrollo de proyectos

9. REFERENCIAS

1. Beatty, J., & Chen, A. (2012). *Visual Models for Software Requirements*. USA: Microsoft Press.
2. Leffingwell, D. (2010). *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. USA: Pearson Education.



3. Robertson, S. & Robertson, J. (2013). *Mastering the Requirements Process: Get the requirements right* (3ª ed.). USA: Pearson Education.
4. Wiegers, K. (2006). *More about Software Requirements: Thorny Issues and Practical Advice*. USA: Microsoft Press.
5. Wiegers, K. & Beatty, J. (2013). *Software Requirements* (3ª ed.). USA: Microsoft Press.
6. Withall, S. (2007). *Software Requirement Patterns*. USA: Microsoft Press.
7. Yayici, E. (2013). *Business Analyst's Mentor Book: With Best Practice Business Analysis Techniques and Software Requirements Management Tips*. BA- Works.
8. Yayici, E. (2015). *Business Analysis: Methodology Book*. BA- Works.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional en investigación y/o trabajo en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Responsabilidad Social Universitaria

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Responsabilidad Social Universitaria		
b. Clasificación	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Primer semestre		
e. Duración total en horas	96	Horas presenciales 48	Horas no presenciales 48
f. Créditos	6		



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos
previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Al término del curso, el estudiante podrá explicar y practicar la responsabilidad social universitaria (RSU), en forma individual y colaborativa, siendo capaz de interrogar críticamente su propia educación y la manera como se construye la formación profesional y humanística en su universidad, a la luz de los desafíos económicos, sociales y medioambientales globales, a fin de querer ser una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad, desde su vida profesional, ciudadana y personal.

3. RELACIÓN CON OTROS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

La asignatura de Responsabilidad Social Universitaria, al ser una asignatura institucional obligatoria tiene una relación transversal con las competencias de egreso de los programas educativos de la universidad a nivel licenciatura y posgrado.

4. COMPETENCIA DE LA ASIGNATURA

Practica la Responsabilidad Social Universitaria, en forma individual y colaborativa, como interrogación crítica de los impactos de la formación universitaria humanística y profesional mediante el uso de herramientas de investigación de RSU en la misma universidad, y evaluada a la luz del contexto sistémico económico, social y medioambiental global, a fin de querer ser una persona prosocial y creativa, agente de cambio para un desarrollo más justo y sostenible de su sociedad.

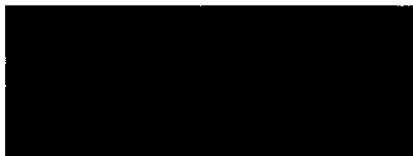
5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Formula, gestiona y evalúa proyectos en su ejercicio profesional y personal, considerando los criterios del desarrollo sostenible.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.

Comisión Permanente Académica
Consejo Universitario



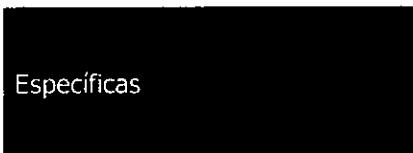


- Promueve el desarrollo sostenible en la sociedad con su participación activa.
- Valora la diversidad y multiculturalidad en su quehacer cotidiano, bajo los criterios de la ética.



Disciplinares

- Explica los desafíos globales y locales del desarrollo social justo y sostenible a la luz de informaciones actualizadas y científicamente sustentadas.
- Reconoce, describe y explica la relación entre los problemas sociales y ambientales localmente aparentes y las estructuras globales subyacentes que los provocan, en forma científicamente sustentada.
- Identifica los impactos sociales y medioambientales de sus acciones personales, profesionales y ciudadanas, de manera proactiva y responsable.
- Identifica y argumenta frente a sus colegas los impactos negativos (riesgos sociales y ambientales) y limitaciones actuales de su profesión, en forma creativa y prospectiva para la mejora continua técnica y deontológica de su profesión.
- Organiza actividades colectivas prosociales a la luz de los problemas económicos, sociales y medioambientales que diagnostica en su entorno, en forma argumentada, democrática y responsable.
- Busca y utiliza las soluciones técnicas, gerenciales y metodológicas que le permitan evitar los impactos sociales y ambientales negativos en su quehacer profesional.
- Incorpora las exigencias de la responsabilidad social y las metas del desarrollo social justo y sostenible en su actividad profesional y personal, en forma coherente y creativa.
- Valora la congruencia entre el hacer y el decir, la transparencia en el quehacer profesional y la participación democrática de todas las partes interesadas en dicho quehacer, en todas las organizaciones en la que participa y trabaja.
- Incorpora el hecho de reflexionar, antes de actuar, en los impactos y riesgos sociales y ambientales que puedan surgir de su actividad profesional, en cualquier situación laboral.



Específicas

- Identifica y explica los desafíos globales (sociales y ambientales) del desarrollo mundial actual, a la luz de los impactos negativos de las rutinas sistémicas económicas y sociales.



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





- Reconoce las contradicciones de la educación universitaria y profesional actual a la luz de los desafíos globales (sociales y ambientales) del desarrollo mundial actual.
- Argumenta y diseña, en forma colaborativa, soluciones posibles a los desafíos globales (sociales y ambientales) del desarrollo mundial actual.
- Aplica y evalúa herramientas de investigación-diagnóstico RSU en su comunidad universitaria, en forma colaborativa.
- Toma conciencia de su responsabilidad compartida en cuanto a los problemas sociales y ambientales que diagnostica, así como de su potencial personal para participar en su solución.
- Valora y promueve la RSU en su Alma Mater, en forma personal y colaborativa.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- El carácter insostenible (social y ambiental) de nuestro desarrollo actual.
- Desarrollo justo y sostenible.
- Ética en 3D, mirada crítica hacia la educación.
- ISO 26000, Pacto Global.
- Herramientas diagnóstico RSU del Manual de primeros pasos en RSU.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje informativo
- Aprendizaje colaborativo
- Investigación con supervisión
- Argumentación de ideas
- Uso de debates
- Aprendizaje autónomo y reflexivo

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Reporte de revisión de fuentes de información
- Ensayos escritos
- Redacción informes
- Participación en foros virtuales

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Evaluación de producto – 40%

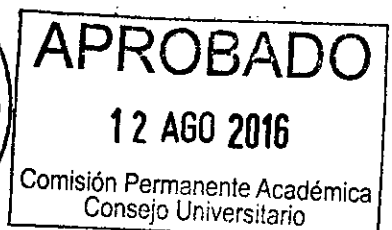
- Presentación del informe final de los resultados del diagnóstico RSU

9. REFERENCIAS

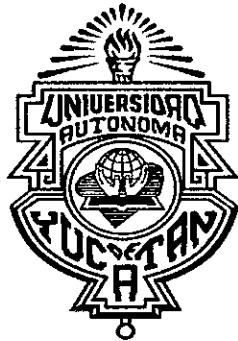
1. ONU (2000). *Declaración del milenio*. Resolución de las Naciones Unidas.
2. La Carta de la Tierra (2000). Recuperado de: <http://www.earthcharterinaction.org/contenido/pages/La-Carta-de-la-Tierra.html>
3. ONU (1999). Pacto Global. Recuperado de: <http://www.un.org/es/globalcompact/>
4. ISO (2010). *Norma Internacional ISO 26000. Guía de responsabilidad social*. Ginebra: ISO
5. WWF (2012). *Living Planet Report*. WWF International, Gland.
6. Vallaeys, et al. (2009). *Manual de primeros pasos en RS*. México: McGraw Hill.

10. PERFIL DESEABLE DEL PROFESOR

- Formación específica en RSU
- Competencias en el manejo de la enseñanza virtual (técnica y pedagógicamente)
- Conocimiento de la temática del desarrollo social sostenible
- Valore y quiera promover la RSU en la UADY, participando más allá del curso en un comité de autodiagnóstico y mejora continua de la RSU en la UADY.



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Servicio Social

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Presencial

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Servicio Social		
b. Tipo	Obligatoria		
c. Modalidad	Presencial		
d. Ubicación sugerida	Séptimo semestre		
e. Duración total en horas	480	Horas presenciales 480	Horas no presenciales 0
f. Créditos	12		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Para la asignación de algún proyecto de servicio social, el estudiante deberá haber obtenido al menos 252 créditos del Plan de Estudios de la Licenciatura en Ingeniería de Software.

2. JUSTIFICACIÓN DEL SERVICIO SOCIAL EN EL PE

El Servicio Social es importante para los estudiantes de la Licenciatura en Ingeniería de Software porque permite, mediante programas de responsabilidad compartida, articular esfuerzos de las instituciones educativas y estudiantes con el sector gubernamental, productivo y social.

El Servicio Social integra la formación académica del estudiante, desarrolla valores, favorece la inserción al mercado laboral y coadyuva al desarrollo del país.

3. COMPETENCIAS DE EGRESO QUE SE FAVORECERÁN POR MEDIO DEL SERVICIO SOCIAL

El Servicio Social contribuye al logro de las cuatro competencias de egreso.

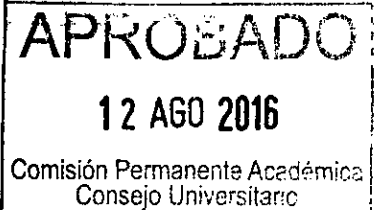
- Desarrolla productos de software de calidad de pequeña a gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificable.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.
- Administra los procesos de desarrollo, mantenimiento, calidad y configuración del software, mediante un enfoque sistemático, disciplinado y cuantificable, optimizando los recursos humanos, materiales, económicos y de tiempo, con apego a la ética profesional.
- Elabora propuestas de mejora en el desarrollo, mantenimiento y administración de los procesos de software, mediante fundamentos matemáticos, ingenieriles, de las ciencias computacionales y los propios de la Ingeniería de Software.

4. ESTRATEGIAS DE ACOMPAÑAMIENTO PARA LA MOVILIZACIÓN Y EL DESARROLLO DE COMPETENCIAS

- Seguimiento interno por parte de profesores asignados al programa.

5. ESTRATEGIAS GENERALES DE EVALUACION

- Evaluación por la institución receptora
- Elaboración del reporte final



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Sistemas Distribuidos

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Sistemas Distribuidos		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Octavo semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Los modelos de Sistemas Distribuidos han sido ampliamente utilizados en el desarrollo de las aplicaciones integradas que se ejecutan en ambientes computacionales que interconectan diferentes componentes de hardware y software, permitiendo la compartición de recursos y servicios.

Esta asignatura proporciona al estudiante los modelos y técnicas de procesamiento e intercambio de información que se utilizan en sistemas computacionales interconectados para implementar aplicaciones distribuidas eficientes acordes a los requerimientos informáticos de las organizaciones.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Sistemas Distribuidos se relaciona con las asignaturas de Sistemas Operativos, Arquitectura y Organización de Computadoras, Estructuras de Datos, Programación Estructurada, Programación Orientada a Objetos y Redes y Seguridad de Computadoras; ya que contribuyen al logro de las competencias de egreso: Desarrollo de Software y Mantenimiento de Software.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla aplicaciones eficientes en ambientes de procesamiento compartido, acordes con los modelos y arquitecturas de sistemas distribuidos.

