



**UADY**  
FACULTAD DE  
MATEMÁTICAS

**PROPUESTA DE  
MODIFICACIÓN DEL PLAN DE ESTUDIOS  
DE LA LICENCIATURA EN INGENIERÍA DE SOFTWARE**

**Junio de 2009**



# CONTENIDO

<b>CONTENIDO.....</b>	<b>III</b>
<b><u>I. DATOS GENERALES.....</u></b>	<b><u>1</u></b>
NOMBRE DE LA PROPUESTA.....	1
TÍTULO QUE OTORGARÁ.....	1
DEPENDENCIA.....	1
RESPONSABLES DE LA PROPUESTA.....	1
FECHA DE INICIO.....	1
<b><u>II. PRESENTACIÓN.....</u></b>	<b><u>2</u></b>
<b><u>III. JUSTIFICACIÓN.....</u></b>	<b><u>3</u></b>
<b><u>IV. OBJETIVOS DEL PROGRAMA.....</u></b>	<b><u>8</u></b>
GENERAL.....	8
ESPECÍFICOS.....	8
<b><u>V. PERFIL DE INGRESO.....</u></b>	<b><u>9</u></b>
<b><u>VI. PERFIL DE EGRESO.....</u></b>	<b><u>10</u></b>
<b><u>VII. ESTRUCTURA DEL PLAN DE ESTUDIOS.....</u></b>	<b><u>12</u></b>
MODALIDAD DEL PLAN.....	12
ORGANIZACIÓN DEL PLAN.....	13
MAPA CURRICULAR.....	19
ESTRATEGIAS PEDAGÓGICAS GENERALES.....	22
<b><u>VIII. RÉGIMEN ACADÉMICO-ADMINISTRATIVO.....</u></b>	<b><u>24</u></b>
REQUISITOS DE INGRESO.....	24
REQUISITOS DE PERMANENCIA.....	24
REQUISITOS DE EGRESO.....	26
REQUISITOS DE TITULACIÓN.....	26
<b><u>IX. PLAN DE LIQUIDACIÓN.....</u></b>	<b><u>27</u></b>
<b><u>X. RECURSOS HUMANOS Y FÍSICOS.....</u></b>	<b><u>28</u></b>
RECURSOS HUMANOS.....	28
RECURSOS FÍSICOS.....	28
<b><u>XI. MECANISMOS DE EVALUACIÓN CURRICULAR.....</u></b>	<b><u>29</u></b>

SISTEMA DE EVALUACIÓN.....	29
<b>XII. DESCRIPCIÓN SINTÉTICA DE LAS ASIGNATURAS DEL NIVEL BÁSICO.....</b>	<b>30</b>
<b>ÁLGEBRA SUPERIOR I.....</b>	<b>31</b>
<b>ÁLGEBRA SUPERIOR II.....</b>	<b>32</b>
<b>ÁLGEBRA LINEAL.....</b>	<b>33</b>
<b>CÁLCULO DIFERENCIAL.....</b>	<b>34</b>
<b>INFERENCIA ESTADÍSTICA.....</b>	<b>39</b>
<b>EVALUACIÓN DE PROYECTOS.....</b>	<b>43</b>
<b>PROGRAMACIÓN.....</b>	<b>47</b>
<b>XIII. DESCRIPCIÓN SINTÉTICA DE LAS ASIGNATURAS DEL NIVEL DISCIPLINARIO.....</b>	<b>49</b>
<b>METODOLOGÍA DE LA INVESTIGACIÓN.....</b>	<b>50</b>
<b>GESTIÓN TECNOLÓGICA.....</b>	<b>52</b>
<b>ARQUITECTURA DE COMPUTADORAS.....</b>	<b>55</b>
<b>ESTRUCTURAS DE DATOS.....</b>	<b>57</b>
<b>TEORÍA DE LENGUAJES DE PROGRAMACIÓN.....</b>	<b>59</b>
<b>PROGRAMACIÓN EN LA WEB.....</b>	<b>61</b>
<b>INTERACCIÓN HUMANO COMPUTADORA.....</b>	<b>63</b>
<b>FUNDAMENTOS DE INGENIERÍA DE SOFTWARE.....</b>	<b>66</b>
<b>DESARROLLO DE REQUISITOS DE SOFTWARE.....</b>	<b>68</b>
<b>DISEÑO DE SOFTWARE.....</b>	<b>70</b>
<b>UBICACIÓN DEL DISEÑO DENTRO DEL CICLO DE DESARROLLO.....</b>	<b>70</b>
<b>DISEÑO DETALLADO.....</b>	<b>70</b>
<b>DOCUMENTACIÓN DEL DISEÑO.....</b>	<b>70</b>
<b>EVOLUCIÓN DEL DISEÑO.....</b>	<b>70</b>
<b>DISEÑO ORIENTADO A SISTEMAS DISTRIBUIDOS. ....</b>	<b>70</b>
<b>CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE.....</b>	<b>72</b>

<u>ARQUITECTURAS DE SOFTWARE.....</u>	<u>74</u>
<u>ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE.....</u>	<u>76</u>
<u>ADMINISTRACIÓN DE PROYECTOS I.....</u>	<u>80</u>
<u>MÉTRICAS DE SOFTWARE.....</u>	<u>82</u>
<u>SISTEMAS DISTRIBUIDOS.....</u>	<u>86</u>
<u>REDES Y SEGURIDAD DE COMPUTADORAS.....</u>	<u>88</u>
<u>SISTEMAS OPERATIVOS.....</u>	<u>90</u>
<u>XIV. DESCRIPCIÓN SINTÉTICA DE ÁREAS DE CONCENTRACIÓN.....</u>	<u>92</u>

# **I. Datos Generales**

## **Nombre de la Propuesta**

Modificación del Plan de Estudios de la Licenciatura en Ingeniería de Software.

## **Título que Otorgará**

Ingeniero(a) de Software

## **Dependencia**

Facultad de Matemáticas

## **Responsables de la Propuesta**

Dr. Raúl Antonio Aguilar Vera  
L.C.C. Martín Leonel Chi Pérez  
M.C.C. Luis Ramiro Basto Díaz  
M.C.C Edgar Cambranes Martínez  
M.C.C Luis Fernando Curi Quintal

## **Fecha de inicio**

Agosto de 2009

## II. Presentación

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 consideró desde su concepción elementos característicos del nuevo Modelo Educativo y Académico propuesto para nuestra Universidad en septiembre de 2002; cabe mencionar, que dicho programa curricular se corresponde con uno de los cuatro nuevos perfiles de profesionistas en Informática y Computación identificados por la Asociación de Instituciones de Educación en Tecnologías de la Información (ANIEI).

Actualmente la matrícula de la carrera es de 93 alumnos distribuidos en los cuatro semestres (segundo, cuarto, sexto y octavo) que se encuentran en operación, y debido a su reciente creación, en julio próximo egresará la segunda generación de Ingenieros de Software, habiéndose titulado al momento de elaborar la propuesta, cuatro alumnos de la primera generación. Cabe mencionar que en agosto de 2009 se incorporarán a la matrícula los 40 alumnos de nuevo ingreso que ya han sido seleccionados.

Con la idea de incorporar de una manera más adecuada aquellos mecanismos que conduzcan a lograr efectos deseables en los componentes propuestos por el Modelo Educativo y Académico de la UADY —algunos ya inmersos en el plan de estudios de la carrera de Ingeniero de Software— se propone la presente modificación del Plan de Estudios de la Licenciatura en Ingeniería de Software, la cual consiste fundamentalmente en una ***flexibilización del régimen académico-administrativo***.

Se pretende que con dicha modificación los estudiantes puedan avanzar en su formación de acuerdo con sus capacidades y necesidades particulares; con esta modificación, se espera incidir en una mejora en los indicadores de trayectoria escolar reportados hasta el momento por los estudiantes de la carrera, y para ello se propone eliminar restricciones administrativas, características de los planes rígidos, dejando preparado el plan de estudios de la LIS para futuros cambios académicos.

### III. Justificación

La Universidad Autónoma de Yucatán (UADY) en su interés por transformarse para mantener la vigencia y pertinencia de sus funciones sustantivas, pero sobre todo, para refrendar su compromiso social como institución pública, se ha planteado desde septiembre de 2002 un nuevo Modelo Educativo y Académico; sin embargo, la operación de dicho modelo ha implicado enfrentar un conjunto de desafíos que se han venido abordando con la celeridad que sus recursos le permiten, así como también, bajo los riesgos e implicaciones que dichos cambios implican. Entre dichos desafíos podemos destacar:

- ✓ Igualdad de oportunidades de acceso para los estudiantes y apoyo para desarrollar sus capacidades en condiciones adecuadas para su desempeño óptimo.
- ✓ Aseguramiento de la calidad de los procesos académicos, centrados en el aprendizaje de los estudiantes.
- ✓ Establecimiento de acuerdos y redes eficaces de cooperación que favorezcan la movilidad de estudiantes y profesores.

Por su parte, la Facultad de Matemáticas en su compromiso por lograr la excelencia académica —de manera acorde con las políticas institucionales— ha venido implementando de manera gradual, algunas estrategias internas y otras de carácter institucional para incorporar las características propuestas en el Modelo Educativo y Académico; en particular, para afrontar el primero de los desafíos antes citados, ha venido utilizando como principal instrumento de selección de aspirantes a ingresar a sus programas curriculares en el nivel de Licenciatura, el examen nacional de ingreso a licenciatura administrado por el Centro Nacional de Evaluación para la Educación Superior (CENEVAL).

Con base en los resultados obtenidos en los últimos años, hoy día se puede afirmar que el uso del examen del CENEVAL ha impactado de manera positiva en el singular interés que ha mostrado la UADY en **proporcionar espacios y oportunidades por igual a todos aquellos estudiantes interesados en cursar una licenciatura**; ello se ve reflejado en la diversidad de subsistemas de nivel medio superior de donde provienen actualmente los alumnos que ingresan a la licenciatura; los datos reportados indican que solo el 20% de los alumnos que ingresan a una Licenciatura provienen de las preparatorias de la UADY.



A la par con el impacto positivo —desde el punto de vista institucional— que ha tenido la estrategia adoptada para la selección de alumnos de nuevo ingreso, en la Facultad de Matemáticas se ha observado en los resultados de la prueba diagnóstica que se aplica a los aspirantes aceptados, que éstos no cuentan con todos los conocimientos del perfil de ingreso (de las tres últimas generaciones que han ingresado a la facultad, más del 80%) y a pesar de haberse implementado como estrategia de apoyo un conjunto de cursos de nivelación intensivo con duración de cuatro semanas previas al inicio del semestre, estos han resultado insuficientes para lograr promover el nivel requerido de conocimientos, tal y como lo confirman los bajos índices de aprobación en algunas de las asignaturas de los primeros semestres.

Para ilustrar lo anterior, las tablas 1, 2, 3 y 4 presentan los índices de aprobación reportados en el último curso escolar de la carrera de Ingeniero de Software —agosto de 2008 a enero de 2009— para las asignaturas obligatorias (básicas y disciplinarias) del plan de estudios.

Tabla 1. Primer Semestre

Asignatura	Aprobación
Álgebra Superior I	48%
Cálculo Diferencial	59%
Fundamentos de Ingeniería de Software	53%
Fundamentos de Programación	53%
Metodología de la Investigación	90%

Tabla 2. Tercer Semestre

Asignatura	Aprobación
Álgebra Lineal	50%
Estructuras de Datos	76%
Interacción Humano-Computadora	86%
Probabilidad	44%
Teoría de la Computación	48%

Tabla 3. Quinto Semestre

<b>Asignatura</b>	<b>Aprobación</b>
Bases de Datos	100%
Arquitecturas Software	100%
Bases de Datos	100%
Construcción y Evolución SW	100%
Diseño de Experimentos en IS	78%
Redes y Seguridad de Computadoras	100%

Tabla 4. Séptimo Semestre

<b>Asignatura</b>	<b>Aprobación</b>
Administración de Proyectos I	88%
Evaluación de Proyectos	88%
Métodos Formales en Especificación y Diseño de SW	100%

Para la UADY —de acuerdo con el Modelo Educativo y Académico— la innovación y la flexibilidad representan los dos componentes orientadores del Modelo con los que es posible adoptar mecanismos apropiados para el funcionamiento deseable de los programas de formación; la flexibilidad, en particular, puede ser definida como el conjunto de características que se incorporan al Modelo, dotándolo de cualidades que le permiten acudir al encuentro de las necesidades de todos los involucrados; impulsa la movilidad de los actores universitarios en la generación y socialización del conocimiento a través del diseño y rediseño de planes de estudios, la formación interdisciplinaria, la promoción del auto aprendizaje, la corresponsabilidad en la toma de decisiones, la consideración de cuestiones contextuales, la planificación y crecimiento de la institución y la diversificación de las opciones en la formación profesional.

Con base en dichas premisas, el objetivo principal del grupo que ha elaborado la propuesta de modificación del plan de estudios, consistió en realizar un análisis de las características de flexibilidad curricular que podrían ser incorporadas al plan de estudios de la Licenciatura en Ingeniería de Software con el fin de que se favorezca la participación de los propios estudiantes —orientados por sus tutores— en las decisiones que los afectan: la elección de sus objetos de estudio, la definición del ritmo de su avance en el programa y la opción por los espacios que les sugieran sus intereses y preferencias para especializarse en su área.

El plan de estudios vigente ha incorporado desde su concepción algunos de los elementos de los modelos curriculares semi-flexibles, su estructura se encuentra integrada por cuarenta asignaturas y dos talleres —prácticas profesionales y servicio social— todos ellos distribuidos en ocho semestres; las asignaturas se encuentran organizadas en tres niveles: básico (13), disciplinario (21) y de especialidad (6), siendo obligatorias las asignaturas de los dos primeros niveles, y de libre elección las del área de especialidad.

