



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN**

LICENCIATURA EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PROGRAMA DE ASIGNATURA

ACATLÁN

CLAVE:		SEMESTRE: 5 (QUINTO)			
INGENIERÍA DE SOFTWARE					
MODALIDAD (CURSO, TALLER, LABORATORIO, ETC.)	CARACTER	HORAS SEMESTRE	HORA / SEMANA TEÓRICA PRÁCTICA		CRÉDITOS
CURSO	OBLIGATORIO	64	4	0	8 (OCHO)
ASIGNATURA PRECEDENTE SUGERIDA	NINGUNA				
ASIGNATURA CONSECUENTE SUGERIDA	NINGUNA				

OBJETIVO:

EL ALUMNO MANEJARÁ LOS ASPECTOS IMPORTANTES DEL ANÁLISIS, DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN PARA EL DESARROLLO DE SOFTWARE

Número de horas	Unidad 1. INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE
6	<p><i>Objetivo: El alumno reconocerá el origen e importancia de la ingeniería de software</i></p> <p>Temas:</p> <ul style="list-style-type: none"> 1.1 Introducción a la ingeniería de software. 1.2 Mitos y problemas relacionados con el software: el papel del ingeniero de software y su código de ética. 1.3 Procesos de producción de software: cascada, prototipo, incremental, evolutivo y espiral. 1.4 Paradigmas de la ingeniería de software.
Número de horas	Unidad 2. PROCESOS Y ADMINISTRACIÓN DE SOFTWARE
10	<p><i>Objetivo: El alumno reconocerá la importancia de los procesos involucrados en el desarrollo de software y su administración.</i></p> <p>Temas:</p> <ul style="list-style-type: none"> 2.1 Organización de proyectos. 2.2 Planeación y calendarización del proyecto. 2.3 El modelo de madurez. 2.4 Técnicas de estimación de proyectos.

Número de horas	Unidad 3. INGENIERÍA DE REQUERIMIENTOS
10	<p><i>Objetivo: El alumno reconocerá la importancia de realizar un análisis apropiado del problema para clasificar y delimitar de manera apropiada los requerimientos del usuario.</i></p> <p>Temas:</p> <p>3.1 Análisis de problemas: objetivo del análisis y técnicas para obtener información.</p> <p>3.2 Ingeniería de requerimientos: tipos de requerimientos y notaciones de especificación.</p>

Número de horas	Unidad 4. MODELADO Y ESPECIFICACIÓN FORMAL
10	<p><i>Objetivo: El alumno Identificará la mejor forma de modelar el sistema de solución propuesto.</i></p> <p>Temas:</p> <p>4.1 Modelos de contexto y comportamiento.</p> <p>4.2 Diagramas de flujos de datos.</p> <p>4.3 Modelado de datos.</p> <p>4.4 Modelado de objetos.</p> <p>4.5 La especificación formal en el proceso de software.</p>

Número de horas	Unidad 5. DISEÑO DE SOFTWARE
10	<p><i>Objetivo: El alumno realizará de manera apropiada el diseño de la solución propuesta.</i></p> <p>Temas:</p> <p>5.1 Diagramación del diseño.</p> <p>5.2 Importancia de una interfaz en el desarrollo de aplicaciones.</p> <p>5.3 Criterios ergonómicos en las aplicaciones definidos por INRIA (Institut National de Recherche en Informatique et en Automatique).</p> <p>5.4 Ergonomía en la Web.</p>

Número de horas	Unidad 6. TÉCNICAS Y HERRAMIENTAS DE PROGRAMACIÓN
6	<p><i>Objetivo: El alumno identificará y utilizará técnicas apropiadas de programación.</i></p> <p>Temas:</p> <ul style="list-style-type: none"> 6.1 Ingeniería de la usabilidad. 6.2 Útil vs. utilizable. 6.3 Seguridad. 6.4 Reuso de software. 6.5 Ingeniería de software asistido por computadora (C.A.S.E.). 6.6 Buenas prácticas de programación: del diseño al código, el estilo de la programación, depuración y pruebas, rendimiento.