5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Establece relaciones interpersonales, en los ámbitos en los que se desenvuelve, de manera positiva y respetuosa.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Disciplinares

- Trabaja bajo presión de manera eficaz y eficientemente.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Identifica claramente los modelos lógicos y de arquitecturas de sistemas distribuidos utilizados en aplicaciones informáticas.
- Construye aplicaciones eficientes para el intercambio de datos en procesos distribuidos.
- Implementa aplicaciones de software eficientes que integran un conjunto de servicios web.
- Diseña aplicaciones de software concurrentes que comparten recursos computacionales de manera consistente, utilizando algoritmos de coordinación y acuerdo.
- Aplica las estrategias de seguridad en sistemas distribuidos para el mantenimiento de la integridad, disponibilidad y confiabilidad de la información.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Modelos de Sistemas Distribuidos
- Comunicación entre procesos
- Invocación remota de procesos
- Servicios Web
- Manejo del tiempo en sistemas distribuidos
- Algoritmos de coordinación y acuerdo
- Seguridad en comunicaciones y sistemas distribuidos

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Estudio de casos
- Aprendizaje orientado a proyectos
- Resolución de problemas
- Aprendizaje cooperativo



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

- Prácticas de programación

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Resolución de casos
- Reportes de prácticas de programación
- Pruebas de desempeño

Evaluación de producto – 30%

- Desarrollo de proyecto de programación

9. REFERENCIAS

1. Anthony, R. J. (2015). *Systems Programming: Designing and Developing Distributed Applications*. USA: Morgan Kaufmann.
2. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed Systems: Concepts and Design* (5th ed.). USA: Pearson.
3. Ghosh, S. (2014). *Distributed Systems: An Algorithmic Approach*. (2nd ed.). USA: Chapman & Hall/CRC Press.
4. Kshemkalyani, A.D., & Singhal, M. (2011). *Distributed Computing: Principles, Algorithms and Systems*. Inglaterra: Cambridge University Press.
5. Tanenbaum, A.S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms* (2nd ed.). USA: Pearson.
6. Varela, C. (2013). *Programming Distributed Computing Systems: A Foundational Approach*. USA: MIT Press.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional o en investigación en el área de Redes y Sistemas Distribuidos.
- Mínimo un año de experiencia docente en el nivel superior.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



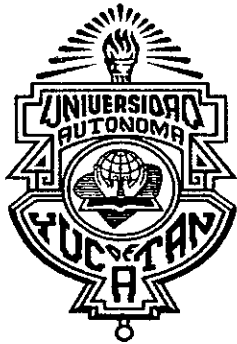
APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE



Sistemas Operativos

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Sistemas Operativos		
b. Tipo	Obligatorio		
c. Modalidad	Mixta		
d. Ubicación sugerida	Cuarto semestre		
e. Duración total en horas	112	Horas presenciales 72	Horas no presenciales 40
f. Créditos	7		
g. Requisitos académicos	Ninguno		

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Los Sistemas Operativos es esencial en la formación de un profesionista en el área de cómputo, pues le permitirá analizar la organización, estructura y algoritmos que utilizan los componentes de los administradores de recursos de un equipo de cómputo; y los principios para el diseño de un sistema operativo y sus relaciones con el hardware, desarrollo de aplicaciones, programas de aplicación, etc.

Esta asignatura proporciona los fundamentos para analizar el funcionamiento de los diferentes componentes de los sistemas operativos.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Los Sistemas Operativos se relaciona con las asignaturas Arquitectura y Organización de Computadoras, Matemáticas Discretas, Redes y Seguridad de Computadoras, Estructuras de Datos y Diseño de Bases de Datos; ya que contribuyen al logro de las competencias de egreso:

- Desarrolla productos de software de calidad en pequeña y gran escala aplicando técnicas, herramientas, métodos y procedimientos, a través de un enfoque sistemático, disciplinado y cuantificado.
- Mantiene productos de software heredados en diferentes dominios de aplicación, optimizando los recursos humanos, materiales, económicos y de tiempo, y atendiendo las necesidades de la organización.

4. COMPETENCIA DE LA ASIGNATURA

Analiza la operación de los diferentes componentes de los administradores de recursos de un sistema operativo, considerando el marco teórico de las ciencias computacionales.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



5. COMPETENCIAS GENERICAS, DISCIPLINARES Y ESPECIFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

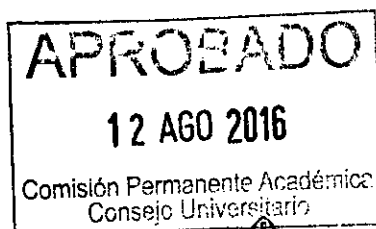
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal utilizando correctamente el idioma.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Desarrolla su pensamiento, en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Manifiesta comportamientos profesionales y personales, en los ámbitos en los que se desenvuelve, de manera transparente y ética.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.

Disciplinares

- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Describe el objetivo, la función y las características de diferentes componentes de los sistemas operativos modernos, de manera clara y ordenada.
- Describe con claridad los diferentes tipos de Sistemas Operativos utilizados actualmente.
- Describe las características de las técnicas para la administración del procesador, de la memoria, de los dispositivos de E/S, y de información, utilizadas en un sistema operativo, de manera clara y ordenada.
- Analiza las técnicas y algoritmos para la administración de procesos y del procesador, memoria, dispositivos de E/S e información, utilizados en los sistemas operativos.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Conceptos fundamentales de Sistemas Operativos
- Administración del procesador: procesos, concurrencia, comunicación
- Administración de la memoria: técnicas de asignación de memoria
- Administración de los dispositivos de E/S
- Administración de la información: Sistema de archivos, asignación de espacio

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Proyectos de Investigación
- Resolución de problemas y ejercicios
- Prácticas supervisadas
- Aprendizaje basado en problemas

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Prácticas supervisadas
- Resolución de ejercicios y problemas

Evaluación de producto – 30%

- Elaboración de proyectos
- Investigaciones documentales

9. REFERENCIAS

1. Deitel, HH., Deitel, P., Choffnes, R. (2004). *Operating Systems* (3rd. ed.). New Jersey, USA: Pearson Prentice Hall.
2. Silberschatz, A., Galvin, P., & Gagne, G. (2012). *Operating Systems*. (9th ed.) New Jersey, USA: Wiley & Sons.
3. Silberschatz, A., Galvin, P. B. & Gagne, G. (2013). *Operating system concepts: International student version*. Hoboken, New Jersey: Wiley.
4. Stallings, W. (2014). *Internals and Design Principles*. (8th. ed). Pearson

APROBADO

12 AGO 2016

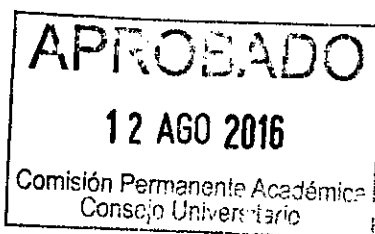
Comisión Permanente Académica
Consejo Universitario



5. Tannenbaun, A., Boss, H. (2014) *Modern Operating Systems*. (4th. ed.). USA: Pearson.
6. Tannenbaun, A., Woodhill, A. (2006). *Operating Systems Design and Implementations*. (3rd. ed.) USA: Pearson Prentice Hall.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado en el área de Tecnologías de Información.
- Mínimo dos años de experiencia profesional de trabajo en el área de programación.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

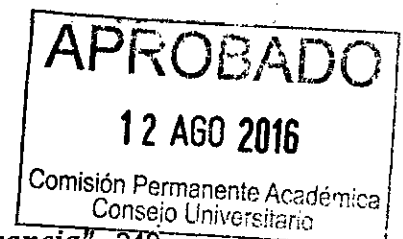


Taller de Emprendedores

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Taller de Emprendedores		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Sexto semestre		
e. Duración total en horas	96	Horas presenciales 48	Horas no presenciales 48
f. Créditos	6		



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Taller de Emprendedores es una asignatura integradora, que permitirá al estudiante demostrar las competencias desarrolladas durante la licenciatura, al diseñar proyectos tecnológicos innovadores y con potencial de éxito.

Esta asignatura aporta las herramientas necesarias para planear, elaborar y ejecutar un proyecto tecnológico, considerando los aspectos éticos, financieros y legales.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Taller de Emprendedores se relaciona con todas las asignaturas del Plan de Estudios de la Licenciatura en Ingeniería de Software; por lo que contribuye al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Diseña un proyecto tecnológico orientado a la resolución de un problema del entorno, con responsabilidad, innovación, iniciativa y sentido de la ética.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal utilizando correctamente el idioma.
- Gestiona el conocimiento, en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.
- Interviene con iniciativa y espíritu emprendedor en su

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





ejercicio profesional y personal de forma autónoma y permanente.

- Promueve el desarrollo sostenible en la sociedad con su participación activa.



Disciplinares

- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



Específicas

- Identifica los atributos de un profesional emprendedor, de acuerdo con los marcos de referencia propios de la disciplina.
- Describe, de manera clara y ordenada, las etapas del desarrollo de un proyecto emprendedor.
- Identifica aspectos legales y de finanzas necesarios en el desarrollo de proyectos innovadores, considerando las necesidades actuales y futuras del mismo.
- Determina los recursos necesarios para el desarrollo de un proyecto tecnológico e innovador que resuelve un problema del entorno.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al emprendimiento.
- Generación y validación de la idea.
- Modelos del negocio.
- Plan de ejecución y operación.
- Aspectos éticos, financieros y legales



7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Desarrollo de proyectos.
- Aprendizaje colaborativo.
- Estudios de casos.
- Aprendizaje basado en situaciones problema.

8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 60%

- Resolución de problemas.
- Pruebas de desempeño.
- Reporte de estudio de caso.
- Uso de organizadores gráficos.

Evaluación de producto – 40%

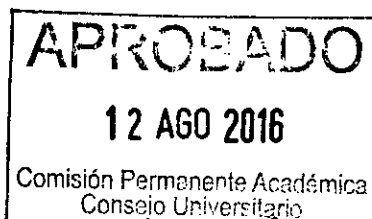
- Desarrollo de proyecto de emprendimiento.

9. REFERENCIAS

1. Catmull, E. E. & Wallace, A. (2014). *Creativity, Inc.: Overcoming the unseen forces that stand in the way of true inspiration*. Random House.
2. Cornwall, J. R., Vang, D. O. & Hartman, J. M. (2013). *Entrepreneurial financial management: An applied approach*. Armonk, NY: M.E. Sharpe.
3. Osterwalder, A., Pigneur, Y. & Clark, T. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*. Hoboken, NJ: Wiley.
4. Osterwalder, A., Pigneur, Y., Bernarda, G. & Smith, A. (2014). *Value proposition design: How to create products and services customers want* (Strategyzer). Wiley.
5. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses* (2012). S.I.: Nikken BP Sha.

10. PERFIL DESEABLE DEL PROFESOR

- Profesional con Maestría en el área de las Tecnologías de la Información.
- Mínimo dos años de experiencia profesional.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

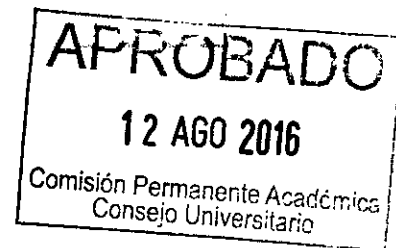


Teoría de la Computación

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACION

a. Nombre de la asignatura	Teoría de la Computación		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Tercer semestre		
e. Duración total en horas	128	Horas presenciales 72	Horas no presenciales 56
f. Créditos	8		



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

La Teoría de la Computación es el fundamento para el entendimiento de los lenguajes formales, ya que permite diseñar compiladores para hardware específico.

Esta asignatura proporciona al estudiante los fundamentos matemáticos y algorítmicos, para identificar las capacidades y limitaciones de las computadoras.

3. RELACION CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Teoría de la Computación se relaciona con las asignaturas Programación Estructurada, Matemáticas Discretas, Programación Orientada a Objetos y Teoría de Lenguajes de Programación; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Diseña algoritmos computacionales que reconocen los elementos básicos de un lenguaje de programación, con base en los modelos formales de la teoría computacional.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.
- Utiliza habilidades de investigación, en sus intervenciones profesionales con rigor científico.
- Desarrolla su pensamiento en intervenciones profesionales y personales, de manera crítica, reflexiva y creativa.
- Trabaja con otros en ambientes multi, inter y transdisciplinarios de manera cooperativa.
- Responde a nuevas situaciones en su práctica profesional y en su vida personal, en contextos locales, nacionales e internacionales, con flexibilidad.
- Trabaja bajo presión de manera eficaz y eficientemente.

AP

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



Disciplinares

- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos y del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.

Específicas

- Resuelve correctamente problemas computacionales, utilizando lenguajes y expresiones regulares.
- Diseña algoritmos útiles en verificación de patrones, utilizando autómatas finitos.
- Diseña lenguajes de programación básicos, mediante gramáticas libres de contexto.
- Diseña autómatas de pila que reconocen correctamente lenguajes libres de contexto.
- Diseña algoritmos eficientes para el reconocimiento de lenguajes, usando máquinas de Turing.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Lenguajes y Expresiones Regulares.
- Autómatas Finitos.
- Gramáticas y Lenguajes Libres de Contexto.
- Autómatas de Pila.
- Máquinas de Turing.