El principal elemento de flexibilidad inmerso en el plan 2004 lo ofrecen las asignaturas de especialidad, dichas asignaturas representan el 15% del total de las asignaturas del plan de estudios; su inclusión en el currículo tiene como finalidad completar la formación profesional de un Ingeniero de Software en al menos un área de especialidad, al ser de libre elección, se le brinda al alumno la oportunidad de seleccionar asignaturas que vayan acorde con sus intereses profesionales.

Actualmente el régimen administrativo que rige la operación de todos los planes de estudio de la Facultad obliga al estudiante a mantener una inscripción semestral, en la cual, la carga académica está predeterminada por el conjunto de asignaturas propias del semestre al cual se inscriben. Por otro lado, el avance de un alumno en su formación está regido por la condición de que se podrán inscribir a un semestre si aprobaron al menos el 50% de las asignaturas del semestre inmediato anterior y el 100% del segundo anterior.

Esta condición, junto con los bajos índices de aprobación que se observan en los primeros semestres, han provocado elevados índices de rezago, deserción e incluso de retraso ocioso al cursar el programa, ya que las asignaturas son ofrecidas en períodos anuales, lo que provoca que en ocasiones los estudiantes dejen pasar un semestre sin posibilidad de inscripción, esperando el semestre en el cual la asignatura será ofertada nuevamente.

Ante esta situación, y con el deseo de ofrecer a los estudiantes mejores oportunidades curriculares que les permitan ajustar sus cargas académicas y administrar su avance de acuerdo a sus capacidades y disponibilidad de tiempo, se realizaron los trabajos para la elaboración de la presente modificación del plan de estudios de la Licenciatura en Ingeniería de Software, en la que se propone una versión curricular flexible —en su régimen académico-administrativo— para el plan de estudios vigente.

Con esta visión, la carrera de Ingeniero de Software no presenta cambio alguno en cuanto a los objetivos curriculares, perfiles de ingreso o de egreso, asignaturas obligatorias y/o talleres de apoyo, créditos asociados a las actividades curriculares, requisitos de idioma, opciones de titulación. Por el contrario, el esquema propuesto reduce las restricciones administrativas para el avance de los alumnos a lo largo de su formación, refuerza la importancia del programa de tutorías, ofrece mejores condiciones para el programa de movilidad estudiantil, y establece un conjunto de recomendaciones —en función de los requisitos académicos— que deben de ser consideradas por el alumno al momento de inscribirse a cada una de las 34 asignaturas obligatorias de su plan de estudios. Adicionalmente, se realizaron ligeros ajustes en la cantidad de horas teóricas, prácticas y totales de las asignaturas para considerar actividades que se realizan en ellas y que completan el número horas para los créditos originalmente asignados.

## IV. Objetivos del Programa

### 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

El objetivo general del programa curricular puede ser trasladado a una serie de objetivos específicos, de acuerdo con los cuales, se espera que el egresado de la carrera de Ingeniero de Software:

1. 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.
2. Describa y aplique las herramientas necesarias para la especificación, diseño, verificación y validación de sistemas de software.
3. Se desempeñe en al menos un área de concentración, trabaje y se comunique de forma profesional en equipos interdisciplinarios.
4. Aplique el conocimiento y las habilidades para mejorar el proceso de desarrollo de software.
5. Contribuya al avance de la Ingeniería de Software con un acervo de conocimientos tanto teóricos como prácticos.

## **V. Perfil de ingreso**

El aspirante a ingresar al programa de la Licenciatura en Ingeniería de Software debe poseer:

Conocimientos sobre:

1. La operación básica de una computadora y de al menos una aplicación de oficina.
2. Conceptos básicos de geometría y trigonometría.
3. Conceptos básicos de operaciones algebraicas y ecuaciones.
4. Conceptos básicos de pre cálculo.
5. Nivel básico del idioma inglés.
6. El entorno político, económico y social en la actualidad.

Habilidades para:

1. Tener hábitos y métodos adecuados para el estudio.
2. Expresarse correctamente en forma oral y escrita en la descripción de los procesos que llevan a la solución de un problema dado.
3. Analizar y sintetizar problemas prácticos y de la vida real.

Y es deseable que posea las siguientes:

Actitudes:

1. Interés y gusto por la aplicación de las matemáticas y la computación.
2. Interés y gusto por la investigación tanto teórica como práctica.
3. Interés y gusto para proponer e implementar soluciones.
4. Disposición para el trabajo académico, en forma cooperativa y participativa, dentro y fuera del aula de clases.

## **VI. Perfil de egreso**

Al concluir la Licenciatura en Ingeniería de Software el egresado tendrá:

Conocimientos sobre:

1. 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.
2. Las ciencias computacionales, que le permitan la aplicación viable de algoritmos, estructuras de datos y lenguajes de programación en el desarrollo de software.
3. 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.
4. Administración y optimización de tiempo y costo de los recursos involucrados en el desarrollo de software.
5. El entorno social para describir los factores sociales, políticos, ecológicos y éticos relacionados con el desarrollo tecnológico y el desempeño profesional.
6. Las áreas de investigación relacionadas con la Ingeniería de Software.

Habilidades para:

1. Trabajar como parte de un equipo en el desarrollo y evolución de productos de software.
2. Explicar el proceso para determinar las necesidades del cliente y traducirlos a requisitos de software.
3. Reconciliar objetivos en conflicto, considerando compromisos con las limitaciones de costo, tiempo, conocimiento, sistemas existentes y de las organizaciones involucradas.
4. Diseñar soluciones apropiadas en una o más áreas de concentración, usando enfoques de ingeniería que integren temas éticos, sociales, legales y económicos.
5. Entender y aplicar teorías, modelos y técnicas que provean una base para el diseño, desarrollo, verificación e implantación del software.
6. Negociar y trabajar profesionalmente, liderar cuando sea necesario y comunicarse efectivamente con los interesados en un ambiente de Ingeniería de Software.

Y es deseable que posea

Actitudes de:

1. Liderazgo en equipos de trabajo multidisciplinarios.
2. Perseverancia en la solución de problemas.
3. Capacidad de mantenerse actualizado en su área de trabajo.
4. Afán de superación.
5. Responsabilidad y ética en su desempeño profesional.
6. Conducta emprendedora e innovadora.
7. Conciencia de la realidad social y responsabilidad ecológica.



## VII. Estructura del plan de Estudios

### Modalidad del plan

#### Tipo de Plan:

Los contenidos del plan de estudios se encuentran organizados en asignaturas (34 obligatorias y al menos 6 de libre elección) y talleres de apoyo (prácticas profesionales y servicio social).

#### Duración máxima:

La permanencia del estudiante, será como máximo de 16 períodos semestrales, contabilizados a partir de su primer ingreso al programa educativo. El tiempo recomendable para cursarlo es de 8 períodos semestrales.

#### Periodicidad de ingreso:

Anual.

#### Características:

1. La **inscripción** al plan de estudios será realizada semestralmente, siendo responsabilidad del alumno —con el apoyo de un tutor— la elección de las asignaturas y/o talleres a cursar en cada semestre, lo anterior, de acuerdo a un mínimo de asignaturas previamente establecido (ver requisitos de permanencia).
2. El plan de estudios se organiza en tres niveles: **básico**, donde se promoverá el desarrollo de las habilidades intelectuales básicas y técnicas indispensables para la formación profesional; **disciplinario**, donde se adquirirán los conocimientos y habilidades necesarias relativas al área de las ciencias computacionales y la Ingeniería de Software; y **de especialidad**, donde adquirirán las competencias necesarias en, al menos un área de concentración, promoviendo la adquisición de conocimientos y habilidades.

3. Los niveles básico y disciplinario estarán integrados por treinta y cuatro asignaturas obligatorias consideradas fundamentales y que han sido definidos en función de los objetivos educativos y curriculares, que se vinculan estrechamente con el logro de ellos.
4. El nivel de especialidad, estará integrado por al menos seis asignaturas de libre elección que complementen la formación profesional, apoyen las asignaturas obligatorias, brinden posibilidades de orientación y refuercen el énfasis de interés y especialización. Cada área de concentración deberá estar integrado por un mínimo tres asignaturas. Se podrán elegir dos áreas de concentración pudiendo repetir el área de concentración con diferentes asignaturas.

## Organización del plan

### Asignaturas Obligatorias

Las treinta y cuatro asignaturas obligatorias que conforman los bloques **básico** y **disciplinario** se listan en la Tablas 5 y 6; para cada asignatura se indica las horas teóricas y prácticas que serán requeridas para su impartición, y en consecuencia, sus créditos. De acuerdo al período lectivo establecido por la institución, se consideran **16 semanas** por semestre.

Tabla 5. Relación de las once asignaturas del bloque básico

Clave	Asignatura	Horas		Total de horas	Créditos
		Teóricas	Prácticas		
MT-01	Álgebra Superior I	75	0	75	10
MT-02	Álgebra Superior II	75	0	75	10
MT-03	Álgebra Lineal	75	0	75	10
MT-04	Cálculo Diferencial	105	15	120	15
MT-05	Cálculo Integral	105	15	120	15
MT-08	Probabilidad	75	0	75	10
MT-09	Inferencia Estadística	75	0	75	10
MT-12	Matemáticas Discretas	60	15	75	9
MT-20	Evaluación de Proyectos	60	15	75	9
AP-01	Fundamentos de Programación	45	30	75	8

AP-02	Programación	45	30	75	8
<b>Totales</b>		<b>795</b>	<b>120</b>	<b>915</b>	<b>114</b>

Tabla 6. Relación de las veintitrés asignaturas del bloque disciplinario

Clave	Asignatura	Horas		Total de horas	Créditos
		Teóricas	Prácticas		
ES-02	Metodología de la Investigación	75	0	75	10
ES-03	Gestión Tecnológica	75	0	75	10
MT-13	Teoría de la Computación	75	0	75	10
AC-01	Arquitectura de Computadoras	60	15	75	9
AP-03	Estructuras de Datos	45	30	75	8
AP-04	Teoría de Lenguajes de Programación	45	30	75	8
AP-07	Programación en la WEB	45	30	75	8
IH-05	Interacción Humano-Computadora	45	30	75	8
SI-01	Bases de Datos	75	0	75	10
IS-01	Fundamentos de Ingeniería de Software	60	15	75	9
IS-02	Desarrollo de Requisitos Software	45	30	75	8
IS-03	Diseño de Software	45	30	75	8
IS-05	Construcción y Evolución del Software	45	30	75	8
IS-06	Arquitecturas de Software	45	30	75	8
IS-07	Aseguramiento de la Calidad del Software	45	30	75	8
IS-08	Métodos Formales en Especificación y Diseño de Software	60	15	75	9
IS-09	Administración de Proyectos I	45	30	75	8
IS-10	Métricas de Software	75	0	75	10
IS-11	Administración de Proyectos II	15	60	75	6
IS-12	Diseño de Experimentos en Ingeniería de Software	60	15	75	9
RE-02	Sistemas Distribuidos	60	15	75	9
RE-03	Redes y Seguridad de Computadoras	60	15	75	9
SB-03	Sistemas Operativos	75	0	75	10
<b>Totales</b>		<b>1275</b>	<b>450</b>	<b>1725</b>	<b>200</b>

Las tablas 7 y 8 presentan la relación de conocimientos y habilidades respectivamente, con respecto a las asignaturas obligatorias y talleres considerados en el plan de estudios.

Tabla 7. Relación de conocimientos del perfil de egreso con las asignaturas obligatorias

Conocimientos	Asignaturas
1. 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.	Fundamentos de Programación, Fundamentos de Ingeniería de Software, Programación, Estructura de Datos, Interacción Humano Computadora, Diseño de Software, Arquitecturas de Software, Construcción y Evolución de Software, Programación en la Web, Aseguramiento de la Calidad del Software, Desarrollo de Requisitos de Software, Métodos Formales en Especificación y Diseño de Software, Administración de Proyectos I, Evaluación de Proyectos, Métricas de Software, Administración de Proyectos II.
2. Las ciencias computacionales, que le permitan la aplicación viable de algoritmos, estructuras de datos y lenguajes de programación en el desarrollo de software.	Cálculo Diferencial, Álgebra Superior, Fundamentos de Programación, Cálculo Integral, Álgebra Superior II, Programación, Matemáticas Discretas, Teoría de la Computación, Álgebra Lineal, Estructura de Datos, Arquitectura de Computadoras, Teoría de Lenguajes de Programación, Sistemas Operativos, Bases de Datos, Programación en la Web, Sistemas Distribuidos.
3. 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.	Fundamentos de Programación, Diseño de Software, Redes y Seguridad de Computadoras, Bases de Datos, Construcción y Evolución de Software, Aseguramiento de la Calidad del Software, Métodos Formales en Especificación y Diseño de Software, Administración de Proyectos I, Evaluación de Proyectos, Métricas de Software, Administración de Proyectos II.
4. Administración y optimización de tiempo y costo de los recursos involucrados en el desarrollo de software.	Probabilidad, Inferencia Estadística, Diseño de Experimentos en Ingeniería de Software, Desarrollo de Requisitos de Software, Administración de Proyectos I, Administración de Proyectos II.
5. El entorno social para describir los factores sociales, políticos, ecológicos y éticos relacionados con el desarrollo tecnológico y el desempeño profesional.	Metodología de la Investigación, Fundamentos de Ingeniería de Software, Gestión Tecnológica, Administración de Proyectos I, Administración de Proyectos II, Taller de Servicio Social, Taller de Prácticas Profesionales.
6. Las áreas de investigación relacionadas con la Ingeniería de Software.	Metodología de la Investigación, Fundamentos de Ingeniería de Software, Diseño de Software, Construcción y Evolución de Software, Diseño de Experimentos en Ingeniería de Software, Aseguramiento de la Calidad del Software, Desarrollo de Requisitos de Software, Métodos Formales en Especificación y Diseño de Software, Métricas de Software, Áreas de Concentración (las seis), Taller de Servicio Social, Taller de Prácticas Profesionales.