Número de horas	Unidad 7. VERIFICACIÓN, VALIDACIÓN Y PRUEBAS DEL SOFTWARE
6	<p><i>Objetivo: El alumno reconocerá la importancia de desarrollar sistemas de calidad que satisfagan de manera apropiada los requerimientos para los que fueron creados.</i></p> <p>Temas:</p> <ul style="list-style-type: none"> 7.1 Verificación y validación. 7.2 Confiabilidad del software. 7.3 Seguridad del software. 7.4 Inspecciones y revisiones. 7.5 Verificación estadística. 7.6 Verificación formal. 7.7 Pruebas de caja negra. 7.8 Pruebas de integración. 7.9 Pruebas del sistema. 7.10 Métricas técnicas del software.

Número de horas	Unidad 8. CALIDAD DEL SOFTWARE
6	<p><i>Objetivo: El alumno conocerá estándares de calidad para crear proyectos bajo estos criterios.</i></p> <p>Temas:</p> <ul style="list-style-type: none"> 8.1 CMM desarrollado por el software Engineering Institute, en Carnegie Mellon University. 8.2 ISO/IEC TR 15504-2:1998. 8.3 MOPROSOFT (modelo de procesos para la industria de software: MoProSoft).

BIBLIOGRAFÍA BÁSICA

Hamlet y Maybee, *The engineering of software*, Addison Wesley, E.U.A., 2001

Jacobson, I., *Object oriented software engineering. A use .case driven approach*, Addison Wesley, E.U.A., 1992

Kendall y Kendall, *Análisis y diseño de sistemas*, Prentice Hall, México, 1991

Pressman R., *Ingeniería de software, un enfoque práctico*, McGraw Hill, México, 2001

Yourdon, E., *Análisis estructurado moderno*, Prentice Hall, México, 1993

BIBLIOGRAFÍA COMPLEMENTARIA

Burch, J., *Diseño de sistemas de información: teoría y práctica*, Megabyte, México, 1992

Couger, D., *Systems analysis techniques*, John Wiley & Sons, E.U.A., 1980

Cuevas, G., *Ingeniería del software: Práctica de la programación*, Serie Paradigma, México, 1993

Fisher, A., *CASE using software development tools*, Wiley, E.U.A., 1991

Fuentes, A., *El enfoque de sistemas en la solución de problemas la elaboración del modelo conceptual*, Facultad de Ingeniería UNAM, México, 1995

McConnell, S., *Desarrollo y gestión de proyectos informáticos*, McGraw Hill, México, 1996

Naur, P., *Computing a human activity*, Addison Wesley, E.U.A., 1992

Randolph y Posner, *Gerencia de proyectos*, McGraw Hill, México, 1993

Schulmeyer, G., *Zero defect software*, McGraw Hill, E.U.A., 1990

Suárez, R., *Un modelo cualitativo del proceso de solución de problemas. El modelo del diamante*, Facultad de Ingeniería UNAM, México, 1995

Thimbleby, H., *User interface design*, Addison Wesley, E.U.A., 1990

SUGERENCIAS DIDÁCTICAS

- Introducir y exponer los temas y contenidos de las diferentes unidades, con ejemplos claros y sencillos.
- Propiciar la participación de los alumnos a través del empleo de diferentes técnicas de trabajo en grupo.
- Supervisar y guiar a los alumnos cuando los temas sean expuestos y desarrollados por ellos.
- Utilizar algún lenguaje de programación para realizar sistemas computacionales en el que involucren los conocimientos adquiridos en la materia.
- Fomentar en los alumnos la investigación relacionada con la materia, así como tratar temas relevantes que se encuentren en revistas especializadas o en diversas fuentes bibliográficas.

SUGERENCIAS DE EVALUACIÓN

- Exámenes parciales.
- Aplicación práctica de la teoría en el desarrollo de proyectos.
- Trabajos de Investigación.
- Participación en clase.

PERFIL PROFESIOGRÁFICO QUE SE SUGIERE

El profesor que impartirá el curso deberá tener el título de licenciado en Matemáticas Aplicadas y Computación, Ing. en Computación o carreras afines.