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Resolución de ejercicios y problemas
- Aprendizaje basado en problemas
- Prácticas de programación
- Seminario
- Aprendizaje colaborativo



8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 80%

- Pruebas de desempeño
- Informes de prácticas
- Seminario

Evaluación de producto – 20%

- Proyecto de programación

9. REFERENCIAS

1. Kelley, D. (1995). *Teoría de autómatas y lenguajes formales*. Madrid: Prentice Hall. (Clásico)
2. Brookshear, J. G. (1993). *Teoría de la Computación, Lenguajes Formales, Autómatas y Complejidad*. Addison Wesley, Iberoamericana. (Clásico)
3. Brena, R. (2003). *Lenguajes Formales y Autómatas*. Centro de Inteligencia Artificial, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey. (Clásico)
4. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). *Introduction to Automata Theory, Languages, and Computation*. (3rd ed.). Addison Wesley. (Clásico)
5. Sipser, M. (2012). *Introduction to the Theory of Computation*. (3rd ed.). Course Technology.
6. Goddard, W. (2008). *Introducing the Theory of Computation*. USA: Jones & Bartlett Publishers.
7. Kozen, D. C. (2010). *Theory of Computation (Texts in Computer Science)*, Springer.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software, Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado.
- Mínimo dos años de experiencia profesional.
- Mínimo dos años de experiencia docente en el área.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



LICENCIATURA EN INGENIERÍA DE SOFTWARE

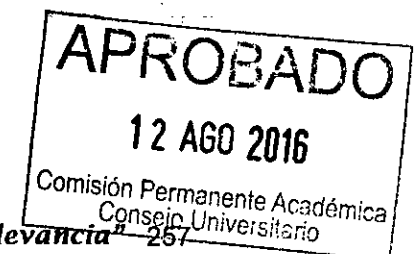


Teoría de Lenguajes de Programación

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Teoría de Lenguajes de Programación		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Cuarto semestre		
e. Duración total en horas	96	Horas presenciales 64	Horas no presenciales 32
f. Créditos	6		



g. Requisitos académicos previos

Ninguno

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

El estudio de Teoría de Lenguajes de Programación es importante para la formación de los estudiantes de Ingeniería de Software, ya que les permitirá seleccionar adecuadamente el lenguaje de programación para implementar la solución de un problema.

El propósito de esta asignatura es aportar los fundamentos, conceptos básicos y aspectos de diseño e implementación de los principales lenguajes de programación.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Teoría de Lenguajes de Programación se relaciona con las asignaturas Algoritmia, Fundamentos de Ingeniería de Software, Programación Estructurada, Programación Orientada a Objetos, Teoría de la Computación, Arquitectura y Organización de Computadoras, Diseño de Software, Estructuras de Datos, Sistemas Operativos, Arquitecturas de Software, Construcción de Software, Diseño de Bases de Datos, Desarrollo de Aplicaciones Web, Métricas de Software, Aseguramiento de la Calidad del Software, Requisitos de Software, Interacción Humano Computadora, Experimentación en Ingeniería de Software, Verificación y Validación de Software, Redes y Seguridad de Computadoras, Innovación Tecnológica, Administración de Proyectos I, Mantenimiento de Software, Sistemas Distribuidos y Administración de Proyectos II; ya que contribuyen al logro de las cuatro competencias de egreso: Desarrollo de Software, Mantenimiento de Software, Administración de los Procesos de Software e Innovación en Ingeniería de Software.

4. COMPETENCIA DE LA ASIGNATURA

Desarrolla software utilizando diversos paradigmas de programación, con base en la eficiencia y pertinencia del lenguaje.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

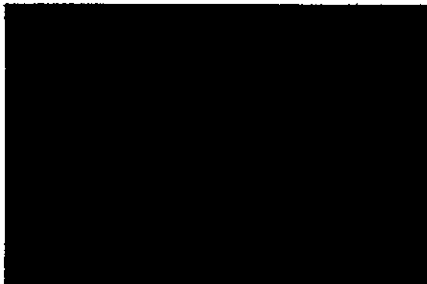
- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.

APR

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario





Disciplinares

Específicas

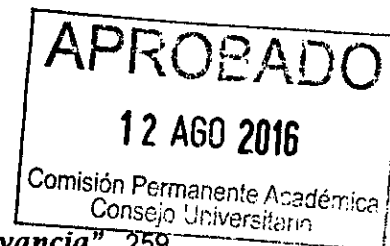
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Actualiza sus conocimientos y habilidades para su ejercicio profesional y su vida personal, de forma autónoma y permanente.
- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Trabaja bajo presión de manera eficaz y eficientemente.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Identifica los conceptos básicos de los lenguajes de programación, con argumentos congruentes y lógicos.
- Analiza los conceptos sintácticos y semánticos de los identificadores, variables y tipos de datos de un lenguaje de programación de manera fundamentada.
- Analiza las expresiones, estructuras de control y subprogramas utilizados en los lenguajes de programación, de manera fundamentada.
- Desarrolla aplicaciones representativas usando los paradigmas de programación, de acuerdo a sus marcos de referencia.

6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Conceptos básicos de lenguajes de programación.
- Principios básicos del diseño de los lenguajes.
- Expresiones, estructuras de control y subprogramas.
- Paradigmas de programación

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje autónomo y reflexivo
- Resolución de problemas y ejercicios
- Aprendizaje mediado por las TIC
- Prácticas en laboratorio
- Uso de organizadores gráficos



8. ESTRATEGIAS GENERALES DE EVALUACIÓN

Evaluación de proceso – 70%

- Pruebas de desempeño
- Resolución de situaciones problema
- Investigación documental

Evaluación de producto – 30%

- Desarrollo de proyectos

9. REFERENCIAS

1. Gabbrielli, M. & Martini, S. (2010). *Programming languages: principles and paradigms*. London: Springer.
2. López, F. y García, A. (2014). *Teoría de los Lenguajes de Programación*. España: Editorial Universitaria Ramón Areces.
3. Louden, K. & Lambert, K. (2012). *Programming languages: principles and practice*. Boston: Course Technology/Cengage Learning.
4. Louden, K. (2004). *Lenguajes de Programación. Principios y Práctica*. México: Thomson.

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado.
- Mínimo un año de experiencia profesional en el área.
- Mínimo dos años de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



LICENCIATURA EN INGENIERÍA DE SOFTWARE

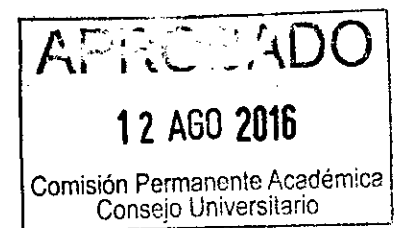


Verificación y Validación de Software

Tipo de asignatura
Obligatoria
Modalidad de la asignatura
Mixta

1. DATOS GENERALES DE IDENTIFICACIÓN

a. Nombre de la asignatura	Verificación y Validación de Software		
b. Tipo	Obligatoria		
c. Modalidad	Mixta		
d. Ubicación sugerida	Séptimo semestre		
e. Duración total en horas	96	Horas presenciales 64	Horas no presenciales 32
f. Créditos	6		
g. Requisitos académicos	Ninguna		



previos

2. INTENCIONALIDAD FORMATIVA DE LA ASIGNATURA

Verificación y Validación de Software es importante para la formación del Ingeniero de Software porque aborda conceptos fundamentales de revisiones, inspecciones, pruebas de software y tolerancia a fallas de productos de software.

Esta asignatura proporciona los métodos y técnicas para el diseño de un plan para verificación y validación basado en un estándar propuesto por un organismo reconocido.

3. RELACIÓN CON OTRAS ASIGNATURAS EN ALINEACIÓN CON LAS COMPETENCIAS DE EGRESO

Verificación y Validación de Software se relaciona con las asignaturas Construcción de Software, Aseguramiento de la Calidad, Administración de Proyectos I, Administración de Proyectos II y Mantenimiento de Software; ya que en conjunto contribuyen al desarrollo de las competencias de egreso: Desarrollo de Software, Mantenimiento de Software y Administración de Procesos de Software.

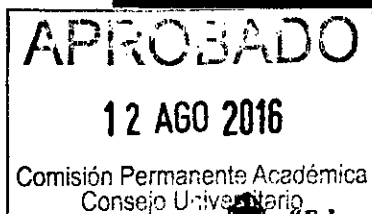
4. COMPETENCIA DE LA ASIGNATURA

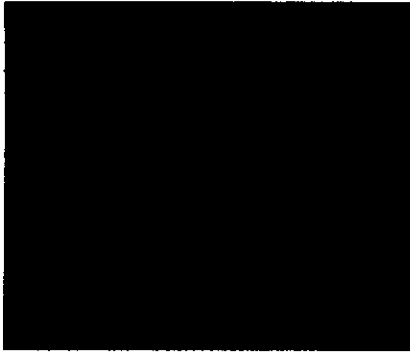
Implementa métodos y técnicas de verificación y validación, contribuyendo al aseguramiento la calidad de un producto de software construido.

5. COMPETENCIAS GENÉRICAS, DISCIPLINARES Y ESPECÍFICAS A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas

- Se comunica en español en forma oral y escrita en sus intervenciones profesionales y en su vida personal, utilizando correctamente el idioma.
- Usa las TIC en sus intervenciones profesionales y en su vida personal de manera pertinente y responsable.

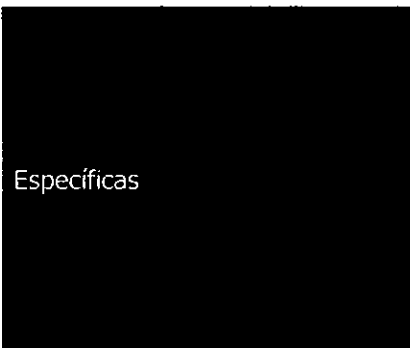




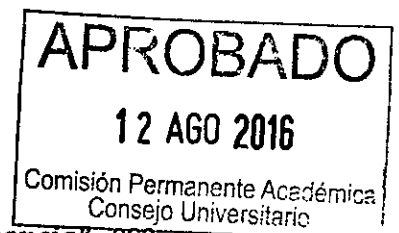
- Gestiona el conocimiento en sus intervenciones profesionales y en su vida personal, de manera pertinente.
- Aplica los conocimientos en sus intervenciones profesionales y en su vida personal con pertinencia.
- Toma decisiones en su práctica profesional y personal, de manera responsable.
- Pone de manifiesto su compromiso con la calidad y la mejora continua en su práctica profesional y en su vida personal de manera responsable.



- Interpreta correctamente tablas, gráficas, diagramas y textos expresados con lenguaje matemático y científico que se utilizan en las matemáticas de nivel superior.
- Construye modelos matemáticos e ingenieriles mediante la aplicación de principios y procedimientos algebraicos, geométricos, del cálculo, para la comprensión y análisis de situaciones reales e hipotéticas.
- Diseña algoritmos computacionales eficientes aplicando conceptos básicos de matemáticas discretas, lógica, algoritmia y estructuras de datos.
- Resuelve problemas computacionales aplicando el conocimiento de la estructura, organización, funcionamiento, programación e interconexión de sistemas de cómputo.



- Aplica revisiones de software, utilizando las técnicas y métodos adecuados.
- Implementa pruebas de software, utilizando las técnicas y métodos adecuados.
- Identifica las actividades del análisis de seguridad y confiabilidad, para la creación de aplicaciones de software óptimas.
- Diseña un plan de verificación y validación asegurando la calidad del software y empleando los estándares pertinentes.



6. CONTENIDOS ESENCIALES PARA EL DESARROLLO DE LA COMPETENCIA DE LA ASIGNATURA

- Introducción al proceso de verificación y validación
- Revisiones de Software: Inspecciones, Recorridos y Revisiones Personales
- Pruebas de Software
- Análisis de Seguridad y Confiabilidad
- Tolerancia a Fallas
- Planeación de la Verificación y Validación del Software

7. ESTRATEGIAS DE ENSEÑANZA Y APRENDIZAJE

- Aprendizaje cooperativo
- Resolución de problemas y ejercicios
- Investigación documental
- Proyectos de investigación
- Aprendizaje orientado a proyecto
- Desarrollo de programas de cómputo
- Investigación bibliográfica

8. ESTRATEGIAS GENERALES DE EVALUACION

Evaluación de proceso – 70%

- Resolución de casos
- Investigación documental
- Desarrollo de proyectos

Evaluación de producto – 30%

- Portafolio de evidencias
- Proyecto Integrador

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



9. REFERENCIAS

1. IEEE (2012). *IEEE Std. 1012-2012 Standard for System and Software Verification and Validation*.
2. Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston, United States: Addison-Wesley.
3. Pressman, R. (2000). *Ingeniería de Software – Un enfoque práctico*. Madrid, España: Mc Graw Hill (Clásico)

10. PERFIL DESEABLE DEL PROFESOR

- Ingeniero de Software o carrera afín, preferentemente con posgrado en el área de Tecnologías de Información.
- Mínimo dos años de experiencia profesional de trabajo en el área de programación.
- Mínimo un año de experiencia docente.
- Es necesario que el profesor posea todas las competencias que se declaran en la asignatura que va a impartir.