Tabla 8. Relación de habilidades del perfil de egreso con las asignaturas obligatorias

Habilidades	Asignaturas
1.Trabajar como parte de un equipo en el desarrollo y evolución de productos de software.	Fundamentos de Programación, Fundamentos de Ingeniería de Software, Programación, Estructura de Datos, Diseño de Software, Arquitecturas de Software, Construcción y Evolución de Software, Aseguramiento de la Calidad del Software, Desarrollo de Requisitos de Software, Métodos Formales en Especificación y Diseño de Software, Métricas de Software, Administración de Proyectos I, Administración de Proyectos II, Evaluación de Proyectos, Taller de Servicio Social, Taller de Prácticas Profesionales.
2.Explicar el proceso para determinar las necesidades del cliente y traducirlos a requisitos de software.	Fundamentos de Ingeniería de Software, Aseguramiento de la Calidad del Software, Desarrollo de Requisitos de Software.
3.Reconciliar objetivos en conflicto, considerando compromisos con las limitaciones de costo, tiempo, conocimiento, sistemas existentes y de las organizaciones involucradas.	Diseño de Software, Arquitecturas de Software, Construcción y Evolución de Software, Desarrollo de Requisitos de Software, Evaluación de Proyectos, Administración de Proyectos II.
4.Diseñar soluciones apropiadas en una o más áreas de concentración, usando enfoques de ingeniería que integren temas éticos, sociales, legales y económicos.	Metodología de la Investigación, Fundamentos de Ingeniería de Software, Probabilidad, Inferencia Estadística, Métodos Formales en Especificación y Diseño de Software, Métricas de Software, Áreas de Concentración (las seis materias).
5. Entender y aplicar teorías, modelos y técnicas que provean una base para el diseño, desarrollo, verificación e implantación del software.	Cálculo Diferencial, Álgebra Superior I, Fundamentos de Programación, Metodología de la Investigación, Fundamentos de Ingeniería de Software, Cálculo Integral, Álgebra Superior II, Programación, Matemáticas Discretas, Teoría de la Computación, Álgebra Lineal, Estructura de Datos, Interacción Humano Computadora, Arquitectura de Computadoras, Teoría de Lenguajes de Programación, Sistemas Operativos, Diseño de Software, Redes y Seguridad de Computadoras, Bases de Datos, Arquitecturas de Software, Construcción y Evolución de Software, Diseño de Experimentos en Ingeniería de Software, Programación en la Web, Sistemas Distribuidos, Aseguramiento de la Calidad del Software, Métodos Formales en Especificación y Diseño de Software, Administración de Proyectos I, Métricas de Software, Administración de Proyectos II.
6.Negociar y trabajar profesionalmente, liderar cuando sea necesario y comunicarse efectivamente con los interesados en un ambiente de Ingeniería de Software.	Fundamentos de Ingeniería de Software, Desarrollo de Requisitos de Software, Evaluación de Proyectos, Administración de Proyectos II.

### Asignaturas de Libre elección

En cuanto a las asignaturas de **especialidad**, estas podrán comenzar a cursarse una vez aprobados el 70% de los créditos de las asignaturas

obligatorias —220 créditos aproximadamente— teniendo un valor de entre 7 y 10 créditos cada una.

Debido a la flexibilidad considerada para completar la formación del Ingeniero de Software mediante las asignaturas de especialidad, el alumno podrá cursar las asignaturas de las áreas de concentración ofrecidas por la dependencia, 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 el coordinador de la carrera en Ingeniería de Software.

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 otras asignaturas de especialidad —que integren un área de concentración— como es el caso de las áreas de concentración denominadas: **Programación WEB, VideoJuegos, Arquitecturas de SW Distribuidas, e Informática Educativa.**

#### Talleres de Apoyo a la Formación

Como parte del proceso de formación integral del Ingeniero de Software, el plan de estudios contempla dos talleres de apoyo a la formación: Taller de Prácticas Profesionales y Taller de Servicio Social.

**Taller de Prácticas Profesionales:** Las prácticas profesionales podrán acreditarse luego de haber aprobado 150 créditos de las asignaturas obligatorias del plan de estudios; se considera que en ese momento ya contarán con los conocimientos y habilidades para poder desarrollarse al menos como programadores.

Las prácticas profesionales se acreditarán a través del Taller de Prácticas Profesionales con valor curricular de *ocho créditos*. El propósito del taller es que el alumno realice una práctica profesional efectiva de su carrera. La calificación podrá ser de Aprobado o No Aprobado. Para aprobarlo, el alumno deberá realizar al menos *320 horas* de práctica que se acreditarán a través de un plan de trabajo inicial y los informes de actividades acordados en el plan, ambos deberán ser avalados por un tutor de la Facultad y por el responsable del programa en la Institución donde se realice la práctica.

Las horas prácticas se podrán realizar en las diferentes dependencias de la UADY o en cualquier otra institución, empresa u organización, previa solicitud y aprobación de la Secretaría Académica. Es deseable que el

alumno realice sus prácticas profesionales en el transcurso de un solo semestre, sin embargo, se pueden considerar otras opciones sujetas a la aprobación de la Secretaría Académica y del Coordinador de la carrera.

**Taller de Servicio Social:** De acuerdo con el Reglamento de Servicio Social de la Universidad, éste puede iniciarse al acreditarse el 70% de los créditos del plan de estudios, es decir, luego de aprobar 263 créditos.

En el plan de estudios de la carrera de Ingeniero de Software, el servicio social se acreditará en el marco de un taller con valor curricular de *doce créditos*, y tendrá una duración mínima de *480 horas*. La calificación podrá ser de Aprobado o No Aprobado. El propósito del taller es que el alumno, a través de un programa bien definido, y haciendo uso de los conocimientos, habilidades y actitudes propios de su perfil profesional, retribuya a la sociedad parte de lo que ésta le ha proporcionado durante su preparación académica.

#### Resumen de créditos y horas del programa

En consecuencia, la Licenciatura en Ingeniería de Software requiere para su finalización un tiempo estimado de:

Totales de Horas			
Total de horas de asignaturas obligatorias			2,640
Total de horas de asignaturas de especialidad			
	Mínimo	60 X 6 = 360	
	Máximo	75 X 6 = 450	
Total de horas del taller de Prácticas Profesionales			320
Total de horas del taller de Servicio Social			480
Total de horas del Plan de Estudios			
	Mínimo		3,800
	Máximo		3,890

y cuenta con un valor en créditos de:

Totales de créditos			
Total de créditos de asignaturas obligatorias			314
Total de créditos de asignaturas de especialidad			
	Mínimo	7 X 6 = 42	
	Máximo	10 X 6 = 60	
Total de créditos del taller de Prácticas Profesionales			8
Total de créditos del taller de Servicio Social			12
Total de créditos del Plan de Estudios			
	Mínimo		376

## Mapa Curricular

Debido a las características de flexibilidad del plan de estudios, cada estudiante podrá configurar de manera individualizada el conjunto de asignaturas a cursar en cada semestre, de tal manera que el mapa curricular de la Licenciatura en Ingeniería de Software se reduce a un listado de asignaturas, las cuales, por restricciones de disponibilidad en cuanto a recursos físicos y humanos, éstas serán distribuidas en los dos períodos semestrales del curso escolar. La Tabla 9 presenta los períodos en que serán ofrecidas las asignaturas y talleres del plan de estudios.

Tabla 9. Oferta de asignaturas y talleres del plan de estudios (LIS)

Agosto-Enero	Enero-Junio
Cálculo Diferencial, Cálculo Integral, Álgebra Superior I, Álgebra Superior II, Fundamentos de Programación, Programación, Metodología de la Investigación, Fundamentos de Ingeniería de Software, Teoría de la Computación, Álgebra lineal, Estructuras de Datos, Interacción Humano-Computadora, Probabilidad, Redes y Seguridad de Computadoras, Bases de Datos, Arquitecturas Software, Construcción y Evolución del Software, Diseño de Experimentos en Ingeniería de Software, Métodos Formales en Especificación y Diseño de Software, Administración de Proyectos I, Evaluación de Proyectos, Taller de Prácticas Profesionales, Taller de Servicio Social, Asignatura del área de Concentración A, Asignatura del área de Concentración B.	Cálculo Diferencial, Cálculo Integral, Álgebra Superior I, Álgebra Superior II, Fundamentos de Programación, Programación, Matemáticas Discretas, Gestión Tecnológica, Arquitectura de Computadoras, Teoría de lenguajes de Programación, Sistemas Operativos, Diseño de Software, Inferencia Estadística, Programación en la WEB, Sistemas Distribuidos, Aseguramiento de la calidad del Software, Desarrollo de Requisitos Software, Métricas de Software, Administración de proyectos II, Taller de Prácticas Profesionales, Taller de Servicio Social, Asignatura del área de Concentración A, Asignatura del área de Concentración B.

Cabe mencionar que el listado de asignaturas en cada uno de los dos períodos semestrales podrá ampliarse en función de la demanda de



asignaturas y de la disponibilidad de recursos físicos y humanos con que cuente la Facultad en cada período escolar.

Con el propósito de servir de guía a cada estudiante para el diseño del mapa curricular, se ha elaborado una tabla (ver tablas 10a y 10b) de requisitos académicos de cada una de las asignaturas del plan de estudios, dicha guía pretende asistir tanto al alumno como al tutor a decidir la carga de asignaturas a cursar por semestre, y considera sugerencias académicas en cuanto a los requisitos de asignaturas y/o temas que se recomienda debería saber el alumno al inscribir cada asignatura o conjunto de asignaturas.

Tabla 10a. Requisitos académicos para las asignaturas obligatorias (LIS)

Asignaturas	Requisitos Académicos	
	Asignatura	Contenido
Cálculo Diferencial	Perfil de ingreso	
Álgebra Superior I	Perfil de ingreso	
Fundamentos de Programación	Perfil de ingreso	
Metodología de la Investigación	Perfil de ingreso	
Fundamentos de Ing. de Software	Perfil de ingreso	
Cálculo Integral	Cálculo Diferencial	Graficación de funciones Límite (el concepto y la parte mecánica) Derivada (la parte mecánica)
Álgebra Superior II	Álgebra Superior I	Lógica, métodos de demostración Cuantificadores universales
Programación	Fundamentos de Programación	Lógica de programación
Matemáticas Discretas	Álgebra Superior I	Lógica, métodos de demostración Relaciones y funciones
Gestión Tecnológica	No tiene	
Teoría de la Computación	Álgebra Superior I	Lógica, métodos de demostración y conjuntos Relaciones y funciones
	Matemáticas Discretas	Teoría de grafos Teoría de árboles
Álgebra Lineal	Álgebra Superior II	Números complejos Polinomios Matrices
Estructuras de Datos	Programación	Todos los temas
Interacción Humano-Computadora	Fundamentos de Ingeniería de Software	Todo el contenido
Probabilidad	Cálculo Diferencial	Todo el contenido
Arquitectura de Computadoras		
Teoría de Lenguajes de Programación	Teoría de la computación	Lenguajes y expresiones regulares Gramáticas y lenguajes libres de contexto
	Estructuras de datos	Árboles y listas
Sistemas Operativos	Estructuras de datos	Ordenamiento y búsqueda

Asignaturas	Requisitos Académicos	
	Asignatura	Contenido
		Interrupciones Multiprocesamiento

Tabla 10b. Requisitos académicos para las asignaturas obligatorias (LIS)

Asignaturas	Requisitos Académicos	
	Asignatura	Contenido
Diseño de Software	Estructuras de Datos Fundamentos de Ingeniería de Software	Todo el contenido
Inferencia Estadística	Probabilidad	Todo el contenido
Redes y Seguridad de Computadoras	Programación	
Bases de Datos	Álgebra Superior I Estructura de Datos	Relaciones y funciones
Arquitecturas de Software	Diseño de Software	Todo el contenido
Construcción y Evolución de Software	Diseño de Software	Todo el contenido
Diseño de Experimentos en Ingeniería de Software	Inferencia Estadística Fundamentos de IS	Todo el contenido
Programación en la web	Bases de Datos	Sistemas Manejadores de Bases de Datos
Sistemas Distribuidos	Redes y Seguridad de Computadoras	Seguridad en Equipos de Interconexión (Modelo de Referencia TCP/IP).
Aseguramiento de la Calidad del Software	Construcción y Evolución de Software	Todo el contenido
Desarrollo de Requisitos de Software	Fundamentos de Ingeniería de Software	Todo el contenido
Métodos Formales en Especificación y Diseño de Software	Álgebra Superior I Diseño de Software Desarrollo de requisitos Software	Relaciones y funciones
Administración de Proyectos I	Diseño de Software	Todo el contenido
Evaluación de Proyectos	Desarrollo de requisitos Software	Todo el contenido
Métricas de Software	Inferencia Estadística	Todo el contenido
Administración de Proyectos II	Administración de Proyectos I	Todo el contenido

En el caso de las asignaturas de especialidad, los requisitos académicos deberán estar definidos en la descripción de la propuesta del área de concentración.

## **Estrategias Pedagógicas Generales**

En general, la metodología de enseñanza en las clases será mediante exposiciones por parte de profesores y estudiantes, prácticas individuales, y proyectos en equipo. Los conocimientos y habilidades adquiridos por los estudiantes se consolidarán mediante trabajos extra clase a los que se les ha asignado un peso adecuado en los criterios de evaluación de las asignaturas. En los programas detallados de cada asignatura se presentará la metodología correspondiente.

Las horas prácticas que se mencionan en los programas sintéticos de las asignaturas del plan se realizan bajo la supervisión del instructor; no obstante, es importante hacer énfasis que en la Facultad de Matemáticas es común, dadas las características de los programas curriculares del área de computación, que parte de la práctica de muchas de las asignaturas se desarrolle fuera de los salones de clase y sin la presencia de los instructores, para ello, las estrategias de supervisión más utilizadas son: sesiones presenciales de asesoría grupal, o el seguimiento de dinámicas de aprendizaje —individuales o por equipo— con asistencia del Sistema de Educación en Línea (SEL).

### **Trabajo en Equipo**

Para el trabajo en equipo se recomienda el uso de salas de trabajo grupal, es decir, espacios físicos acondicionados con una mesa redonda para discusión, pizarrón, equipo de cómputo, vídeo-proyector. Estas aulas se hacen lo más parecido al campo de trabajo y en donde el estudiante podrá desarrollar las habilidades necesarias para el trabajo en equipo simulando la situación real de la Ingeniería de Software siguiendo lo aprendido en clase.

### **Programa de Tutorías**

Para la UADY las tutorías son consideradas como un proceso de acompañamiento y orientación de tipo personal y académico a lo largo del proceso formativo del estudiante para mejorar su rendimiento académico, solucionar problemas escolares, desarrollar hábitos de estudio, de trabajo, de reflexión y de convivencia social.

En la Facultad de Matemáticas, el programa de tutorías comenzó a ofrecerse en febrero de 2003, poco antes de ingresar la primera generación

de la carrera de Ingeniero de software.

El objetivo de las tutorías dentro de la Facultad de Matemáticas es “Contribuir a elevar la calidad del proceso formativo en el ámbito de la construcción de valores, actitudes y hábitos positivos con la promoción del desarrollo de habilidades intelectuales en los estudiantes, mediante la utilización de estrategias de atención personalizada que complementen las actividades docentes regulares, con el fin de abatir la deserción, el rezago y el fracaso escolar permitiendo así el cumplimiento de la misión de la institución”.

Con dicho programa, es el profesor quien asume de manera individual la guía del proceso formativo del estudiante y está permanentemente ligado a las actividades académicas de los alumnos bajo su tutela, orientando, asesorando y acompañando al mismo durante el proceso educativo con la intención de conducirlo hacia su formación integral, estimulando su responsabilidad por aprender y alcanzar sus metas educativas.

El programa de tutorías adquiere una importancia significativa para el modelo de plan semi-flexible, ya que en esta modalidad, será el tutor quién oriente al estudiante en la configuración, y en su caso, adecuación del mapa curricular a seguir para su formación. Cabe destacar que la responsabilidad de la selección e inscripción de asignaturas será del alumno, y en este sentido, el tutor fungirá sólo como orientador.

La asignación de los tutores será hecha por el Secretario Académico —al ingreso del alumno a la carrera— a propuesta del Comité de Tutorías, no obstante, el alumno podrá renunciar a ser tutorado por algún profesor. Para dar seguimiento a un alumno en su proceso formativo, el tutor mantendrá un expediente por cada tutorado, dicho expediente deberá ser actualizado con la información recabada en las sesiones con el estudiante durante el semestre, y con la información de su trayectoria escolar proporcionada por el departamento de control escolar al final de cada curso.

### Movilidad Estudiantil

Los estudiantes podrán cursar asignaturas de otros programas curriculares de la UADY y, a partir de haber aprobado 150 créditos, asignaturas de programas curriculares de otras instituciones de educación superior nacionales o extranjeras reconocidas, previa autorización de la Secretaría Académica de la Facultad. En el caso de asignaturas obligatorias

se reconocerá el mismo número de créditos que establece este plan de estudios y, para el caso de las asignaturas de especialidad, el valor en créditos a reconocer del plan de estudios de la IES receptora, estará en el rango de 7 a 10 créditos para cada asignatura.

## **VIII. Régimen Académico-Administrativo**

Los requisitos de ingreso, permanencia, egreso y titulación de los alumnos de la Licenciatura en Ingeniería de Software, se sustentan en los reglamentos que rigen la normatividad vinculada con los programas curriculares en el nivel de licenciatura de la Universidad Autónoma de Yucatán, así como en el Reglamento Interior de la Facultad de Matemáticas.

### **Requisitos de Ingreso**

Para ingresar a la Licenciatura en Ingeniería de Software se requiere que el aspirante participe en el proceso de admisión a nivel de licenciatura, de acuerdo a la convocatoria respectiva aprobada por el Consejo Universitario. Una vez que el aspirante ha sido admitido, deberá presentar una evaluación diagnóstica, cuyo resultado se tomará en cuenta para recomendar los cursos de nivelación que se consideren necesarios, a propuesta de la Secretaría Académica de la Facultad de Matemáticas.

Al inicio de las actividades académicas de los estudiantes de primer ingreso, la Secretaría Académica organizará un taller de inducción, en donde se darán a conocer la misión institucional, los diversos reglamentos de la Universidad Autónoma de Yucatán y de la Facultad de Matemáticas, la estructura del plan de estudios, el régimen académico-administrativo vigente para su administración, el sistema de tutorías, así como los procedimientos para utilizar los servicios de cómputo, bibliotecarios y escolares.

### **Requisitos de Permanencia**

Todo alumno que haya cubierto los requisitos de ingreso, estará en posibilidad de inscribirse al primer período del mapa curricular individualizado que haya sido configurado con asistencia de su tutor.

Para concluir el plan de estudios en el tiempo recomendable de 8 semestres, el alumno deberá elegir al menos 47 créditos en cada inscripción. En caso de seleccionar cargas semestrales menores, el estudiante deberá tomar en cuenta que en los periodos semestrales

posteriores tendrá que acreditar una cantidad de créditos suficiente para completar el plan de estudios en un máximo de 16 semestres. Cabe mencionar, que una vez realizada la inscripción de asignaturas obligatorias para un período semestral, no se podrá cancelar dicha inscripción; y solo en el caso de las asignaturas de libre elección, se contará con un periodo de 10 días hábiles para su cancelación.

En caso de que el alumno no apruebe el examen ordinario de alguna de las asignaturas en las que esté inscrito por primera vez, para aprobarla, tendrá derecho a lo más a tres oportunidades para presentar examen extraordinario, y a repetir la asignatura una sola vez. Las tres únicas oportunidades de examen extraordinario podrá utilizarlas antes o después de repetir la asignatura, pero no al mismo tiempo de cursarla, y podrán ser distribuidas en el orden que el estudiante requiera. La calificación mínima aprobatoria de cada asignatura es de 60 puntos.

Cuando un estudiante pretenda cambiar de carrera entre las licenciaturas que ofrece la Facultad, sólo se podrá inscribir en caso de no haber agotado todas las oportunidades para acreditar alguna de las asignaturas comunes, y cuando el número de oportunidades utilizadas sea menor que el máximo establecido en el plan al que se pretende inscribir. Además, el número de oportunidades para cada una de las asignaturas en cuestión será el resultado de restarle las oportunidades ya utilizadas al número máximo de oportunidades. Por lo anterior, un estudiante no podrá inscribirse a alguna de las otras licenciaturas en la Facultad al haber agotado sus oportunidades en alguna de las asignaturas comunes.

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 basados en créditos, se presenta la Tabla 11 que establece cierta equivalencia entre los créditos aprobados por un alumno a lo largo de su trayectoria académica, y el semestre que podría acreditar —de acuerdo a un plan rígido.

Tabla 11. Relación de equivalencia entre créditos y semestres acreditados

Total de créditos aprobados	Semestre equivalente acreditado
27	1º
69	2º
122	3º

167	4°
220	5°
261	6°
318	7°
376	8°

Cabe resaltar, que la equivalencia propuesta es solamente para fines estadísticos, y no determina de manera alguna el período restante de que dispone un alumno para concluir el plan de estudios; dicho período se encuentra delimitado por la duración máxima establecida para concluir la carrera: 16 semestres.

## Requisitos de Egreso

Para egresar de la carrera el alumno debe de acreditar:

- ✓ 314 créditos correspondientes a las asignaturas obligatorias,
- ✓ 20 créditos correspondientes los talleres obligatorios,
- ✓ Al menos 6 asignaturas de especialidad —con un mínimo de 42 créditos.
- ✓ Acreditar el dominio de lectura y comprensión de inglés técnico.

## Requisitos de Titulación

Una vez acreditados los requisitos de egreso, el alumno estará en posibilidad de iniciar el proceso de titulación. Las opciones de titulación serán las establecidas en el Reglamento de Inscripciones y Exámenes de la UADY y el Reglamento Interior de la Facultad, bajo las condiciones establecidas en el manual de procedimiento de titulación correspondiente.

## IX. Plan de Liquidación

El plan de liquidación para el plan de estudios de 2004 se realizará de acuerdo a dos estrategias:

1. Se realizará un proceso de reconocimiento de estudios con base en lo establecido en el Reglamento de Incorporación y Revalidación de Estudios de la UADY, a aquellos alumnos que actualmente se encuentran inscritos al plan de estudios 2004 y que cumplan con alguna de las siguientes condiciones:
  - A. Que sean alumnos regulares o irregulares que hayan ingresado al primer semestre de la Licenciatura en Ingeniería de Software en agosto de 2008, sin importar si están inscritos o no a algún semestre actualmente.
  - B. Que sean alumnos regulares o irregulares, que pertenezcan a alguna de las generaciones anteriores al ingreso de agosto de 2008, que no hayan tenido inscripción a quinto semestre y que en agosto de 2009 no cumplan los requisitos para inscribirse a dicho semestre.
2. Para aquellos alumnos que no cumplan las condiciones establecidas en la primera estrategia, no habrá modificación alguna en su régimen académico-administrativo y permanecerán bajo las condiciones del plan de estudios 2004 hasta su egreso.

En el caso de los alumnos que estando inscritos al plan 2004 se incorporen al plan 2009, su período de permanencia en el programa será contabilizado considerando la fecha de ingreso al programa; así por ejemplo: un alumno que ingresó en septiembre de 2006 tendrá como límite para concluir la carrera, el semestre enero-julio de 2014.

Por otro lado, en el caso de asignaturas que habiéndose cursado bajo el esquema rígido no hayan sido aprobadas por el alumno al momento del proceso de reconocimiento, el número de oportunidades restantes se calculará considerando las oportunidades utilizadas hasta el momento; así



por ejemplo: un alumno que habiendo cursado una sola vez la asignatura Álgebra Lineal, y utilizado dos inscripciones a examen extraordinario, luego de su incorporación al plan 2009, le restará una segunda oportunidad de cursar la asignatura, y una tercer oportunidad de aprobarla en examen extraordinario.

## **X. Recursos Humanos y Físicos**

### **Recursos Humanos**

La planta docente que atiende a la Licenciatura en Ingeniería de Software es suficiente para llevar a cabo la implementación de la presente modificación; sin embargo, como parte del proceso de consolidación de la carrera se requerirá incrementar el número de profesores con posgrado en el área de Ingeniería de Software, de preferencia en el nivel de doctorado; dicho proceso permitirá consolidar las áreas disciplinarias que sustentan la carrera, y ofrecer alternativas de especialización —a cargo de los Cuerpos Académicos— para los estudiantes, en áreas de investigación, innovación y desarrollo, propias de la Ingeniería de Software. Así mismo, resulta conveniente promover convenios de colaboración con Instituciones de Educación Superior Nacionales o Extranjeras, reconocidas en el área de la Ingeniería de Software, a efecto de establecer mecanismos de movilidad entre profesores de dichas Instituciones.

### **Recursos Físicos**

La Facultad cuenta con la infraestructura suficiente (salones, aulas de cómputo, biblioteca, equipo de cómputo y audiovisual) para la matrícula que actualmente atiende; cabe mencionar que en agosto de 2009 deberá entrar en funcionamiento el Laboratorio de Ingeniería de Software, el cual contará con áreas de trabajo específicas para apoyar diversas asignaturas de los bloques disciplinario y de concentración; no obstante, en la medida en que los Cuerpos Académicos responsables de Líneas de Generación y Aplicación de Conocimiento (LGAC) vinculados con áreas de conocimiento disciplinarias o de especialidad se integren o consoliden, se requerirá adecuar o construir nuevos espacios físicos a efecto de facilitar la vinculación entre docencia y la investigación, desarrollo e innovación, propias de una disciplina ingenieril, como es el caso de la Ingeniería del Software.

Es importante mencionar, que a pesar de contar con el equipo suficiente, se requerirá la actualización constante de dicho equipos y programas de

cómputo que apoyan el desarrollo de la licenciatura. Por otro lado, será también necesaria la actualización periódica del material bibliográfico disponible, sobre todo, el material especializado que asiste las asignaturas de los bloques disciplinario y de especialidad en área de la Ingeniería de Software.

## **XI. Mecanismos de Evaluación Curricular**

Para mantener un óptimo desarrollo y una actualización curricular constante, el plan de estudios de la Licenciatura en Ingeniería de Software se evaluará en forma progresiva y permanente, mediante el trabajo de un comité de profesores pertenecientes a Cuerpos Académicos responsables de LGAC vinculados con la Ingeniería de Software.

### **Sistema de Evaluación**

Para obtener la información necesaria para el análisis y la adquisición de elementos de juicio para la evaluación interna del plan de estudios, al finalizar cada semestre el Comité de Evaluación Curricular, dependiente de la Secretaría Académica, realizará encuestas a alumnos y a profesores, donde se consideren diversos aspectos tales como:

- ✓ El logro de los objetivos de aprendizaje de cada asignatura.
- ✓ La calidad de los contenidos.
- ✓ Las estrategias de enseñanza utilizada por los profesores.
- ✓ Los criterios de evaluación de las asignaturas.
- ✓ Los logros terminales de los estudiantes comparados con el perfil del egresado.