11. METODOLOGÍA DE EVALUACIÓN DEL PLAN DE ESTUDIOS

La Facultad de Matemáticas establecerá un proceso sistemático de seguimiento del programa educativo y de evaluación del plan de estudios de la Licenciatura en Ingeniería de Software, el cual permitirá retroalimentar en forma continua la operación de la Licenciatura en Ingeniería de Software. El plan de estudios deberá evaluarse integralmente una vez que egrese la primera generación o, en su caso, cada cinco años, y se prevé que participen en dicha evaluación tanto organismos externos, como Comités Internos integrados para este fin.

La evaluación interna, analizará entre otras cosas:

- Los fundamentos y contexto del Plan de Estudios
- La congruencia, vigencia, continuidad, actualidad y operatividad del Plan de Estudios
- Las actitudes, valores, ideología y principios éticos del Plan de Estudios
- Los contenidos de las asignaturas y las estrategias de enseñanza de cada una de ellas.
- La estructura de la retícula
- El rendimiento académico y factores asociados a éste
- Las tasas de retención, reprobación, rezago y eficiencia terminal
- El número de profesores que dan soporte al Plan de Estudios y los perfiles de éstos.
- La capacidad en infraestructura y equipo de apoyo para la correcta operación de las actividades académicas.
- La opinión de los docentes y alumnos sobre el funcionamiento y operatividad del programa educativo.

La evaluación externa analizará, entre otras cosas:

- Los egresados y sus funciones profesionales
- Los egresados y el mercado laboral
- Las funciones del egresado en la solución de necesidades sociales
- La tendencia de la disciplina a nivel mundial
- Las opiniones de profesionales destacados
- Las recomendaciones de los organismos acreditadores



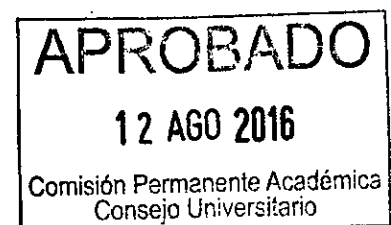
Para *retroalimentar en forma continua* la operación del plan de estudios, la Secretaría Académica y el Coordinador del programa educativo realizarán reuniones periódicas para el análisis de los indicadores de trayectoria escolar, así como situaciones relacionadas con infraestructura, procesos de apoyo y de formación integral, y con aspectos relacionados con los procesos de promoción e ingreso al programa educativo entre otros. Así mismo, los cuerpos académicos y

APROBADO**12 AGO 2016**Comisión Permanente Académica
Consejo Universitario

profesores vinculados con el programa educativo (ver sección 2.5) deberán tener una participación activa en los procesos de evaluación, planeación y actualización docente, entre otros. Adicionalmente, de manera periódica se revisarán aspectos como los siguientes:

- Las competencias de cada asignatura
- La calidad de los contenidos
- Las estrategias de enseñanza utilizada por los profesores
- Los criterios de evaluación de las asignaturas
- Las TIC y bibliografía utilizadas

En cuanto a los principios del MEFI, la evaluación permite valorar en qué medida la Licenciatura en Ingeniería de Software está orientada en la formación integral del estudiante a través de una educación centrada en el aprendizaje y basada en competencias, bajo principios y valores humanistas de la UADY, siempre tomando en cuenta conceptos fundamentales como la responsabilidad social, innovación, flexibilidad e internacionalización para el mejoramiento del programa.



12. FUNCIÓN ACADÉMICO ADMINISTRATIVA

12.1. Introducción

Los lineamientos generales para la operación del Programa Educativo de Licenciatura en Ingeniería de Software se sustentan en el MEFI, en los reglamentos que rigen la normatividad vinculada con los programas curriculares en el nivel de licenciatura de la UADY, así como en el Reglamento Interior de la Facultad de Matemáticas.

12.2. Calendario escolar

Para su operación, el programa educativo se apegará al calendario escolar aprobado por el H. Consejo Universitario; éste inicia en agosto de cada año y finaliza en julio del siguiente, y se encuentra integrado por tres periodos escolares.

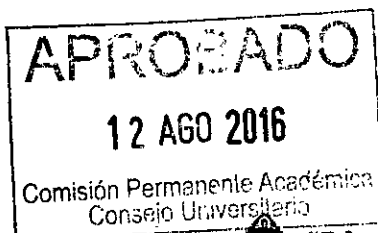
- Dos periodos lectivos de 16 semanas, o bien 80 días hábiles, con una variación de tres días más o tres días menos, denominados "semestres". Un periodo abarcará los meses de agosto a diciembre de cada año y el otro de enero a mayo del siguiente.
- Un periodo lectivo de seis semanas, o bien 30 días hábiles, con una variación de dos días más o dos días menos, denominado "periodo intensivo de verano" el cual iniciará a finales de mayo o principios de junio y concluirá a mediados de julio.

12.3. Ingreso

Para ingresar a la Licenciatura en Ingeniería de Software se requiere que el aspirante participe en el proceso de selección para el nivel licenciatura, de acuerdo a la convocatoria respectiva aprobada por el H. Consejo Universitario. La periodicidad en el ingreso al programa educativo será anual, y se realizará en agosto de cada año.

12.4. Permanencia

Para tener inscripción vigente en un período escolar, el alumno deberá inscribir cuando menos una asignatura del plan de estudios, no obstante, se recomienda inscribir un mínimo de asignaturas equivalente a 26 créditos al semestre. Dicha recomendación toma en consideración el límite máximo —catorce semestres— que se dispone para concluir el plan de estudios. Resulta importante destacar que la malla curricular propuesta representa la trayectoria escolar deseable de un alumno de tiempo completo; que debe cursar entre 37 y 43 créditos al semestre, para concluir su plan de estudios en nueve periodos semestrales.



La calificación mínima aprobatoria de una asignatura es de 70 puntos, con dicha calificación se podrá considerar que el estudiante ha alcanzado las competencias de la misma, y su nivel de dominio dependerá del puntaje obtenido (ver Tabla 27).

Tabla 27. Rangos de calificaciones.

Niveles de Dominio	
Puntaje	Categoría
90 - 100	Sobresaliente (SS)
80 - 89	Satisfactorio (SA)
70 - 79	Suficiente (S)
0 - 69	No Acreditado (NA)

Para acreditar una asignatura el estudiante tendrá cuatro oportunidades: dos cursándola de manera regular y dos con el acompañamiento de un profesor. La primera oportunidad la cursará de manera regular. Si no se acredita la asignatura, el estudiante elegirá de qué manera desea acreditarla (volviendo a cursar la asignatura o con el acompañamiento de algún profesor), así como el orden en que irán utilizando sus tres oportunidades restantes hasta agotarlas, pudiendo ser cualquiera de las siguientes combinaciones: regular-acompañamiento-acompañamiento, acompañamiento-regular-acompañamiento o acompañamiento-acompañamiento-regular. Los estudiantes que no acrediten la asignatura en estas cuatro oportunidades, serán dados de baja del Programa Educativo.

Debido a que algunas instituciones con las que la Universidad mantiene intercambio de información —p.e. instituciones que otorgan becas para estudiantes de licenciatura— aún no consideran los esquemas académico-administrativos que incorporan un sistema basado en créditos, se establece las equivalencias entre los créditos aprobados por un alumno a lo largo de su trayectoria académica, y el semestre que podría acreditar (ver Tabla 28).

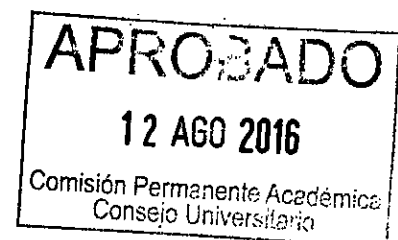


Tabla 28. Equivalencias por créditos.

Total de créditos aprobados	Semestre equivalente acreditado
1 - 35	1º
36 - 75	2º
76 - 117	3º
118 - 156	4º
157 - 195	5º
196 - 235	6º
236 - 276	7º
277 - 317	8º
318 - 360	9º

12.5. Sugerencia de asignaturas obligatorias por período lectivo

La Tabla 29 muestra una propuesta de distribución de las asignaturas obligatorias del Plan de Estudios en dos periodos semestrales; esta lista puede ser modificada o ampliada.

Tabla 29. Asignaturas obligatorias por periodo.

Período: enero-mayo	Período: agosto-diciembre
Álgebra Intermedia	Álgebra Superior
Geometría Analítica	Cálculo Diferencial
Algoritmos	Programación Estructurada
Fundamentos de Ingeniería de Software	Matemáticas Discretas
Responsabilidad Social Universitaria	Cultura Maya
Álgebra Lineal	Probabilidad
Cálculo Integral	Diseño de Software
Programación Orientada a Objetos	Estructuras de Datos
Teoría de la Computación	Sistemas Operativos

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Arquitectura y Organización de Computadoras	Teoría de Lenguajes de Programación
Inferencia Estadística	Métricas de Software
Arquitecturas de Software	Aseguramiento de la Calidad del Software
Construcción de Software	Requisitos de Software
Desarrollo de Aplicaciones Web	Interacción Humano Computadora
Experimentación en Ingeniería de Software	Administración de Proyectos I
Verificación y Validación de Software	Mantenimiento de Software
Redes y Seguridad de Computadoras	Sistemas Distribuidos
Innovación Tecnológica	Taller de Emprendedores
Administración de Proyectos II	
Taller de Emprendedores	

La oferta de las asignaturas obligatorias durante los cursos de verano y en los periodos semestrales estará en función de los recursos humanos, infraestructura disponible, demanda de estudiantes y características de la asignatura.

12.6. Prácticas Profesionales

Las Prácticas Profesionales tienen un valor curricular de 8 créditos, y podrá inscribirse una vez cubierto 180 créditos del plan de estudios. El estudiante deberá realizar al menos 320 horas de práctica profesional, a través de práctica supervisada en el escenario real. Las prácticas profesionales se acreditarán a través de un proyecto que deberá ser autorizado al inicio de su desarrollo y avalado en su finalización, por el coordinador del programa educativo, quien será responsable de verificar la correspondencia del proyecto con la práctica profesional.

12.7. Servicio Social

El Servicio Social tienen un valor curricular de 12 créditos, y podrá inscribirse una vez cubierto el 70% de los créditos —252 créditos— del plan de estudios. El estudiante deberá realizar al menos 480 horas de servicio social, mediante práctica supervisada en el escenario real, en un tiempo de seis meses como mínimo y dos años como máximo.

El Servicio Social se acreditará a través de uno de los proyectos aprobados por el Departamento de Apoyo al Servicio Social.



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario



12.8. Emprendedores

Las actividades que promoverán el desarrollo del espíritu emprendedor e innovador en el alumno de la Licenciatura en Ingeniería de Software, se desarrollarán en el marco del Taller de Emprendedores, el cual resulta importante en un contexto donde el déficit de empleo y las acciones de impacto social requiere de personas con iniciativa propia y generadoras de cambio en la sociedad; es por ello que el propósito de esta asignatura es generar una actitud positiva hacia el emprendimiento como medio de superación y progreso continuo en lo personal, profesional y social. La asignatura de Taller de Emprendedores posee un valor curricular de 6 créditos. De acuerdo con la malla curricular propuesta, esta asignatura podrá ser cursada a partir del sexto semestre.

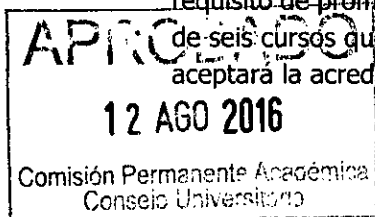
12.9. Movilidad

Los estudiantes podrán acreditar hasta un 50% de los créditos del Programa Educativo, en asignaturas de otros programas educativos de la UADY, así como de programas educativos de otras Instituciones de Educación Superior (IES) nacionales o extranjeras reconocidas. Para lo anterior, el estudiante deberá recibir la autorización de homologación, por parte de la Secretaría Académica, de las asignaturas a cursar en la institución receptora. En el caso de las asignaturas obligatorias se reconocerá el número de créditos que establece el programa educativo de la asignatura, y en el caso de asignaturas optativas y libres, se reconocerá el número de créditos del plan de estudios de la IES receptora. Cuando la IES receptora utilice una escala de calificaciones diferente al de la UADY, se aplicará una tabla de equivalencias para el reconocimiento del nivel de dominio de la asignatura.