A su vez, además de profesores y alumnos, se encuestará a los egresados y se entrevistará a expertos para la evaluación externa que permita valorar:

- ✓ La eficiencia del programa curricular en cuanto a su vinculación con las necesidades sociales en el área de su competencia.
- ✓ Las funciones que desempeñan los egresados, así como su campo de trabajo.
- ✓ La demanda permanente de interesados.
- ✓ El avance en los conocimientos y la tecnología relacionados.

## **XII. Descripción Sintética de las Asignaturas del Nivel Básico.**

## ÁLGEBRA SUPERIOR I

Horas: 75 T  
Créditos: 10  
Clave MT-01

### OBJETIVO:

Utilizar con fluidez el lenguaje matemático, manejar los aspectos de la lógica matemática que justifican los métodos de demostración matemáticos, aplicar los conceptos y resultados fundamentales de relaciones y funciones entre conjuntos, así como el cálculo combinatorio y estructuras algebraicas.

### CONTENIDO.

---

1. Lógica, métodos de demostración y conjuntos
2. Relaciones y Funciones
3. Cálculo Combinatorio
4. Introducción a las estructuras algebraicas

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, grupos de discusión, trabajo en equipo, demostración.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Knuth, Donald. The art of Computer Programming. Vol. 1,2,3. Addison Wesley, 1998.
- 2.- Ash, R.B. A Primer of Abstract Mathematics. Mathematical Association of America, 1998.
- 3.- Cárdenas, Humberto, et. Al. Álgebra Superior, Trillas, 1974.
- 4.- Castro Gustavo, et.al. Álgebra II, Sección Matemática Educativa, CIEM-IPN.1986

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## ÁLGEBRA SUPERIOR II

Horas: 75 T  
Créditos: 10  
Clave MT-02

### OBJETIVO:

Manejar la estructura numérica de los complejos, los resultados fundamentales de la divisibilidad en el anillo de los enteros, de los polinomios y matrices, y sus operaciones.

### CONTENIDO.

---

1. Números complejos
2. Divisibilidad en los números enteros
3. Polinomios
4. Matrices con componentes en un campo arbitrario

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, grupos de discusión, trabajo en equipo, demostración.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Knuth, Donald. The art of Computer Programming. Vol. 1,2,3. Addison Wesley, 1998.
- 2.- Ash, R.B. A Primer of Abstract Mathematics. Mathematical Association of America, 1998.
- 3.- Cárdenas, Humberto, et. Al. Álgebra Superior, Trillas, 1974.
- 4.- Castro Gustavo, et.al. Álgebra II, Sección Matemática Educativa, CIEM-IPN.1986

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## ÁLGEBRA LINEAL

Horas: 75 T  
Créditos: 10  
Clave MT-03

### OBJETIVO:

Manejar en forma teórica y práctica los conceptos fundamentales para el Álgebra Lineal aplicada, dándole mayor importancia a la aplicación en computación.

### CONTENIDO.

---

1. Sistemas de ecuaciones lineales
2. Determinantes
3. Espacios vectoriales
4. Matrices y transformaciones lineales
5. Valores y vectores propios

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y en tareas, tormenta de ideas, grupos de discusión, trabajo en equipo, demostración, investigación bibliográfica.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Noble, Benn, et. al. Álgebra Lineal Aplicada 3ª. Ed. Prentice Hall 1989.
- 2.- Pita Ruiz, Claudio. Álgebra Lineal. McGraw Hill 1991.
- 3.- Perry, William. Álgebra Lineal con Aplicaciones. McGraw Hill 1990.
- 4.- Foley, James, et. al. Introduction to computer graphics. Addison Wesley. 1994
- 5.- Eran, Donald, et. al. Gráficas por computadora, 2ª Ed. Prentice Hall. 1995

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## CÁLCULO DIFERENCIAL

Horas: 105 T, 15 P  
Créditos: 14  
Clave MT-04

### OBJETIVO:

Manejar las propiedades de los números reales y el concepto de la derivada; deducir y aplicar las técnicas de derivación, demostrar y aplicar los principales resultados que provienen del concepto de la derivada, y resolver problemas geométricos y físicos empleando las propiedades, técnicas y principales resultados del cálculo diferencial.

### CONTENIDO.

---

1. Los Números Reales
2. Funciones reales de variables reales
3. Límites y continuidad
4. Derivación de funciones reales de variables reales
5. Teoremas de derivación
6. Aplicaciones de la derivada

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	85
Taller de ejercicios	15
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Apóstol, Tom M. Calculus, Vol I. México: Reverté, 1979.
- 2.- Haaser, Norman B. Análisis Matemático, Vol I. México: Trillas, 1970.
- 3.- Hagin, F.;Cohen, J. Calculus exploration using Matlab, 1999.
- 4.- Hughes-Hallet, Deborah, et.al. Cálculo aplicado. México: CECSA, 1999.
- 5.- Hughes-Hallet, Deborah, et.al. Cálculo 2ª. Edición. México: CECSA, 2001.
- 6.- Spivak, Michael. Calculus infinitesimal. México: Reverté, 1988.
- 7.- Stewart, J. Cálculo: conceptos y contextos. México: Thompson, 1998.

8.- Stewart, J. Cálculo: trascendentales tempranas. México: Thompson, 1998.

9.- Strang, G. Calculus. USA: Wellesley Cambridge Press, 1991

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.



## CÁLCULO INTEGRAL

Horas: 105 T, 15 P  
Créditos: 15  
Clave MT-05

### OBJETIVO:

Manejar el concepto de integral y las fórmulas básicas de integración; deducir y aplicar las fórmulas de integración, demostrar y manejar los principales resultados de integración de funciones reales de variable real, resolver problemas geométricos y físicos, empleando las propiedades, técnicas y principales resultados del cálculo integral. Manejar, demostrar y aplicar el concepto de convergencia para sucesiones y series.

### CONTENIDO.

---

1. Integración de funciones reales de variable real
2. Aplicaciones de la integral: Cálculo de áreas y volúmenes
3. Teoría de integración
4. Relación entre integración y derivación
5. Integrales impropias
6. Aplicaciones: área de superficies de revolución, longitud de arco de una curva, probabilidad, masa y momento, fuerza, trabajo y energía
7. Sucesiones
8. Series
9. Funciones que admiten una expresión en serie
10. Convergencia uniforme

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	85
Taller de ejercicios	15
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Apóstol, Tom M. Calculus, Vol I. México: Reverté, 1979.
- 2.- Bartle, Robert;. The element of real analysis. Nueva York: John Wiley, 1975.

- 3.- Courant, John. Introducción al cálculo y al análisis matemático, vol. 2. Limusa, 1979.
- 4.- Haaser, Norman B. Análisis Matemático, Vol I. México: Trillas, 1970.
- 5.- Hughes-Hallet, Deborah, et.al. Cálculo de varias variables. México: CECSA, 1999.
- 6.- Stewart, J. Cálculo: conceptos y contextos. México: Thompson, 1998.
- 7.- Stewart, J. Cálculo: transcendentales tempranas. México: Thompson, 1998.
- 8.- Stewart, J. Cálculo Multivariable. México: Thompson, 1999.
- 9.- Strang, G. Calculus. USA: Wellesley Cambridge Press, 1991

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## PROBABILIDAD

Horas: 75 T  
Créditos: 10  
Clave: MT-08

### OBJETIVO:

Manejar los conceptos básicos de probabilidad y estadística, identificar los problemas clásicos que pueden resolverse con las técnicas fundamentales del área y aplicar la teoría en la resolución de problemas.

### CONTENIDO.

---

1. Conceptos básicos de probabilidad
2. Variables aleatorias unidimensionales
3. Familias paramétricas especiales de distribuciones univariadas
4. Variables aleatorias multidimensionales
5. Distribuciones de funciones de variables aleatorias
6. Muestreo y distribuciones muestrales

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, grupos de discusión, trabajo en equipo, demostración.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Meyer, P. Probabilidad y Aplicaciones Estadísticas. Fondo Educativo Interamericano. 1973.
- 2.- Hoffman-Jorgensen, J. Probability with a view towards statistics. Chapman and Hall. 1994
- 3.- Ross, S.M. A first course in probability. McMillan Publishing Co. 1988
- 4.- Cheng, K.L. Elementary Probability Theory with Stochastic Processes. Springer-Verlag. 1975.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## INFERENCIA ESTADÍSTICA

Horas: 75 T  
Créditos: 10  
Clave MT-09

### OBJETIVO:

Manejar y aplicar los conceptos de estimación, prueba de hipótesis y regresión lineal a problemas prácticos, que indicarán la importancia de la inferencia estadística como apoyo en la investigación científica.

### CONTENIDO.

---

1. Estimación puntual
2. Estimación de intervalos
3. Pruebas de hipótesis
4. Regresión lineal simple

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, grupos de discusión, trabajo en equipo, investigación bibliográfica.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Chatterjee, S. y Price B. Regression Analysis by Example, 2ª edición. Nueva York: John Wiley & Sons, 1991.
2. Guttman, I. Linear Models: An Introduction. Nueva York: John Wiley & Sons, 1982.
3. Hogg, R. y Craig, A. T. Introduction to Mathematical Statistics, 5ª edición. Nueva Jersey, EE. UU.: Prentice Hall, 1995.
4. Iversen, G.R. y Georgen, M.S. Statistics: The Conceptual Approach. Springer, 1997.
5. Kiefer, J. C. Introduction to Statistical Inference. Nueva York: Springer – Verlag, 1987.
6. Kreyszig, E. Introducción a la Estadística Matemática: Principios y métodos. México, D.F.: Grupo Editorial Iberoamérica, 1979.

7. Lehman, E. L. Testing Statistical Hypothesis. Nueva York: John Wiley & Sons, 1980.
8. Mendenhall, W., Wackerly, D.D. y Scheaffer, R.L. Estadística Matemática con Aplicaciones, 2ª edición. México, D.F.: Grupo Editorial Iberoamérica, 1994.
9. Montgomery D.C. y Peck E. Introduction to Linear Regression Analysis. John Wiley & Sons, 1992.
10. Mood, A. M., Graybill, F. A. y Boes, D. Introduction to the Theory of Statistics, 3ª edición. Nueva York: Mc Graw Hill, 1974.
11. Mukhopadhyay, N. Probability and Statistical Inference. Marcel Dekker, 2000.
12. Silvey, S.D. Statiscal Inference. Londres, Inglaterra: Chapmand Hall, 1975.
13. Sincich, T. Statistics by Example. 4ª edición. San Francisco, EE. UU: Dellen-Macmillan, 1990.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Matemáticas, Licenciado en Enseñanza de las Matemáticas o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## MATEMÁTICAS DISCRETAS

Horas: 60 T 15 P  
Créditos: 9  
Clave MT-12

### OBJETIVO:

Aplicar los conceptos matemáticos fundamentales para analizar los procesos algorítmicos, enfatizando en el razonamiento formal de los mismos y poder implementarlos con eficiencia.

### CONTENIDO.

---

1. Funciones Recursivas
2. Funciones Enteras
3. Teoría de grafos
4. Teoría de árboles

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, grupos de discusión, trabajo en equipo, desarrollo de programas de cómputo.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Kenneth, Rosen. Discrete Mathematics and its Applications. Mc Graw Hill. 1991
2. Anderson, James. Discrete Mathematics with Combinatorics. Prentice Hall, 2001
3. Mattson, H.F. Discrete Mathematics with Applications. John Wiley & Sons, 1993
4. Anderson, Ian. First Course in Discrete Mathematics. Springer, 2000
6. Graham, Knuth, Patashnik. Concrete Mathematics. Addison Wesley. 1994
7. Knuth, Donald. The Art of Computer Programming, Vol 1. Addison Wesley 1997.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## EVALUACIÓN DE PROYECTOS

Horas: 60 T 15P  
Créditos: 9  
Clave MT-20

### OBJETIVO:

Tomar decisiones con relación a la alternativa más económica y deseable en propuesta de inversión, mediante la aplicación de los métodos y técnicas de la Ingeniería Económica.

### CONTENIDO.

---

1. Introducción a la Ingeniería Económica
2. Valor del dinero en el tiempo
3. Anualidades
4. Bases para la comparación de alternativas
5. Toma de decisión para la selección de alternativas
6. Evaluación de alternativas en situaciones de reemplazo
7. Análisis de alternativas considerando impuestos
8. Evaluación de Proyectos

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Baca, Urbina (1996). Fundamentos de Ingeniería Económica. Mc Graw Hill
2. Blank, Leland; Tarquín, Anthony (1992). Ingeniería Económica. Mc Graw Hill
3. Canada, John (1985). Técnicas de Análisis económico para Administradores e Ingenieros. Diana.
4. Coss, Raúl (1990). Análisis y Evaluación de Proyectos de Inversión. Limusa-Wiley.
5. Taylor, George (1990). Ingeniería económica. Limusa – Wiley.



**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Actuaría, en Economía o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## FUNDAMENTOS DE PROGRAMACIÓN

Horas: 45 T 30 P  
Créditos: 8  
Clave: AP-01

### OBJETIVO:

Aplicar los fundamentos de la programación en el desarrollo de programas de cómputo básicos y utilizar el material de referencia básico para ampliar su conocimiento de bibliotecas de funciones especializadas y herramientas para el desarrollo de programas.

### CONTENIDO.

---

1. Lógica de programación
2. Variables, operadores y expresiones
3. Estructuras de control y arreglos
4. Paso de parámetros y manejo de funciones
5. Pilas y colas
6. Alcance de variables
7. El proceso de desarrollo
8. Inspección de código
9. Modularidad
10. Pruebas unitarias

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, grupos de discusión, resolución de problemas de programación en clase y en tareas, trabajo en equipo, desarrollo de programas de cómputo, investigación bibliográfica.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas y proyectos	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Knuth, Donald, The Art of Computer Programming, Addison Wesley. 1997
- 2.- Schildt, Herbert. Lenguaje C, Programación Avanzada. McGraw Hill.
- 3.- Booch, Grady. Diseño Orientado a Objetos con Aplicaciones, Addison Wesley, 1995.

- 4.- Bronson, Gary. Algorithm Development and Program Design Using C, PWS Publishing Co.; Book and Disk edition (February 15, 1996) ISBN: 0314069879
- 5.- Standish, Thomas. Data Structures, Algorithms, and Software Principles in C, Addison-Wesley Pub Co; 1st edition (September 30, 1994), ISBN: 0201591189
- 6.- Linden, Peter. Expert C Programming, Prentice Hall, 1994, ISBN 0131774298
- 7.- Reek, Kenneth. Pointers on C. Addison Wesley, 1997. ISBN 067399866
- 8.- Andrew, James. An Introduction to Data Structures and Algorithms, Birkhauser Boston; 1st edition (November 9, 2001) ISBN: 0817642536
- 9.- Kernighan, Brian W. & Dennis, M. Richie. El lenguaje de programación C. 2ª Ed. Prentice Hall. 1988
- 10.- Humphrey, Watts. A Discipline for Software Engineering. Addison-Wesley, Boston, 1995.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## PROGRAMACIÓN

Horas: 45 T 30 P  
Créditos: 8  
Clave AP-02

### OBJETIVO:

Aplicar los fundamentos de la programación orientada a objetos en el desarrollo de programas de cómputo, considerando los criterios de codificación y elaboración de programas para llevar a una formalidad y eficiencia en la ejecución de los mismos.

### CONTENIDO.

---

1. Clases y objetos
2. Diagramas y códigos
3. Herencia y eventos
4. Excepciones y polimorfismo
5. Recursión y sobrecarga
6. Aserciones y liberación
7. Hilos
8. Separación de lógica e interfase
9. Uso de la IDE
10. APIs

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, grupos de discusión, trabajo en equipo, desarrollo de programas de cómputo, investigación bibliográfica, resolución de problemas de programación en clase y en tareas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas y proyectos	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Budd, Timothy. Introducción a la programación orientada a objetos. Addison Wesley, 1994
- 2.- Bloch, Joshua. Effective Java Programming Language Guide. Addison Wesley, 2001. ISBN 0201310058

- 3.- Lafore, Robert. Data Structure and Algorithms in Java (2<sup>nd</sup> Edition), SAMS edition, ISBN 0672324539
- 4.- Lafore, Robert. Object Oriented Programming in C++ (4<sup>th</sup> edition). SAMS edition, 2001. ISBN 0672323087
- 5.- Allen, Mark. Data Structures and Algorithm Analysis in C++ (2nd Edition). Pearson Addison Wesley, 1998, ISBN 0201361221

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o carrera afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

### **XIII. Descripción Sintética de las Asignaturas del Nivel Disciplinario.**

## METODOLOGÍA DE LA INVESTIGACIÓN

Horas: 75 T  
Créditos: 10  
Clave ES-02

### OBJETIVO:

Utilizar y diferenciar las metodologías de la investigación científica a través de un modelo de aprendizaje orientado a la comprensión de los procesos de la investigación científica.

### CONTENIDO.

---

1. Introducción a la Investigación Científica
2. La investigación científica en computación
3. Documentos de investigación

**ESTRATEGIAS DE ENSEÑANZA:** Al principio del curso, el alumno recibirá un artículo de investigación científica en computación y un protocolo de investigación sobre un tema ajeno a la computación. Sobre el primer documento, el alumno deberá identificar las partes de este, así como investigar el estado del arte descrito en el artículo, justificar la propuesta, escribir un texto de divulgación sobre la teoría fundamental del artículo. Habrán exámenes rápidos sobre las lecturas, participaciones, programas y voluntarias de cada alumno, exámenes parciales y tareas. Análisis de los principales artículos de la ingeniería de software.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Ruiz y Ayala, El método de las ciencias. El método de la ciencias, FCE, 2000,
2. Sanchez Mora, Ana María, La divulgación de la Ciencia como Literatura, Dirección General de Divulgación de la Ciencia, Universidad Nacional Autónoma de México, 1998.
3. Chavez Arredondo, Nemesio, Todo por Saber, Dirección General de Divulgación de la Ciencia, Universidad Nacional Autónoma de México, 1998.
4. Bonfíl, Martín, Notas del Diplomado en Divulgación de la Ciencia.2001.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o carrera afín, con posgrado y experiencia docente, y/o de investigación.



## GESTIÓN TECNOLÓGICA

Horas: 75 T  
Créditos: 10  
Clave ES-03

### OBJETIVO:

Explicar los principios de gestión tecnológica basados en la generación, adquisición y transformación del conocimiento y describir la tecnología con un enfoque sistémico.

### CONTENIDO.

---

1. Gestión tecnológica: descripción, campo de acción e implicaciones
2. El ciclo de la tecnología
3. La base tecnológica de la empresa
4. Análisis de la tecnología: una base para la experiencia tecnológica
5. Tecnología, estrategia y competitividad
6. Investigación básica orientada: asociaciones industria - universidad
7. Manejo de la innovación basada en la tecnología
8. La cultura técnica y el imperativo del conocimiento
9. Propiedad intelectual e industrial
10. Casos de estudio

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, grupos de discusión, trabajo en equipo, presentaciones orales y escritas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Revistas Harvard Business Review, The Economist
- 2.- Gaynor, Gerard, Manual de Gestión en Tecnología, Mc Graw Hill, 2001
- 3.- Rosegger, Gerhard. The Economics of Production and Innovation, B-H 1996
- 4.- Rothery, Brian. Outsourcing, Limusa 2000.
- 5.- Kubr, Milan. La consultoría de empresas. Limusa, 2000.

6.- Derry, TT.K. et al, Historia de la Tecnología, Vol. I, II, III, IV, V, VI. Siglo XXI Editores, 1977.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o carrera afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## TEORÍA DE LA COMPUTACIÓN

Horas: 75 T  
Créditos: 10  
Clave MT-13

### OBJETIVO:

Aplicar los mecanismos matemáticos asociados a las máquinas finitas para representar procesos, considerando sus gramáticas, y expresiones regulares asociadas para optimizar los autómatas.

### CONTENIDO.

1. Lenguajes y Expresiones Regulares
2. Autómatas Finitos
3. Gramáticas y Lenguajes Libres de Contexto
4. Autómatas de Pila

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

1. Hopcroftt, John; Ullman Jeffrey. Introduction to Automata Theory languages, and Computation. Addison Wesley. 1993.
2. Brena, Ramón. Lenguajes Formales y Autómatas. Centro de Inteligencia Artificial, Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Monterrey. 2003.
3. Brookshear, J. Glenn. Teoría de la Computación, Lenguajes formales, Autómatas y Complejidad. Addison Wesley Iberoamericana. 1993.
4. Sanchez, Jesús; Quintana, Maricela. Material del curso Teoría de la Computación, impartida en el ITESM campus Estado de México. 1998.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## ARQUITECTURA DE COMPUTADORAS

Horas: 60 T 15 P  
Créditos: 9  
Clave AC-01

### OBJETIVO:

Describir el desarrollo tecnológico de la arquitectura de computadoras; identificar y describir sus niveles de lógica digital, microprogramación y máquina convencional a partir del núcleo de un sistema operativo.

### CONTENIDO.

---

1. Organización y diseño básico de computadoras
2. Nivel de lógica digital
3. Nivel de microprogramación
4. Nivel de máquina convencional
5. Nivel de programación y sistema operativo
6. Arquitecturas avanzadas

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tareas, desarrollo de prototipos, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Tareas	20
Proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Morris, Mano. Arquitectura de Computadora. Tercera Edición. Prentice Hall.
2. William, Staleings. Organización y arquitectura de computadoras. Cuarta Edición, Prentice Hall.
3. Andrew, Tanenbaum. Organización de computadoras, un enfoque estructurado. Tercera Edición, Prentice Hall.
4. Maccabe. Sistemas Computacionales, arquitectura y organización. Primera edición, Editorial Irwing.

5. Bovet, Daniel; Cesati, Marco. Understanding the Linux Kernel. O'Reilly & Associates, 2<sup>nd</sup> edition. 2002. ISBN 0596002130

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero en Electrónica, Licenciado en Ciencias de la Computación o afines, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## ESTRUCTURAS DE DATOS

Horas: 45 T 30 P  
Créditos: 8  
Clave AP-03

### OBJETIVO:

Utilizar un lenguaje de programación orientado a objetos para desarrollar programas de cómputo que utilicen estructuras de datos avanzadas, considerando la optimización de recursos como el espacio de memoria y de disco.

### CONTENIDO.

---

1. Programación por contratos
2. Programación defensiva
3. Árboles y listas
4. Ordenamiento y búsqueda
5. Diccionarios
6. Interrupciones
7. Multiprocesamiento
8. Índices invertidos

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, grupos de discusión, trabajo en equipo, desarrollo de programas de cómputo, investigación bibliográfica, resolución de problemas en clase y en tareas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas y proyectos	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Heileman, Gregory, Data Structures, Algorithms, and Object Oriented Programming, McGraw Hill, 1996
- 2.- Stroustrup, Bjarne, "The C++ Programming Language" 2° Edition - Addison Wesley 1994
- 3.- Eckel, Bruce. Thinking in C++, Prentice Hall. 2000.

- 4.- Cormen, T.H., Leiserson, Ch. E., Rivest, R.L., Stein, C. Introduction to Algorithms. MIT Press, 2001.
- 5.- Goodrich, Michael; tamassia, Roberto. Data Structures and Algorithms in Java. Wiley Text books, 2003, ISBN 0471469831
- 6.- Allen, Mark. Data Structures and algorithm analysis in java. Peachpit Press, 1998. ISBN 0201357542.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## TEORÍA DE LENGUAJES DE PROGRAMACIÓN

Horas: 45 T 30 P  
Créditos: 8  
Clave AP-04

### OBJETIVO:

Conocer los fundamentos, conceptos básicos y aspectos de diseño e implementación de los principales lenguajes de programación, para realizar una elección más adecuada del paradigma y lenguaje de programación a utilizar en la implementación de sistemas que formen parte de la solución a un problema.

### CONTENIDO.

---

1. Conceptos básicos de lenguajes de programación
2. Semántica y sintaxis
3. Identificadores y variables
4. Tipos de datos
5. Expresiones y asignaciones
6. Estructuras de control
7. Subprogramas
8. Paradigmas de programación

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios de programación en clase y de tarea, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	70
Tareas y proyectos	30
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Sebesta, Robert. Concepts of Programming Languages. 4<sup>a</sup>. Edición. Addison Wesley. 1999.
2. Loudon, Kenneth. Programming Languages. Principles and Practice. PWS Publishing Company. USA 1993.
3. Pratt, Terrence W.; Zelkowitz, Marvin V. Programming Languages, Design and Implementation. Prentice Hall. 1999.



4. Tucker, A; Noonan, R. Lenguajes de Programación, principios y paradigmas. McGraw-Hill. 1998.
5. Friedman, Daniel; Wand, Mitchell; Hayness, Chistoper. Essentials of Programming Languages. McGraw-Hill. 1994. Caps. 1,2,3.
6. Moriber, Harry. Structured Basic Programming. Charles E. Merrill Publishing Co. 1992.
7. Hedefmeim, Goodman. Introduction to the design and analysis of algorithms. USA. 1993

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## PROGRAMACIÓN EN LA WEB

Horas: 45 T 30P  
Créditos: 8  
Clave AP-07

### OBJETIVO:

Aplicar las principales técnicas y herramientas de la programación en web para construir aplicaciones que se ejecuten en páginas web.

### CONTENIDO.

---

1. Utilización del Web como plataforma de Desarrollo.
2. Arquitecturas de organización y desarrollo.
3. Tecnologías del lado del Cliente.
4. Servidores Web
5. Tecnologías del lado del Servidor.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas, desarrollo de prototipos.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Trabajos de investigación y prácticas	20
Proyecto	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Darnell Rick, HTML 4 Unleashed, SAMS, 2a. edición, 1998.
2. Meyer Eric; Cascading Style Sheets: The Definitive Guide; O'Reilly & Associates; 2000.
3. Flanagan David; JavaScript: The Definitive Guide; O'Reilly & Associates; 4a. edición, 2001.
4. Laurie, Laurie; Apache: The Definitive Guide; O'Reilly & Associates; 3a edición, 2002.
5. Tulloch Mitch; IIS 6 Administration; Osborne Distributed Product; 2003.
6. Hamilton Jacqueline; CGI Programming 101; CGI101.com; 2000

7. Sybex, Sybex Inc, Sybex Systems; Perl, CGI, and JavaScript Complete; Sybex; 2a edición, 2003.
8. Homer Alex; Professional Active Server Pages 3.0; Wrox Press Inc; 1999.
9. Welling, Thomson; PHP and MySQL Web Development; SAMS; 2a. edición, 2003.

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## INTERACCIÓN HUMANO COMPUTADORA

Horas: 45 T 30 P  
Créditos: 8  
Clave IH-05

### OBJETIVO:

Aplicar los fundamentos y métodos utilizados para la construcción de interfaces de usuario que faciliten el uso de la computadora por los usuarios finales.