12.10. Inglés como segundo idioma

El estudiante debe acreditar el dominio del idioma inglés en el nivel B1, de acuerdo al Marco de Referencia Europeo (2005) —promovido por el Programa Institucional de Inglés— desde su primera inscripción al Programa Educativo, y hasta finalizar el equivalente al sexto semestre (235 créditos). De no aprobar el nivel B1 al finalizar el plazo establecido, el estudiante no podrá seguir cursando las asignaturas que integran el plan de estudios, en tanto no acredite dicho nivel de dominio.

El Centro Institucional de Lenguas (CIL) de la UADY imparte cursos de idioma Inglés como parte del Programa Institucional de Inglés (PII). Este programa se ofrece a través de un currículo innovador, apoyado en las nuevas tecnologías y en modalidades flexibles de aprendizaje; dicho programa representa una alternativa para que los estudiantes de licenciatura logren acreditar el requisito de promoción relativo al inglés. El nivel B1 puede ser alcanzado por el estudiante a través de seis cursos que se ofrecen articulados con las asignaturas del plan de estudios; no obstante, se aceptará la acreditación del inglés en instituciones reconocidas por el CIL.



12.11. Titulación

El estudiante del Programa Educativo podrá obtener el título de Ingeniero(a) de Software con cualquiera de las siguientes opciones:

- A. Aprobar el total de los créditos de su plan de estudios y obtener desempeño satisfactorio, por lo menos, en 50% de las áreas que conforman el Examen General de Egreso de la Licenciatura en Ingeniería de Software (EGEL-ISOFT).
- B. Aprobar el total de los créditos de su plan de estudios y presentar una tesis, misma que deberá elaborarse durante el proceso de formación y no al finalizar el plan de estudios. El plan de estudios contempla asignaturas obligatorias que promueven en el estudiante competencias para el desarrollo de su tesis; por otro lado, el estudiante podrá seleccionar asignaturas optativas que profundicen sobre un área de interés en investigación.

En el momento en el que la UADY adopte un Examen General de Egreso (EGEL) distinto al EGEL-ISOFT —para el programa de LIS— los alumnos deberán acreditar por lo menos el 50% de las áreas que conformen dicho examen.

Los lineamientos generales del procedimiento y requisitos administrativos se establecen en los reglamentos aplicables de la UADY y en el Reglamento Interior de la Facultad vigentes.

12.12. Plan de liquidación

La Liquidación del Plan de Estudios 2009, se realizará de acuerdo a la siguiente estrategia:

- (i). A partir de agosto de 2016, las asignaturas correspondientes a los períodos semestrales recomendados en el mapa curricular del plan de estudios 2009, del tercer semestre y posteriores, se impartirán de acuerdo con el avance idóneo de la Generación XII, hasta su egreso en julio de 2019.
- (ii). Como estrategia de liquidación, para aquellos alumnos que por algún motivo el avance en su trayectoria escolar no fue de acuerdo a lo descrito en (i), a partir de agosto de 2016, y en paralelo con los semestres recomendados en la malla curricular del plan 2016, se ofrecerán nuevamente las asignaturas correspondientes a los semestres del plan 2009, hasta su conclusión en julio de 2020.

Las situaciones no previstas en los incisos anteriores serán resueltas por la Secretaría Académica en conjunto con el coordinador del programa de estudios. En la Tabla 30 se ilustra el Plan de Liquidación descrito anteriormente.

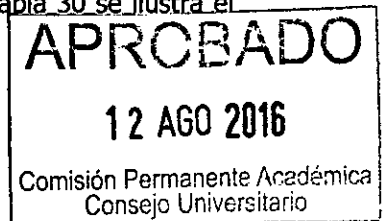


Tabla 30. Plan de liquidación.

	Semestres								
Generación XII UIS (Plan 2009)	III	IV	V	VI	VII	VIII			(I)
Repetición (Plan 2009)	I	II	III	IV	V	VI	VII	VIII	(II)
Curso Escolar	2016-2017		2017-2018		2018-2019		2019-2020		

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



13. PLAN DE DESARROLLO

A continuación se presenta el Plan de Desarrollo de la Licenciatura en Ingeniería de Software, elaborado de acuerdo a los lineamientos de la Coordinación de Licenciatura; esta sección se basa en el documento entregado en enero de 2013.

13.1 Visión a 2020

Ser una licenciatura con trascendencia nacional e internacional con un plan de estudios actualizado que responde a las necesidades de la región y del país, acorde con el modelo educativo vigente, para formar profesionales en el proceso de desarrollo y la evolución de sistemas de software a pequeña y gran escala, que solucionen problemas dentro de diferentes especialidades, utilizando las herramientas adecuadas para optimizar los recursos de tiempo y costo, con apego a la ética profesional.

13.2 Objetivos estratégicos

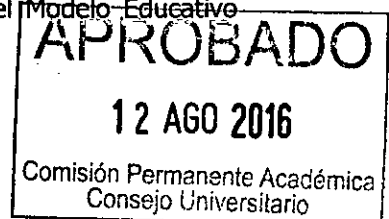
1. Contar con un programa educativo de alta calidad, con reconocimiento nacional e internacional.
2. Consolidar el cuerpo académico de Ingeniería de Software para apoyar al plan de estudios de la licenciatura.
3. Fortalecer y mantener actualizada la infraestructura para el desarrollo adecuado del plan de estudios.
4. Establecer programas que promuevan la experiencia profesional y la actualización continua de los egresados, para atender las necesidades de la sociedad.

Políticas y estrategias para hacer realidad la Visión

Para el objetivo estratégico 1. Contar con un programa educativo de alta calidad, con reconocimiento nacional e internacional.

Políticas y estrategias:

1. Mantener actualizado el Plan de estudios de la LIS con base en el Modelo Educativo vigente.



- a) Analizar y revisar el Modelo Educativo vigente.
 - b) Realizar cambios pertinentes al plan de estudios con base en las debilidades y amenazas identificadas.
2. Mantener actualizado el Plan de estudios de la LIS con base en los referentes nacionales e internacionales de las tendencias de los modelos curriculares.
- a) Analizar y revisar los modelos curriculares vigentes.
 - b) Analizar las necesidades locales y nacionales.
 - c) Realizar cambios pertinentes al plan de estudios con base en los resultados obtenidos en los análisis previos.
2. Re-acreditar y certificar el plan de estudios a través de asociaciones externas.
- a) Someter a evaluación el plan de estudios.
 - b) Considerar las recomendaciones de asociaciones externas para la acreditación y certificación en la actualización del plan de estudios.
3. Realizar el seguimiento de los egresados de la LIS, para conocer su desempeño profesional.
- a) Definir un programa para el seguimiento de los egresados de la LIS.
 - b) Implementar y dar continuidad al programa del seguimiento de egresados para proponer cambios en el plan de estudios.
4. Impulsar la movilidad nacional e internacional para los estudiantes de la Licenciatura en Ingeniería de Software.
- a) Promover la movilidad estudiantil.
 - b) Crear un catálogo de instituciones con las que se tenga convenios de movilidad.
 - c) Establecer un punto de contacto para el estudiante hacia la institución receptora, así como a los departamentos institucionales correspondientes.
5. Promover la formación integral de los alumnos.
- a) Promover el servicio social y de prácticas profesionales en estancias en empresas.
 - b) Promover la participación de los estudiantes en actividades culturales y deportivas.
 - c) Promover el estudio del lenguaje Inglés.
 - d) Desarrollar actividades como concursos, congresos, exposiciones, talleres, entre otros, que promuevan la innovación y creatividad en los estudiantes.
 - e) Fomentar la responsabilidad social a través de programas del área de formación integral institucional.

APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

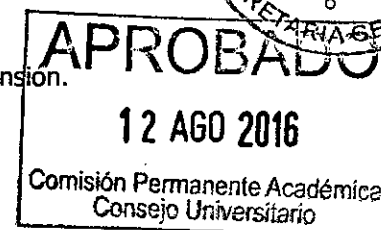


6. Impulsar la movilidad nacional e internacional de los profesores.
 - a) Definir un programa de intercambio nacional e internacional de experiencias académicas.
 - b) Promover el programa de intercambio.

Para el objetivo estratégico 2. Consolidar el cuerpo académico de Ingeniería de Software para apoyar al plan de estudios de la licenciatura.

Políticas y estrategias:

1. Buscar la definitividad de los miembros de los CA asociados al programa educativo.
 - a) Gestionar dos plazas de maestría y una de doctor con perfil de ingeniería del software.
2. Mejorar el grado académico de los integrantes del CA.
 - a) Establecer un plan escalonado para la obtención del grado de doctor de los miembros del CA, de acuerdo a la perspectiva de crecimiento individual.
 - b) Gestionar los recursos necesarios para que cada miembro, de acuerdo a su plan, obtenga el grado de doctor.
3. Incrementar la planta docente con grado de doctor.
 - a) Identificar un doctor en el área que el CA pretenda desarrollar.
 - b) Gestionar una plaza de doctor.
4. Promover las actividades de investigación.
 - a) Equilibrar las funciones de los miembros del CA, dándole énfasis a la investigación.
 - b) Desarrollar de manera colegida las líneas de investigación propuestas en el CA.
 - c) Priorizar las actividades de investigación en el presupuesto establecido para el CA.
5. Establecer redes de investigación con otras instituciones.
 - a) Establecer convenios con otras instituciones del área de computación.
 - b) Desarrollar proyectos de investigación conjuntos.
6. Promover la experiencia profesional de los miembros del CA.
 - a) Motivar la participación de los docentes en proyectos de extensión.



7. Promover la habilitación pedagógica de cada uno de los miembros del CA.

- a) Motivar a tomar los cursos de pedagogía y tutoría.

Para el objetivo estratégico 3. Fortalecer y mantener actualizada la infraestructura para el desarrollo adecuado del plan de estudios.

Políticas y estrategias:

1. Generar recursos para la actualización y mantenimiento de los equipos de cómputo de telecomunicaciones.

- a) Motivar el desarrollo de proyectos de investigación y de aplicación del conocimiento.
- b) Motivar la participación de profesores en la atención de necesidades del sector privado.

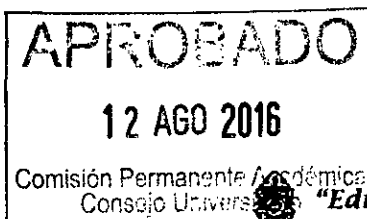
Para el objetivo estratégico 4. Establecer programas que promuevan la experiencia profesional y la actualización continua de los egresados, para atender las necesidades de la sociedad.

1. Establecer convenios de vinculación, que permitan desarrollar la experiencia profesional del estudiante.

- a) Crear un área de vinculación que promueva la firma de convenios y el acceso a escenarios reales de aprendizaje en beneficio de la formación profesional de los estudiantes del programa.
- b) Definir convenios de vinculación con la iniciativa pública y privada con fines de Prácticas Profesionales o Servicio Social para los estudiantes del programa.
- c) Promover los convenios de vinculación establecidos.

2. Impulsar un área de servicios de TI al interior de la facultad, que atienda necesidades del público en general y que desarrolle las habilidades y competencias de los alumnos en las áreas de cómputo.

- a) Formalizar los procesos y servicios.
- b) Generar una cartera de proyectos.
- c) Motivar el involucramiento de los profesores en los proyectos.
- d) Motivar la participación de los alumnos y egresados.



3. Establecer esquemas que promueven la certificación de los egresados en al menos un área de interés.
 - a) Analizar las necesidades del mercado para determinar las áreas pertinentes de certificación.
 - b) Establecer un centro de certificación de egresados.
 - c) Motivar la participación de los profesores y certificar a los interesados.
 - d) Fomentar la participación de los egresados.

4. En conjunto con la Unidad de Extensión de la Facultad, crear programas de actualización para los egresados.
 - e) Analizar las necesidades del mercado.
 - f) Crear cursos, diplomados, talleres, entre otros.
 - g) Difundir los programas de actualización definidos.



Indicadores y metas 2011 – 2020

Tasas de egreso y titulación

Hasta la fecha, de las cuatro generaciones que han egresado, la tasa de egreso es de 17% con un total de 29 egresados. De éstos, se han titulado el 59%. Los valores propuestos de las tasas de egreso y titulación para el año 2015 son 35% y 70%, respectivamente. Para el 2020, se pretende alcanzar una tasa de egreso del 50%, y una tasa de titulación del 90%.