### CONTENIDO.

---

1. Estilos de interacción
2. Diseño de interfaces centradas en el usuario
3. Soporte e información en línea
4. Arquitecturas de presentación

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tareas, desarrollo de prototipos, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	75
Trabajos de investigación, tareas y proyectos	25
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Shneiderman Ben; Designing the User Interface; Pearson Addison Wesley; 3a. edición, 1997.
2. Dix, Finlay, Abowd, Beale, Finley; Human-Computer Interaction; Prentice Hall; 2a. edición, 1998.
3. Vredenburg, Isensee, Righi; User-Centered Design: An Integrated Approach; Prentice Hall, 2001.
4. Cooper, Reimann; About Face 2.0: The Essentials of Interaction Design; John Wiley & Sons; 2a. edición 2003.
5. Kuniavsky Mike; Observing the User Experience: A Practitioner's Guide to User Research; Morgan Kaufmann; 2003.
6. Isaacs, Walendowski; Designing from Both Sides of the Screen: How Designers and Engineers Can Collaborate to Build Cooperative Technology; SAMS; 2001

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## BASES DE DATOS

Horas: 75 T  
Créditos: 10  
Clave: SI-01

### OBJETIVO:

Aplicar los conceptos y características sobre los modelos de Bases de Datos en la solución de problemas relacionados con sistemas de información.

### CONTENIDO.

---

1. El modelo relacional
2. El diseño lógico
3. Arquitecturas de Bases de Datos
4. Sistemas Manejadores de Bases de Datos

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tareas, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, desarrollo de prototipos.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Date, C.J. Introducción a los Sistemas de Bases de Datos (1993). Quinta Edición. Addison Wesley
2. Elmasri, Ramez y Navathe, Shamkant (1997). Sistemas de Bases de Datos. Addison Wesley – Iberoamericana.
3. Silberschatz, Abraham; Korth, Henry y Sudarshan, S. (1998). Fundamentos de Bases de Datos. Tercera Edición. Mc Graw Hill
4. Martin, James (1988). Organización de las Bases de Datos. Prentice Hall.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

Horas: 60 T 15 P  
Créditos: 9  
Clave IS-01

### OBJETIVO:

Describir el contexto y la ética de la profesión de la ingeniería de software, así como los diversos modelos y procesos de desarrollo de software conociendo ampliamente cada una de las etapas por las que atraviesa un proyecto de desarrollo de software, y aplicar los conceptos básicos del análisis y diseño para un sistema de software.

### CONTENIDO.

1. El Contexto de la Ingeniería de Software: inicios, presente y futuro (problemas y avances).
2. La ética y la Ingeniería de Software.
3. Modelos de Ciclo de Vida.
4. Descripción de las fases de los modelos de ciclo de vida.
5. Modelos de Mejora y de Determinación de Capacidades de Procesos de Software, tanto a nivel de organización, a nivel de equipo como a nivel de Ingeniero de Software.
6. Introducción al análisis y diseño del software.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo en equipo, investigaciones bibliográficas, presentaciones orales y escritas; análisis y diseño de problemas en clase y de tarea, ejercicios prácticos de programación en clase y de tarea para poner en práctica prácticas probadas de la ingeniería de software.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Tareas y proyectos	40
Total	100 puntos

## BIBLIOGRAFÍA.

---

1. Ghezzi, Carlo & Jazayeri, Mehdi & Mandrioli, Dino. Fundamentals of Software Engineering (2nd Edition). Prentice Hall; 2nd edition (September 19, 2002). ISBN: 0133056996
2. Baase, Sara. A Gift of Fire, 2nd edition. Prentice-Hall, 2002. Prentice Hall; 2nd edition (July 15, 2002) ISBN: 0130082155
3. Software Engineering Code of Ethics and Professional Practice (Version 5.2) as recommended by the ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices and jointly approved by the ACM and the IEEE-CS as the standard for teaching and practicing software engineering. <http://www.acm.org/serving/se/code.htm>
4. Beck, Kent. Extreme Programming Explained: Embrace Change. Addison-Wesley Pub Co; 1st edition (October 5, 1999) ISBN: 0201616416
5. Humphrey, Watts. A Discipline for Software Engineering. Addison-Wesley, Boston, 1995.
6. Humphrey, Watts. Introduction to the Team Software Process. Addison-Wesley, Boston, 2000.
7. Paul, Mark, Webber, C., Curti, B., and Chrissis, M., The Capability Maturity Model for Software v. 1.1 CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh Pa, February 1993
8. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Continuous Representation. CMU/SEI 2000
9. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Staged Representation. CMU/SEI 2000
10. ISO/IEC 15504, Software Process Improvement and Capability Determination.
11. Maciaszek, Leszek. Requirements Analysis and System Design, Developing Information Systems with UML. Addison-Wesley, 2001.
12. Sommerville, Ian. Ingeniería de Software (5ª o 6ª edición). Addison-Wesley Pub Co;
13. Pressman, Roger. Ingeniería de Software: Un enfoque práctico (5ª o 6ª edición). McGraw-Hill.
14. Pfleeger, Lawrence. Ingeniería de Software, Teoría y Práctica. Pearson Educación. 2002

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.



## DESARROLLO DE REQUISITOS DE SOFTWARE

Horas:	45 T 30 P
Créditos:	8
Clave	IS-02

### Objetivo

Especificar los requisitos de un sistema de software usando diversos métodos, herramientas y técnicas para la educación (elicitación), análisis y su seguimiento, asegurando la calidad de los mismos. Todo comparando metodologías pesadas y ligeras dentro de diversos dominios de aplicación (sistemas de información, sistemas bajo el web, sistemas de ingeniería y científicos, sistemas empotrados, etc.).

### CONTENIDO.

---

1. El contexto del proceso de desarrollo de requisitos de software (problemas y beneficios).
2. El proceso de desarrollo de requisitos de software.
3. Métodos, herramientas y técnicas para la educación de requerimientos.
4. Métodos, herramientas y técnicas para el análisis de requerimientos.
5. Métodos, herramientas y técnicas para la especificación de requerimientos (requerimientos funcionales, no funcionales, atributos de calidad de los requerimientos).
6. Métodos, herramientas y técnicas para la verificación y validación de requerimientos.
7. El proceso de administración de requisitos de software.
8. Introducción a la especificación formal de requerimientos (diagramas de estado, Z).

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo personal y en equipo, investigaciones bibliográficas, presentaciones orales y escritas, desarrollo de ejemplos de especificación de requisitos, simulaciones para la educación de requisitos.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Proyectos	40
Tareas	10
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. G. Montoya and I. Somerville. Requirements Engineering: processes and Techniques. John Wiley and Sons. ISBN 0-471-97208-8.
2. Dean Leffingwell , Don Widrig. Managing Software Requirements: A Use Case Approach. Addison-Wesley Pub Co; 2nd edition (May 16, 2003) ISBN: 032112247X
3. Maciaszek, Leszek. Requirements Analysis and System Design, Developing Information Systems with UML. Addison-Wesley, 2001 ISBN 0201709449
4. UML Distilled: A Brief Guide to the Standard Object Modeling Language (2nd Edition) by Martin Fowler, Kendall Scott Addison-Wesley Pub Co; 2nd edition (August 25, 1999)
5. Jacky, Jonathan. The Way of Z : Practical Programming with Formal Methods. Cambridge University Press; (November 28, 1996) ISBN: 0521559766
6. Jackson, M., Software Requirements and Specifications, Reading, Massachusetts: Addison Wesley, 1995.
7. Exploring Requirements: Quality Before Design. Donald Gause & Gerald Weinberg. Dorset House Publishing, 1989
8. Mastering the Requirements Process. Suzanne & James Robertson. Addison-Wesley, 1999. (Required)

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## DISEÑO DE SOFTWARE

Horas: 45 T, 30P  
Créditos: 8  
Clave IS-03

### OBJETIVO:

Explicar y aplicar los fundamentos, métodos, técnicas y herramientas para la construcción de sistemas de software a gran escala bajo el contexto de buenas prácticas de diseño de software desde su arquitectura hasta cada uno de sus componentes usando en lo posible patrones de diseño.

### CONTENIDO.

---

Ubicación del diseño dentro del ciclo de desarrollo  
Diseño detallado.  
Documentación del diseño.  
Evolución del diseño.  
Diseño orientado a sistemas distribuidos.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, desarrollo de prototipos, tormenta de ideas, uso de software, trabajo en equipo, análisis y demostración de problemas de diseño, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Trabajos de investigación y tareas	20
Proyecto	30
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Rumbaugh, James; Blaha, Michael; Prenerlani, William; Object Oriented Modeling. Prentice Hall Press. 1991.
- 2.- Rumbaugh, James; Jacobson, Ivar; Booch Grady. The Unified Modeling Lenguaje. Reference Manual. 1998.
- 3.- Kendall y Kendall. Análisis y Diseño de Sistemas . 3ª Ed. Prentice Hall, 1997.
- 4.- Booch, Grady. Diseño Orientado a Objetos con Aplicaciones. Addison Wesley. 1996.
- 5.- Gamma, E. Design Patterns. Addison Wesley. 1995.

6.- Object Oriented Design Heuristics. Addison Wesley. 1996

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE

Horas: 45 T 30 P  
Créditos: 8  
Clave IS-05

### OBJETIVO:

Aplicar la teoría básica relacionada con la construcción y evolución del software para llevar un control riguroso en el desarrollo de software.

### CONTENIDO.

---

1. Estándar y estilo de codificación.
2. Codificación eficiente.
3. Reutilización y Refactorización.
4. Pruebas y revisión al código.
5. Mantenimiento y Documentación

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas, resolución de problemas en clase y de tareas, modificaciones de documentos de análisis, diseño y código seleccionado.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	35
Trabajos de investigación y tareas	15
Proyecto	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Fowler, Martin; Beck, Kent; Brant, John. Refactoring: Improving the Design of Existing Code. 1999.
- 2.-Beck, Kent. Test Driven Development: By Example. 2002.
- 3.- Astels, David. Test Driven Development: A Practical Guide. 2003.
- 4.- Maguire, Steve. Writing Solid Code. Microsoft Press. 1993.
- 5.- Mili, Hamed; Mili, Ali; Yacoub, Sherif; Addy, Edward. Reuse-Based Software Engineering: Techniques, Organizations, and Controls(2001).
- 6.- Piattini, Mario. Mantenimiento de Software. AlfaOmega-Rama. ISBN 970-15-0730-4
- 7.- Pigoski, Thomas. Practical Software Maintenance: Best Practices for Managing your Software Investment. Wiley. ISBN 0-471-17001-1

8.- Bays, Michael. Software Release Methodology. Prentice Hall 1999. ISBN 0136365647

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afin, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## ARQUITECTURAS DE SOFTWARE

Horas: 45 T 30 P  
Créditos: 8  
Clave IS-06

### OBJETIVO:

Aplicar los fundamentos, métodos, técnicas y herramientas para la construcción de sistemas de software a gran escala, desde su arquitectura hasta cada uno de sus componentes usando en lo posible patrones de diseño.

### CONTENIDO.

---

1. Arquitecturas de software.
2. Patrones de diseño.
3. Middleware (CORBA, .NET).
4. Diseño de sistemas distribuidos.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas, resolución de problemas en clase y de tarea, desarrollo de prototipos.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Proyectos	30
Tareas	10
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. L. Bass, P. Clements, and R. Kazman (1998), Software Architecture in Practice, Addison Wesley Inc., New York.
2. Shaw, Mary & Garlan, David. Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall; 1 edition (April 2, 1996). ISBN: 0131829572
3. M.E. Fayad, D.C. Schmidt, and R.E. Johnson (1999), Building Application Frameworks: Object-Oriented Foundations of Framework Design, John Wiley & Sons, Inc., New York.

4. Shalloway, Alan and James R. Trott. Design Patterns Explained: A New Perspective on Object-Oriented Design. Addison-Wesley Pub Co; 1st edition (July 9, 2001) ISBN: 0201715945
5. Gamma, Erich, , Richard Helm, Ralph Johnson, John Vlissides. Design Patterns. Addison-Wesley Pub Co; 1st edition (January 15, 1995) ISBN: 0201633612
6. Microsoft Corporation. Enterprise Solution Patterns Using Microsoft .NET. Microsoft Press; (September 24, 2003) ISBN: 0735618399

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.



## ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

Horas: 45 T 30P  
Créditos: 8  
Clave IS-07

### OBJETIVO:

Utilizar los estándares, métodos, técnicas y herramientas para la validación y verificación de sistemas de software desde sus fases iniciales hasta la liberación y entrega de los mismos.

### CONTENIDO.

---

1. Revisiones.
2. Pruebas (proceso, planes de pruebas, diseño de pruebas, casos de pruebas, pruebas de caja negra, caja blanca, unidad, regresión, integración, betas, liberación, desarrollo conducido por pruebas).
3. Aseguramiento de la calidad (modelos, estándares y procesos)
4. Métricas

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas, valoraciones y auditorías a procesos de desarrollo de software.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	70
Trabajos de investigación	10
Tareas y Proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Galin, Daniel. Software Quality Assurance : From Theory to Implementation. 2003.
2. Ginac, Frank P. Customer Oriented Software Quality Assurance. 1997.
3. Lewis, William E. Software Testing and Continuous Quality Improvement. 2000.
4. Kan, Stephen H. Metrics and Models in Software Quality Engineering (2nd Edition). 2002

5. Buerstein, Llène. Practical Software Testing. Springer Verlag. 2003.
6. O'Reagan, Gerard. A practical approach to Software Quality. Springer Verlag. 2002.

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## MÉTODOS FORMALES EN ESPECIFICACIÓN Y DISEÑO DE SOFTWARE

Horas: 60 T 15 P  
Créditos: 9  
Clave IS-08

### OBJETIVO:

Utilizar los conceptos matemáticos para el modelado de sistemas de software y la aplicación de estos modelos para el análisis de las propiedades y la verificación del diseño de estos sistemas.