La opción de titulación más utilizada por los egresados es Promedio general, ya que les permite integrarse rápidamente al campo laboral o realizar estudios de postgrado. El 88% (15) de los titulados ha optado por esta modalidad. Las otras dos opciones han sido Examen General de Egreso y Curso en opción de titulación.

Tabla 30. Indicadores de Egreso y Titulación.

(1) Nº. de Generación	(2) Año de Ingreso	(3) Año de Egreso	(4) Ingreso	(5) Egresados	(6) Índice de Egreso (5) / (4) %	(7) Titulados	(8) Índice de titulación (7) / (5) %
1	2004	2008	21	7	33	7	100
2	2005	2009	37	19	51	17	90
3	2006	2010	35	12	34	11	92

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



4	2007	2011	36	12	31	11	92
5	2008	2012	44	18	39	17	100
6	2009	2013	35	19	40	10	71
Total			208	81	38	73	90

Tasas de retención y deserción

De la tabla siguiente se observan las tasas de retención y deserción de las primeras siete generaciones de la Licenciatura en Ingeniería de Software. Las tasas de retención en los dos primeros años presentan valores aceptables, en promedio arriba del 70%. Por lo anterior, para el 2020 se pretende mantener estos valores por arriba del promedio.

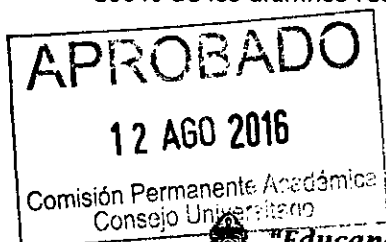
Tabla 31. Tasas de Retención y Deserción.

Generación	Alums Inscr.	Evolución de la Retención %					Egresados/ Pendientes	Deserción
		Primer Año	Segundo Año	Tercer Año	Cuarto Año			
2004	21	90	71	57	43	33	0	67
2005	37	89	76	68	68	27	27	46
2006	35	77	54	54	54	14	23	63
2007	36	97	89	86	78	19	42	39
2008	44	98	95	86				14
2009	35	97	94					6
2010	54	93						7

Porcentaje de estudiantes que reciben tutoría

La Facultad de Matemáticas se apeg a al programa de tutorías establecido por la UADY. Este programa considera que la tutoría es un proceso de asesoramiento y orientación de tipo personal y académico a lo largo del proceso formativo para mejorar el rendimiento del estudiante, solucionar problemas escolares, desarrollar hábitos de estudio, de trabajo, de reflexión y de convivencia social.

El programa de tutorías adquiere una importancia significativa para el modelo del Plan semiflexible, ya que en esta modalidad será el tutor quien oriente al estudiante en la configuración y en su caso, adecuación del mapa curricular a seguir para su formación. Por lo anterior, en el caso de los alumnos de nuevo ingreso, el tutor es asignado por el Comité de Tutorías, por lo cual el 100% de los alumnos reciben tutorías.



Tiempo promedio empleado por los estudiantes para cursar y aprobar la totalidad de las materias del plan de estudios

El tiempo promedio empleado por los estudiantes para cursar y aprobar la totalidad de las materias del plan de estudios es de 9 semestres.

Número y porcentaje de estudiantes con TDS y TDSS en el EGEL

Hasta la fecha han presentado cuatro estudiantes y todos han obtenido el TDSS. Se pretende que para el 2015, al menos el 50% de los egresados presenten el EGEL, y para el 2020 se espera alcanzar que el 100% de los egresados presente el EGEL.

Número y porcentaje de PTC que participan en el PE:

Actualmente, el número de profesores que participan en la Licenciatura en Ingeniería de Software es de 56, de los cuales:

Con posgrado

Actualmente, el porcentaje de profesores con posgrado es de 68%. Se espera que para el 2020 el 100% de los profesores cuente con el nivel de posgrado.

Con doctorado

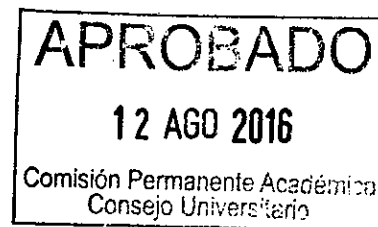
El porcentaje de profesores con doctorado que participan en el programa educativo es de 16%. Para el 2020 se espera que al menos el 30% de la planta académica de la licenciatura tenga el grado de doctorado.

Con Perfil Deseable

La Facultad de Matemáticas cuenta con 38 profesores con perfil PRÓMEP, de los cuales 16 participan en el desarrollo de la Licenciatura en Ingeniería de Software, que representa el 28% de los profesores que participan en el programa educativo. Se espera que para el año 2020 al menos 30 de los profesores que participen en la licenciatura cuenten con el perfil deseable.

Con SNI

Actualmente, la Facultad de Matemáticas cuenta con 15 profesores que pertenecen al SNI, de los cuales 6 participan en el programa educativo, que representa el 10% de los profesores de la licenciatura. Al 2020 se espera que al menos 12 profesores de la Licenciatura en Ingeniería de Software pertenezcan al SNI.



Conclusiones

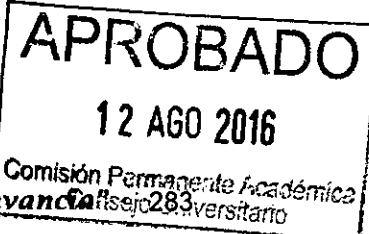
La Licenciatura en Ingeniería de Software es una de las primeras carreras especializadas en esta área que se han ofrecido en el país. El compromiso es que nuestros egresados estén altamente capacitados en la aplicación de conocimientos y habilidades requeridas para desempeñar el trabajo profesional que resuelva los problemas que enfrenta la sociedad, en un marco de disciplina, ética, trabajo en equipo y responsabilidad social. De igual manera, el compromiso de mantener el plan de estudios actualizado y acorde con los lineamientos de la Universidad y con las instituciones externas, de manera que permita continuar con la tarea de formar profesionistas para nuestro país.

El trabajo a realizar es fuerte y de gran importancia, como se aprecia en las actividades propuestas para el desarrollo del programa de estudio, y que se logrará con la participación efectiva de los académicos y de la Administración de la Facultad.

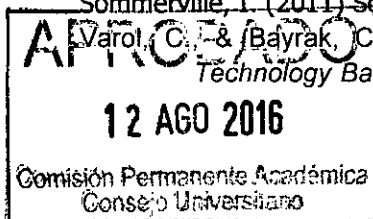


14. REFERENCIAS

- ABET, "ABET Accredited Engineering Programs". April 3, 2007. Retrieved 2007-04-03.
- ACM, C. M. ACM code of ethics and professional conduct. 1992.
- ANIEI & CONAIC (2013). Modelos Curriculares del Nivel Superior de Informática y Computación.
- Ardis, M., & Ford, G. (1989). SEI (Software Engineering Institute) Report on Graduate Software Engineering Education, 1989 (No. CMU/SEI-89-TR-21). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
- Baetjer, H. (1998). Software as Capital: An Economic Perspective on Software Engineering. The Institute of Electrical and Electronics Engineers. *Inc.: Piscataway, NJ.*
- Bauer, F. L. (2003), The Origin of Software Engineering—Letter to Dr. Richard Thayer in Software Engineering, by Thayer, R. H. and M. Dorfman (eds.) (2003), pp. 7–8, John Wiley & Sons, Inc., N.J.
- Blum, B. I. (1992), Software Engineering: A Holistic View, Oxford University Press, New York.
- Boehm B. A view of 20th and 21st century software engineering In Proceedings of the 28th international conference on Software engineering, pp. 12-29, 2006.
- Buxton, J. N., & Randell, B. (Eds.). (1970). Software Engineering Techniques: Report on a Conference Sponsored by the NATO Science Committee. NATO Science Committee; available from Scientific Affairs Division, NATO.
- CANIETI (2015). Estudio de Indicadores de la Industria 2015. Cámara Nacional de la Industria Electrónica, de Telecomunicaciones y Tecnologías de la Información.
- CEDENE (2015). Estudio de Mercado Laboral 2015. Centro de Desarrollo de Negocios de la Facultad de Contaduría y Administración de la UADY.
- CENEVAL, Centro Nacional de Evaluación para la Educación Superior, A.C. <http://www.ceneval.edu.mx/>, Acceso al recurso en febrero, 2014.
- CONAIC (2013). Marco referencia para la Acreditación de Programas Académicos de Informática y Computación. Educación Superior.
- CONCYTEY (2010). Las condiciones para la innovación, el desarrollo tecnológico y la vinculación productiva en Yucatán.
- Congreso del Estado de Yucatán (2011). Ley de Fomento al Desarrollo Científico, Tecnológico y a la Innovación del Estado de Yucatán.
- Cowling, A. J. 1999. The first decade of an undergraduate degree program in software engineering. *Ann. Softw. Eng.* 6, 1-4 (Apr. 1999), 61-90.
- DPSE, Degree programs in Software Engineering, 2010, <http://www.gradschools.com/search-programs/software-engineering>
- EPS-LIS (2015) Estudio de Pertinencia Social. Grupo Formador de la Propuesta de Modificación del PE-LIS.
- FA-LIS (2015) Estudio de Factibilidad. Grupo Formador de la Propuesta de Modificación del PE-LIS. México.
- Fairley, R. E., & Gibbs, N. E. (1987). Software Engineering Institute and Wang Institute of Graduate



- Studies on Software engineering education: the educational needs of the software community.
- Ford, G. (1994). The progress of undergraduate software engineering education. *ACM SIGCSE Bulletin*, 26(4), 51-55.
- Gobierno de Yucatán y CANIETI (2015). La Estrategia Digital Yucatán 2015-2020. Recuperado de http://www.estrategiadigitalyucatan.org/EDY_BOOK.pdf Último acceso: 11 de enero, 2016.
- GRUPO, D. MAGISTERIO/ANECA (2004). La adecuación de las Titulaciones de Maestro al Espacio Europeo de Educación Superior.
- IEEE Standards Coordinating Committee. (1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos, CA: IEEE Computer Society.
- INEGI, C. D. P. (2011). Vivienda 2010. Resultados definitivos, México.
- ISO/IEC TR 19759, Software Engineering - Guide to the Software Engineering Body of Knowledge SWEBOK, 2005.
- Lehman, M. M. (1987). The software engineering first degree at Imperial College, London. In *Software Engineering Education* (pp. 172-181). Springer New York.
- MEFI (2013) Modelo Educativo para la Formación Integral. Dirección General de Desarrollo Académico. Universidad Autónoma de Yucatán. México.
- MEyA (2002). Modelo Educativo y Académico de la UADY. Dirección General de Desarrollo Académico. Universidad Autónoma de Yucatán. México.
- Microsoft (2014). Annual Report. Recuperado de <https://www.microsoft.com/investor/reports/ar14/index.html>
- Naur, P. and Randell, B. (eds.), *Software Engineering: Report on a Conference Sponsored by NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968*, Scientific Affairs Division, NATO, Brussels, 1969.
- Naveda, J. F., & Lutz, M. J. (1997, April). Crafting a baccalaureate program in software engineering. In *Software Engineering Education & Training. Tenth Conference on* (pp. 74-80). IEEE.
- PDI (2015). Plan de Desarrollo Institucional 2014-2022. México: Universidad Autónoma de Yucatán.
- PA (2012) Plan de Acción EMSyS UADY. Universidad Autónoma de Yucatán. Plan de Acción de la Educación Media Superior y Superior en el Estado de Yucatán.
- PDE (2013) Plan estatal de desarrollo 2012-2018 Yucatán. Yucatán: Gobierno del Estado de Yucatán.
- PND (2013). Plan Nacional de Desarrollo 2013 – 2018. Gobierno de la República de México.
- Pressman, R. (2005) *Ingeniería de Software: Un enfoque práctico*. (6ª. edición.). México. Mc Graw Hill.
- RedTIC. (2011) Libro Blanco TIC. México.
- Sommerville, I. (2002) *Ingeniería de Softwar*. (6a. ed.). México. Addison Wesley.
- Sommerville, I. (2011) *Software Engineering*. (9th. ed.). USA. Pearson Education/Addison-Wesley.
- Varol, C. & Bayrak, C. (2005, July). Applied software engineering education. In *Information Technology Based Higher Education and Training, 2005. ITHET 2005. 6th International*



- Conference on* (pp. T3C-25). IEEE.
- Villalobos, M., & Gutiérrez, A. Investigación sobre las prácticas de ingeniería de software en México. Asociación Mexicana de Calidad para la Ingeniería de Software (AMCIS) y el Laboratorio de Sistemas de Información del CIC-IPN.(Agosto de 2001).
- Wirth, N. (2008). A brief history of software engineering. *IEEE Annals of the History of Computing*, 1(3), 32-39.