### CONTENIDO

---

1. Teoría de conjuntos y lógica.
2. Cálculo de predicados.
3. Gramáticas, máquinas de estado finito, redes de petri, grafos, árboles.
4. Relaciones.
5. Álgebra relacional.
6. Conceptos de especificación formal.
7. Especificaciones y modelos algebraicos.
8. Notación Z, VDM u OCL.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Frausto, Juan; Sánchez, Gildardo. Fundamentos de Lógica Computacional. Trillas, 2000. ISBN 9682461006.
2. Woodcock, Jim & Davies, Jim. Using Z, Specification, Refinement, and Proof. University of Oxford (gratis en internet)

3. Andrew, Harry. Formal Methods, fact file, VDM and Z. John Wiley & Sons 1996. ISBN 0-471-94006-2.
4. Huth, M; Ryan, M. Logic in Computer Science: Modeling and reasoning about systems. Cambridge University Press. 2000
5. Peled. Software Reliability Methods. Springer Verlag. 2001.
6. Warmer, J; Kleppe, A. The Object Constraint Language. Precise Modeling with UML. Addison Wesley. 1999
7. Jacky, Jonathan. The Way of Z: Practical Programming with Formal Methods. Cambridge University Press, 1997, ISBN 0-521-55976-6.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación, Ingeniero de Software, o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.

## ADMINISTRACIÓN DE PROYECTOS I

Horas: 45 T 30P  
Créditos: 8  
Clave IS-09

### OBJETIVO:

Describir los estándares, métodos, técnicas y herramientas para la administración de proyectos de software a través del desarrollo de un producto de software desde su concepción hasta la implantación del mismo.

### CONTENIDO.

1. El proceso visto desde las recomendaciones de los modelos de mejora y capacidades (CMM/I, SPICE, MoProSoft, PMBoK).
2. Conceptos de administración de proyectos.
3. Ciclos de vida.
4. Estimación.
5. La Agenda del Proyecto.
6. Análisis de Riesgos.
7. Administración de requerimientos.
8. Seguimiento del proyecto.
9. Administración de la Configuración del Software.
10. Administración del recurso humano.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, supervisión de proyectos (revisión de productos de software simulando ser cliente y gerente de desarrollo).

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	40
Proyectos	40
Tareas	20
Total	100 puntos

## **BIBLIOGRAFÍA.**

---

1. Steve McConnell. Rapid Development. Microsoft Press; 1st edition (July 2, 1996) ISBN: 1556159005.
2. PMI Guide to the Project Management Body of Knowledge, Project Management Institute 2000.
3. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Continuous Representation. CMU/SEI 2000
4. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Staged Representation. CMU/SEI 2000
5. ISO/IEC 15504, Software Process Improvement and Capability dEtermination.
6. Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition) Addison-Wesley Pub Co. 1995. ISBN: 0201835959.
7. Sommerville, Ian. Ingeniería de Software (5ª o 6ª edición). Addison-Wesley Pub Co;
8. Pressman, Roger. Ingeniería de Software: Un enfoque práctico (5ª o 6ª edición). McGraw-Hill.

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## MÉTRICAS DE SOFTWARE

Horas: 75 T  
Créditos: 10  
Clave IS-10

### OBJETIVO:

Explicar y aplicar estándares, métodos, técnicas y herramientas para llevar a cabo un programa de métricas que permita mejorar la capacidad de desarrollo de software de una unidad de desarrollo de software.

### C O N T E N I D O.

---

1. Fundamentos de medición y experimentación.
2. Mediciones en la Ingeniería de Software.
3. Medición y administración.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, tormenta de ideas, trabajo en equipo, demostración, investigaciones bibliográficas, presentaciones orales y escritas, análisis de casos y artículos, análisis de planes de métricas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	60
Proyectos	30
Tareas	10
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Fenton, Norman & Pfleeger, Shari. Software Metrics A Rigorous and Practical Approach. 2<sup>nd</sup> edition. PWS Publishing Company. 1997.
2. Florac, W.; Park, R. y Carleton. A.; Practical Software Measurement: Measuring for Process Management and Improvement, SEI, 1997.
3. Department of Defense and the US Army. PSM: Practical Software Measurement. <http://www.psmc.com/>

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## ADMINISTRACIÓN DE PROYECTOS II

Horas: 15T 60P  
Créditos: 6  
Clave IS-11

### OBJETIVO:

Describir las estrategias para el control del cambio en el desarrollo de proyectos, para la liberación del proyecto y el bosquejo y la evaluación de los contratos y los acuerdos de propiedad intelectual, continuando con el proyecto iniciado en Administración de Proyectos I.

### CONTENIDO.

---

1. Planeación para el cambio.
2. Contratos de software.
3. Propiedad intelectual.
4. Liberación.

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, supervisión de proyectos (revisión de productos de software simulando ser cliente y gerente de desarrollo).

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	40
Proyectos	40
Tareas	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Steve McConnell. Rapid Development. Microsoft Press; 1st edition (July 2, 1996) ISBN: 1556159005.
2. PMI Guide to the Project Management Body of Knowledge, Project Management Institute 2000.
3. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Continuous Representation. CMU/SEI 2000
4. CMMI for Systems Engineering/Software Engineering, version 1.02 (CMMI-SE/SW, v1.02) Staged Representation. CMU/SEI 2000



5. ISO/IEC 15504, Software Process Improvement and Capability dEtermination.
6. Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition) Addison-Wesley Pub Co. 1995. ISBN: 0201835959.
7. Sommerville, Ian. Ingeniería de Software (5ª o 6ª edición). Addison-Wesley Pub Co;
8. Pressman, Roger. Ingeniería de Software: Un enfoque práctico (5ª o 6ª edición). McGraw-Hill.

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## DISEÑO DE EXPERIMENTOS EN INGENIERÍA DE SOFTWARE

Horas: 60 T 15 P  
Créditos: 9  
Clave IS-12

### OBJETIVO:

Describir la importancia de validar empíricamente las técnicas usadas en ingeniería de software y aplicar diversas técnicas de experimentación en el proceso de desarrollo de software.

### CONTENIDO.

---

1. Introducción a la experimentación en ingeniería de software
2. Principales tipos de diseños y análisis de experimentos
3. Experimentos reales en ingeniería de software
4. Trabajos prácticos

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, trabajo en equipo, demostración, investigaciones bibliográficas.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas y proyectos	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Juristo, Natalia; Moreno, Ana. Basics of Software Engineering Experimentation. Kluwer Academic Publishers, Boston, 2001. ISBN 0-7923-7990-X.
- 2.- Wohlin, Claes; Runeson, Per; Höst, Martin; Ohlsson, Magnus C.; Regnell, Björn and Wesslén, Anders, Experimentation in Software Engineering: An Introduction, Kluwer Academic Publishers, ISBN 0-7923-8682-5, 2000.

**PERFIL ACADÉMICO DEL DOCENTE:** Ingeniero de Software o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## SISTEMAS DISTRIBUIDOS

Horas: 60 T 15 P  
Créditos: 9  
Clave RE-02

### OBJETIVO:

Utilizar los conceptos básicos de la computación distribuida para el diseño e implementación de aplicaciones que requieran compartir recursos y distribuir cargas de trabajo entre nodos de procesamiento.

### CONTENIDO.

---

1. Conceptos básicos de los sistemas distribuidos
2. Comunicación y sincronización de procesos
3. Algorítmica distribuida
4. Programación distribuida

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipo, demostración, investigaciones bibliográficas, desarrollo de prototipos.

Se dará el marco teórico de los sistemas distribuidos con el fin de que se defina un problema y se trabaje en su implementación. Para eso contaremos con una plataforma que permita el manejo de un sistema distribuido.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

- 1.- Colouris, G.F. "Sistemas Distribuidos – Conceptos y Diseño". Addison-Wesley, 2001
- 2.- Andrews, G.A. "Concurrent Programming – Principles and Practice". The Benjamin/Cumming Publishing Company, 1991.
- 3.- Andrew Tanenbaum. Redes de Ordenadores, Prentice Hall. 1999
- 4.- Nancy Lynch. Distributed Algorithms, Ed. Morgan Kaufmann Publishers 1996

- 5.- Valmir Barbosa. An introduction to Distributed Algorithms, The MIT Press 1996
- 6.- A. Goscinski. Distributed Operating Systems. The Logical Design. Addison-Wesley Publishing Company, 1991.
- 7.- Sape Mullender. Distributed Systems. 2a. Edición Addison-Wesley 1993.
- 8.- Andrew S. Tanenbaum. Sistemas Operativos Distribuidos, Ed. Prentice Hall, 1995.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## REDES Y SEGURIDAD DE COMPUTADORAS

Horas: 60 T 15 P  
Créditos: 9  
Clave RE – 03

### OBJETIVOS:

Aplicar las técnicas modernas de seguridad en cómputo, con el fin de administrar los sistemas y las redes para conseguir un entorno seguro de negocio en el trabajo y evaluar los riesgos y vulnerabilidad de la información, desarrollando un sistema de seguridad.

### CONTENIDO.

---

1. Seguridad de Sistemas Operativos
2. Seguridad en equipos de interconexión
3. Diseño de sistemas de seguridad para redes
4. Configuración de servidores seguros
5. Detección y eliminación de vulnerabilidades
6. Criptografía de llave pública
7. Análisis de riesgos informáticos
8. Normas y procedimientos de seguridad
9. Auditoría y respuesta a incidentes de seguridad
10. Casos prácticos de seguridad en cómputo

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, uso de software, trabajo en equipo, demostración de ejemplos prácticos, investigaciones bibliográficas, resolución de problemas de programación en clase y de tareas, desarrollo de prototipos.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	50
Tareas y proyectos	50
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Kurose, J. F & K. W. Ross. Computer Networking – A top Down Approach Featuring the Internet.
2. Caballero, Pino; Hernández, Candelaria. Criptología y Seguridad de la Información. Rama Publishing Company, 2000.

3. P. Pfleeger, Charles; Lawrence Shari. Security in Computing. Prentice Hall, 2002
4. Bishop, Matt. Computer Security: Art and Science. Addison Wesley, 2002.
5. Whitman, Michael; Mattord, Herbert. Principles of Information Security. Thompson Learning, 2002.

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con postgrado y experiencia docente, de investigación o de trabajo en el área.

## SISTEMAS OPERATIVOS

Horas: 75 T  
Créditos: 10  
Clave: SB-03

### OBJETIVO:

Explicar las técnicas empleadas por los sistemas operativos para administrar el procesador, organizar la memoria y el sistema de archivos.

### CONTENIDO.

---

1. Administración del procesador
2. Administración de la memoria
3. Administración de Entrada/Salida
4. Sistema de Archivos
5. Tendencias actuales

**ESTRATEGIAS DE ENSEÑANZA:** Conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, tormenta de ideas, uso de software, trabajo en equipo, análisis y demostración de problemas en distintos escenarios, investigaciones bibliográficas, modificaciones al núcleo de un sistema operativo.

### CRITERIOS DE EVALUACIÓN.

CRITERIO	PUNTUACIÓN
Exámenes	80
Tareas y proyectos	20
Total	100 puntos

### BIBLIOGRAFÍA.

---

1. Deitel, H.M. Sistemas Operativos, 2ª Edición. Addison Wesley. 1993.
2. Tanenbaum, Andrew. Sistemas Operativos Modernos. 2da Ed. Prentice Hall Iberoamericana.1994.
3. Tanenbaum, Andrew. Sistemas Operativos, Diseño e Implementación. Prentice Hall. Software Series. 1993.
4. Stallings, William. Sistemas Operativos. Prentice Hall, 2001.
5. Silberschatz;Galvin;Gagne. Operating systems Concepts. John Wiley. 2003
6. Bovet, Daniel; Cesati, Marco. Understanding the Linux Kernel. O'Reilly & Associates, 2<sup>nd</sup> edition. 2002. ISBN 0596002130

**PERFIL ACADÉMICO DEL DOCENTE:** Licenciado en Ciencias de la Computación o afín, preferentemente con posgrado y experiencia docente, de investigación o de trabajo en el área.



## **XIV. Descripción Sintética de Áreas de Concentración.**

Programación en la WEB
<i>Descripción:</i> Esta área abarca principalmente materias que involucra el desarrollo en el ambiente Web mediante la arquitectura de J2EE. Las materias pueden incluir diversos tópicos y tecnologías para el desarrollo de aplicaciones Web mediante J2EE actuales, tales como: JSP's, Servlets, Procesamiento de información

VideoJuegos
<i>Descripción:</i> Esta área abarca asignaturas que comprenden diversos tópicos para el desarrollo de videojuegos. Algunas de las materias podrían ser: gráficas por computadora, animación 2D/3D, programación 3D, Física para videojuegos, Inteligencia Artificial para videojuegos, visión por computadora, etc. El objetivo central de esta área, es proveer al alumno de conocimientos, técnicas y tecnologías actuales en el desarrollo de videojuegos.

Informática Educativa
<i>Descripción:</i> Esta área ofrece un panorama general de las principales líneas de trabajo en el ámbito de la Informática Educativa, con el propósito de inducir al alumno al desarrollo de aplicaciones en alguna(s) de dichas líneas de trabajo. Las líneas se pueden ser sobre: entornos virtuales de aprendizaje, sistemas multimedia, agentes pedagógicos y en general los Sistemas de Software Educativo.

Mejora del Proceso Software
<p><i>Descripción:</i> Esta área ofrece un panorama general de los principales modelos para la mejora del Proceso Software, así como la forma de gestionar dicho proceso, lo anterior con el propósito de que el alumno implemente proyectos de evaluación y mejora de procesos teniendo en cuenta factores como la tecnología, la organización y el propio negocio.</p>

Sistemas Financieros
<p><i>Descripción:</i> Esta área comprende materias administrativas tales como: Contabilidad, Finanzas, Economía, etc. El objetivo principal de esta área es que el alumno conozca los aspectos generales de los registros contables, estados financieros de una empresa, el conocimiento general del proceso económico, las técnicas de evaluación de proyectos de inversión, y la aplicación de las metodologías empleadas en cada una de estas áreas.</p>