ANEXO A. PROFESORES QUE ELABORARON LOS PROGRAMAS DE ESTUDIOS

Lista de profesores que elaboraron los programas de estudio sintéticos de las asignaturas obligatorias.

Asignatura	Profesores
Administración de Proyectos I	M.A.T.I. Edwin J. León Bojórquez M. en C. Carlos Benito Mojica Ruiz
Administración de Proyectos II	M. en C. Cinthia Maribel González Segura Dr. Emilio Gabriel Rejón Herrera
Álgebra Intermedia (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Álgebra Lineal (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Álgebra Superior (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Algoritmos (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Arquitectura y Organización de Computadoras (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Arquitecturas de Software	M. en C. Juan Francisco Garcilazo Ortiz Dr. Francisco José Moo Mena
Aseguramiento de la Calidad del Software	M. en C. Antonio Armando Aguilera Güémez M. en C. Carlos Benito Mojica Ruiz
Cálculo Diferencial (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Cálculo Integral (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D
Constructor de Software	M. en C. Antonio Armando Aguilera Güémez Dr. Juan Pablo Ucan Pech
Cultura Maya	Personal Institucional
Desarrollo de Aplicaciones Web	LCC. Rodrigo Esparza Sánchez, ED. MCC Cinthia Maribel González Segura Dr. Víctor Hugo Menéndez Domínguez
Diseño de Bases de Datos	M. en C. Luis Ramiro Basto Díaz M. en C. Maximiliano Canché Euán M.T.I. Julio César Díaz Mendoza M. en C. Juan Francisco Garcilazo Ortiz
Diseño de Software	Dr. José Luis López Martínez Dr. Francisco José Moo Mena
Estructuras de Datos	M. en C. Luis Ramiro Basto Díaz M. I. Jorge Carlos Reyes Magaña Dr. Raúl Antonio Aguilar Vera
Experimentación en Ingeniería de Software	

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Fundamentos en Ingeniería de Software	Dr. José Luis Batún Cutz Dr. Raúl Antonio Aguilar Vera Dr. José Luis López Martínez
Geometría Analítica (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D. M. en C. Luis Colorado Martínez
Inferencia Estadística	M.C.M. Ernesto Antonio Guerrero Lara M. en C. Salvador Medina Peralta
Innovación Tecnológica	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D. Dr. Emilio Gabriel Rejón Herrera
Interacción Humano Computadora	M. en C. Edgar Cambranes Martínez M. en C. Carlos Miranda Palma
Mantenimiento de Software	MCC. Cinhtia Maribel González Segura Dr. Víctor Hugo Menéndez Domínguez
Matemáticas Discretas (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Métricas de Software	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D. M.C. Salvador Medina Peralta M.C.M. Neyfis Vanessa Solís Baas
Probabilidad	M.C. Luis Colorado Martínez M.C.M. Ernesto Antonio Guerrero Lara
Programación Estructurada (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Programación Orientada a Objetos (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Redes y Seguridad de Computadoras	M.T. Sergio Antonio Cervera Loeza L.C.C. Rodrigo Esparza Sánchez
Requisitos de Software	M.A.T.I. Edwin J. León Bojórquez M. C. Carlos Benito Mojica Ruiz
Responsabilidad Social Universitaria	Personal institucional
Sistemas Distribuidos (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Sistemas Operativos	M.T.I. Julio César Díaz Mendoza Dr. Jorge Gómez Montalvo
Taller de Emprendedores (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Teoría de la Computación (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Teoría de Lenguajes de Programación (Actualización)	M.G.T.I. María Enriqueta Castellanos Bolaños, E.D.
Verificación y Validación de Software	M. en C. Edgar Antonio Cambranes Martínez Dr. Juan Pablo Ucán Pech



APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





ANEXO B. CUESTIONARIO PARA PROFESORES

UNIVERSIDAD AUTONOMA DE YUCATAN
FACULTAD DE MATEMATICAS
LICENCIATURA EN INGENIERÍA DE SOFTWARE

CUESTIONARIO PARA PROFESORES

OBJETIVO

Recabar información sobre las asignaturas que los profesores han impartido en el plan de estudios de la Licenciatura en Ingeniería de Software, con la finalidad de evaluar y actualizar dicho plan de estudios.

INSTRUCCIONES

El siguiente cuestionario consta de cuatro secciones. Las secciones I y II evalúan el Plan de Estudios, por lo que le pedimos la respuesta una sola vez. Las secciones III y IV deberán ser respondidas por cada asignatura que haya impartido en el Plan de Estudios.

Lea detenidamente las preguntas y marque con una X en el cuadro correspondiente a la respuesta que considere apropiada y en su caso, escriba la razón de su respuesta en el espacio proporcionado para tal fin.

Si requiere espacio adicional, vaya a la parte final del documento y escriba el número de la pregunta a la que desee dar respuesta.

MUCHAS GRACIAS POR SU COLABORACIÓN

SECCION I: SOBRE EL PLAN DE ESTUDIOS DE LA LICENCIATURA

a) Con base en la información proporcionada en el Anexo A, indique si está de acuerdo con:

	SI	NO	No sé
1. El objetivo general es pertinente.			
2. Los objetivos específicos son congruentes con el objetivo general.			
3. Los objetivos específicos son congruentes con el perfil del egresado.			

En caso de que su respuesta sea NO en algún aspecto, especifique por qué:



APROBADO

12 AGO 2016

Comisión Permanente Académica
Consejo Universitario

b) Con base en la información proporcionada en el Anexo B, indique si considera que el Plan de Estudios de la licenciatura en Ingeniería de Software requiere:

	SI	NO
1. Agregar alguna asignatura obligatoria.		
2. Excluir alguna asignatura obligatoria.		

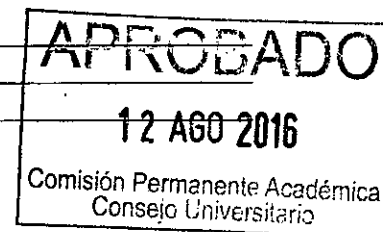
En caso de que su respuesta sea SI en algún aspecto, especifique la modificación recomendada:

SECCION II: SOBRE LA INFRAESTRUCTURA

Señale cuál es su opinión sobre los siguientes recursos:

	Excelente	Bien	Regular	Deficiente	No sé
1. Aulas de clases para el desarrollo de los cursos					
2. Disponibilidad de aulas de cómputo o laboratorios					
3. Libros disponibles en la biblioteca del campus					
4. Número de ejemplares de libros en la biblioteca					
5. Revistas y otros materiales de la biblioteca					
6. Software disponible en el centro de cómputo					
7. Disponibilidad de equipo de cómputo y audiovisual					
8. Servicio de internet					
9. Cubículo para profesores					
10. Número de alumnos por salón					
11. Otro (especifique):					

Comentarios:



ASIGNATURA 1: _____

SECCION III: SOBRE LOS PROGRAMAS DE ASIGNATURA

Respecto al programa de la asignatura, considera que:

	SI	NO	No sé
1. Contribuye al cumplimiento de los objetivos del plan de estudios			
2. Contiene objetivos pertinentes			
3. Contiene objetivos congruentes con el perfil del egresado			
4. Contiene objetivos acordes con las tendencias del área de la asignatura			
5. El número total de horas es suficiente			
6. Tiene un balance adecuado de teoría-práctica			
7. Los criterios de evaluación son adecuados			
8. La bibliografía está actualizada			
9. Los antecedentes académicos deseables son adecuados			
10. Se puede impartir en la modalidad de curso de verano			
11. Se puede impartir en la modalidad de curso en línea			

En caso de que su respuesta sea NO en algún aspecto, especifique por qué:

Si requiere agregar algún tema, mencione cuál:

Si requiere eliminar

algún tema, mencione cuál:

Si requiere modificar el orden del temario, mencione cómo:



SECCION IV: SOBRE EL PROCESO DE ENSEÑANZA APRENDIZAJE

Señale cuáles de los siguientes recursos utiliza para el desarrollo de la asignatura que imparte:

	SI	NO		SI	NO
Exposición del docente			Taller para resolución de problemas		

AFECTADO
 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Estudio de casos		Visitas a centros profesionales	
Lluvia de ideas		Visitas a centros de investigación	
Trabajo por equipos de alumnos		Realización de foros virtuales	
Manejo de portafolio		Realización de seminarios	
Uso de material didáctico		Integración de grupos de investigación	
Empleo de tecnologías de información		Estrategias para uso de segundo idioma	
Grupos de discusión presencial		Asesorías individuales	
Grupos de discusión no presencial		Tutorías	
Laboratorio o taller de prácticas		Otro (especifique):	

ASIGNATURA 2: _____

SECCION III: SOBRE LOS PROGRAMAS DE ASIGNATURA

Respecto al programa de la asignatura, considera que:

	SI	NO	No sé
1. Contribuye al cumplimiento de los objetivos del plan de estudios			
2. Contiene objetivos pertinentes			
3. Contiene objetivos congruentes con el perfil del egresado			
4. Contiene objetivos acordes con las tendencias del área de la asignatura			
5. El número total de horas es suficiente			
6. Tiene un balance adecuado de teoría-práctica			
7. Los criterios de evaluación son adecuados			
8. La bibliografía está actualizada			
9. Los antecedentes académicos deseables son adecuados			
10. Se puede impartir en la modalidad de curso de verano			
11. Se puede impartir en la modalidad de curso en línea			

En caso de que su respuesta sea NO en algún aspecto, especifique por qué:

Si requiere agregar algún tema, mencione cuál:

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



_ Si requiere eliminar algún tema, mencione cuál:

_ Si requiere modificar el orden del temario, mencione cómo:

SECCION IV: SOBRE EL PROCESO DE ENSEÑANZA APRENDIZAJE

Señale cuáles de los siguientes recursos utiliza para el desarrollo de la asignatura que imparte:

	SI	NO		SI	NO
Exposición del docente			Taller para resolución de problemas		
Estudio de casos			Visitas a centros profesionales		
Lluvia de ideas			Visitas a centros de investigación		
Trabajo por equipos de alumnos			Realización de foros virtuales		
Manejo de portafolio			Realización de seminarios		
Uso de material didáctico			Integración de grupos de investigación		
Empleo de tecnologías de información			Estrategias para uso de segundo idioma		
Grupos de discusión presencial			Asesorías individuales		
Grupos de discusión no presencial			Tutorías		
Laboratorio o taller de prácticas			Otro (especifique):		

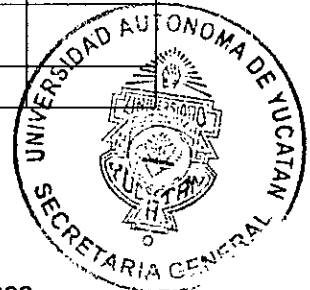
ASIGNATURA 3: _____

SECCION III: SOBRE LOS PROGRAMAS DE ASIGNATURA

Respecto al programa de la asignatura, considera que:

	SI	NO	No sé
1. Contribuye al cumplimiento de los objetivos del plan de estudios			
2. Contiene objetivos pertinentes			
3. Contiene objetivos congruentes con el perfil del egresado			
4. Contiene objetivos acordes con las tendencias del área de la asignatura			
5. El número total de horas es suficiente			

A
ADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



6. Tiene un balance adecuado de teoría-práctica			
7. Los criterios de evaluación son adecuados			
8. La bibliografía está actualizada			
9. Los antecedentes académicos deseables son adecuados			
10. Se puede impartir en la modalidad de curso de verano			
11. Se puede impartir en la modalidad de curso en línea			

En caso de que su respuesta sea NO en algún aspecto, especifique por qué:

Si requiere agregar algún tema, mencione cuál:

_ Si requiere eliminar algún tema, mencione cuál:

Si requiere modificar el orden del temario, mencione cómo:



SECCION IV: SOBRE EL PROCESO DE ENSEÑANZA APRENDIZAJE

Señale cuáles de los siguientes recursos utiliza para el desarrollo de la asignatura que imparte:

	SI	NO		SI	NO
Exposición del docente			Taller para resolución de problemas		
Estudio de casos			Visitas a centros profesionales		
Lluvia de ideas			Visitas a centros de investigación		
Trabajo por equipos de alumnos			Realización de foros virtuales		
Manejo de portafolio			Realización de seminarios		
Uso de material didáctico			Integración de grupos de investigación		
Empleo de tecnologías de información			Estrategias para uso de segundo idioma		
Grupos de discusión presencial			Asesorías individuales		
Grupos de discusión no presencial			Tutorías		
Laboratorio o taller de prácticas			Otro (especifique):		

APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



Comentarios generales:

ANEXO A

LICENCIATURA EN INGENIERÍA DE SOFTWARE

Objetivo General del Plan de Estudios:

General:

Formar profesionales en el *proceso de desarrollo y la evolución de sistemas de software a pequeña y gran escala*, que solucionen problemas dentro de diferentes áreas de concentración, utilizando las herramientas adecuadas para optimizar los recursos de tiempo y costo, con apego a la ética profesional

Específicos:

Explique y aplique un proceso de desarrollo de software sistémico acorde a la magnitud y complejidad de proyectos de aplicación, sean o no éstos nuevos desarrollos, tomando en cuenta la evolución y el cambio de los mismos.

Describa y aplique las herramientas necesarias para la especificación, diseño, verificación y validación de sistemas de software.

Se desempeñe en al menos un área de concentración, trabaje y se comunique de forma profesional en equipos interdisciplinarios.

Aplique el conocimiento y las habilidades para mejorar el proceso de desarrollo de software.

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario



Perfil de egreso:

El egresado del plan de estudios de la Licenciatura en Ingeniería de Software debe contar con las siguientes características:

Conocimientos sobre:

Metodologías y técnicas para el desarrollo de software a pequeña y gran escala, tales como desarrollo de requerimientos, interacción humano-computadora, diseño, construcción, evolución, pruebas y aquellos relacionados con la administración, como el aseguramiento de la calidad y la administración de proyectos.

Las ciencias computacionales, que le permitan la aplicación viable de algoritmos, estructuras de datos y lenguajes de programación en el desarrollo de software.

Las herramientas para la especificación y el modelado de sistemas de software, así como para la evaluación de los procesos y productos de software para su mejora.

Administración y optimización de tiempo y costo de los recursos involucrados en el desarrollo de software.

El entorno social para describir los factores sociales, políticos, ecológicos y éticos relacionados con el desarrollo tecnológico y el desempeño profesional.

Las áreas de investigación relacionadas con la Ingeniería de Software.

Habilidades para:

Trabajar como parte de un equipo en el desarrollo y evolución de productos de software.

Explicar el proceso para determinar las necesidades del cliente y traducirlos a requisitos de software.

Reconciliar objetivos en conflicto, considerando compromisos con las limitaciones de costo, tiempo, conocimiento, sistemas existentes y de las organizaciones involucradas.

Diseñar soluciones apropiadas en una o más áreas de concentración, usando enfoques de ingeniería que integren temas éticos, sociales, legales y económicos.

Entender y aplicar teorías, modelos y técnicas que provean una base para el diseño, desarrollo, verificación e implantación del software.

Negociar y trabajar profesionalmente, liderar cuando sea necesario y comunicarse efectivamente con los interesados en un ambiente de Ingeniería de Software.

Actitudes de:

Liderazgo en equipos de trabajo multidisciplinarios.

Perseverancia en la solución de problemas.

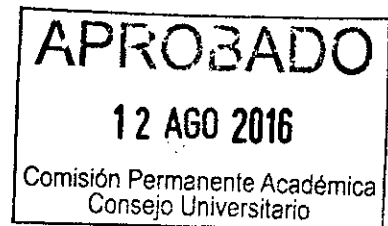
Capacidad de mantenerse actualizado en su área de trabajo.

Afán de superación.

Responsabilidad y ética en su desempeño profesional.

Conducta emprendedora e innovadora.

Conciencia de la realidad social y responsabilidad ecológica.



ANEXO B

I	II	III	IV	V	VI	VII	VIII
Cálculo Diferencial	Cálculo Integral	Teoría de la Computación	Arquitectura de Computadoras	Redes y Seguridad de Computadoras	Programación en la Web	Métodos Formales en Especificación y Diseño de SW	Métricas de Software
Álgebra Superior I	Álgebra Superior II	Álgebra Lineal	Teoría de Lenguajes de Programación	Bases de Datos	Sistemas Distribuidos	Administración de Proyectos I	Administración de Proyectos II
Fundamentos de Programación	Programación	Estructuras de Datos	Sistemas Operativos	Arquitecturas de Software	Área de Concentración I. A.	Área de Concentración I. B.	Área de Concentración I. C.
Metodología de la Investigación	Matemáticas Discretas	Interacción Humano Computadora	Diseño de Software	Construcción y Evolución de Software	Aseguramiento de la Calidad del Software	Evaluación de Proyectos	Área de Concentración II. B.
Fundamentos de Ingeniería de Software	Gestión Tecnológica	Probabilidad	Inferencia Estadística	Diseño de Experimentos en I-SW	Desarrollo de Requisitos de Software	Área de Concentración II. A.	Área de Concentración II. C.
						Taller de Servicio Social	Taller de Prácticas Profesionales

APROBADO
 12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



ANEXO C. CUESTIONARIO PARA ALUMNOS

Evaluación interna del plan de estudios de Licenciatura en Ingeniería de Software Cuestionario para alumnos

Plan de estudios

1) ¿Consideras que en alguna de las asignaturas del plan de estudios requeriste temas o conocimientos previos, adicionales a los del Plan de Estudios, para lograr un mejor aprovechamiento? (Consulta las asignaturas en el mapa curricular adjunto)

Asignatura	Tema

2) ¿Consideras que alguna asignatura incluye temas que no hayan contribuido con tu formación?

Asignatura	Tema

3) ¿Consideras que en alguna de las áreas de conocimiento del plan de estudios faltaron contenidos por incluir? (Consulta el mapa curricular adjunto)

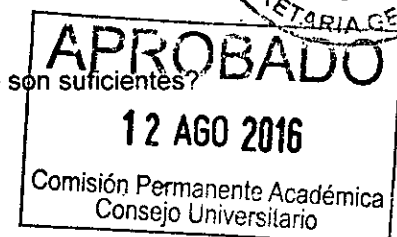
Area de Conocimiento	Contenido

4) Los conocimientos y habilidades adquiridos durante tu formación académica han sido útiles en la elaboración de tareas, proyectos, prácticas profesionales, o servicio social.

Totalmente en desacuerdo	En desacuerdo	Sin opinión	De acuerdo	Totalmente de acuerdo
--------------------------	---------------	-------------	------------	-----------------------

¿Por qué?

5) ¿Consideras que las áreas de concentración que se ofrecen regularmente son suficientes?





¿Por qué? Sí () No ()

Four horizontal lines for writing an answer to the first question.

Infraestructura

1) ¿Consideras que el Campus de CEI cuenta con instalaciones académicas apropiadas, como aulas, laboratorios, salas de cómputo, áreas de estudio, biblioteca?

¿Por qué? Sí () No ()

Two horizontal lines for writing an answer to question 1.

2) ¿Consideras que la Facultad cuenta con equipo de cómputo adecuado para satisfacer las demandas académicas?

¿Por qué? Sí () No ()

Two horizontal lines for writing an answer to question 2.

3) ¿Consideras que el equipo de cómputo de la Facultad se encuentra disponible para la realización de actividades extraclase?

Nunca	A veces	Casi siempre	Siempre
-------	---------	--------------	---------

4) ¿Consideras que los servicios de Internet que la Facultad proporciona son adecuados?

¿Por qué? Sí () No ()

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





5) ¿Consideras que es apropiado el software que se utiliza para las actividades académicas?

Sí () No ()
¿Por qué?

6) ¿Consideras que el acervo bibliográfico disponible en la biblioteca del CCEI es el apropiado?

Sí () No ()
¿Por qué?

7) ¿Consideras que la biblioteca del CCEI cuenta con el acervo bibliográfico suficiente?

Sí () No ()
¿Por qué?

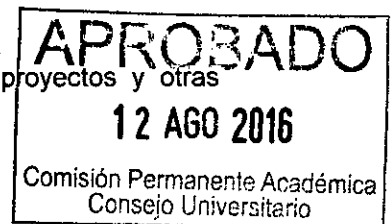
Personal docente y el proceso enseñanza-aprendizaje

1) ¿Consideras que los conocimientos académicos del personal docente, en las asignaturas que imparte, son apropiados para alcanzar o cubrir los objetivos del programa de estudios?

Sí () No ()
¿Por qué?



2) ¿consideras que la evaluación de cada asignatura (pruebas parciales, proyectos y otras actividades) se basa en los criterios establecidos en la programa de estudios?



Sí ()
¿Por qué?

No ()

3) ¿Han sido cubiertos completamente los programas de estudio de las asignaturas?

Sí ()
¿Por qué?

No ()

4) ¿Considera que la licenciatura se encuentra alineado con los objetivos y el enfoque presentado al inicio de la misma en el Plan de Estudios?

Sí ()
¿Por qué?

No ()



Administración

1) Al inicio de la licenciatura ¿la administración proporcionó información general sobre los objetivos y el enfoque de la carrera?

Sí ()

No ()

2) Al inicio de la licenciatura ¿fue proporcionada información general por parte de los responsables sobre el funcionamiento de la misma, en cuanto a horarios, servicio secretarial, trámites a realizar, situaciones especiales, etcétera, por parte de los responsables?

Sí ()

No ()

3) En caso de respuesta haya sido afirmativa en la pregunta anterior ¿fue apropiada la información?

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario





Sí ()

No ()

Vinculación con el medio

1) ¿Consideras que las actividades extracurriculares son adecuadas para complementar su formación integral?

Sí ()

No ()

¿Por qué?

2) ¿Consideras que es adecuada la información y orientación para la realización de las prácticas profesionales?

Sí ()

No ()

¿Por qué?

3) ¿Consideras que se presentan las oportunidades para integrarse a proyectos o actividades de vinculación de los estudiantes con el medio académico interno y externo?

Sí ()

No ()

¿Por qué?

Satisfacción General

1) Grado de satisfacción general con respecto a la licenciatura:

Muy bueno	Bueno	Malo	Muy malo
-----------	-------	------	----------



APROBADO
12 AGO 2016
 Comisión Permanente Académica
 Consejo Universitario



2) Comentarios adicionales:

ANEXO A. AREAS DE CONOCIMIENTO – LIS

I	II	III	IV	V	VI	VII	VIII
Cálculo Diferencial MAT	Cálculo Integral MAT	Teoría de la Computación MAT	Arquitectura de Computadoras AC	Redes y Seguridad de Computadoras RED	Programación en la Web PISw	Métodos Formales en Especificación y Diseño de SW PISw	Métricas de Software PISw
Álgebra Superior I MAT	Álgebra Superior II MAT	Álgebra Lineal MAT	Teoría de Lenguajes de Programación PISw	Bases de Datos TI	Sistemas Distribuidos RED	Administración de Proyectos I PISw	Administración de Proyectos II PISw
Fundamentos de Programación PISw	Programación PISw	Estructuras de Datos PISw	Sistemas Operativos SwB	Arquitecturas de Software PISw	Área de Concentración I. A. PISw	Área de Concentración I. B. PISw	Área de Concentración I. C. PISw
Metodología de la Investigación PISw	Matemáticas Discretas MAT	Interacción Humano Computadora IHM	Diseño de Software PISw	Construcción y Evolución de Software PISw	Aseguramiento de la Calidad del Software PISw	Evaluación de Proyectos ES	Área de Concentración II. B. PISw
Fundamentos de Ingeniería de Software PISw	Gestión Tecnológica ES	Probabilidad MAT	Inferencia Estadística MAT	Diseño de Experimentos en Ing-Sw MAT	Desarrollo de Requisitos de Software PISw	Área de Concentración II. A. PISw	Área de Concentración II. C. PISw
						Taller de Servicio Social	Taller de Prácticas Profesionales

ES	Entorno Social	MAT	Matemáticas
AC	Arquitectura de Computadoras	RED	Redes
SwB	Software de Base	PISw	Programación e Ingeniería de Software
TI	Tratamiento de la Información	IHM	Interacción Humano Computadora

APROBADO
12 AGO 2016
Comisión Permanente Académica
Consejo Universitario

