



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

PLAN DE ESTUDIOS DE LA LICENCIATURA EN
MATEMÁTICAS APLICADAS Y COMPUTACIÓN

PROGRAMA DE ASIGNATURA



SEMESTRE:5(QUINTO)

Ingeniería de Software

CLAVE:

MODALIDAD	CARÁCTER	TIPO	HORAS AL SEMESTRE	HORAS SEMANA	HORAS TEÓRICAS	HORAS PRÁCTICAS	CRÉDITOS
Curso	Obligatoria	Teórico-Práctica	64	4	2	2	6

ETAPA DE FORMACIÓN	Profundización
CAMPO DE CONOCIMIENTO	Computación

SERIACIÓN	Indicativa
ASIGNATURA(S) ANTECEDENTE	Bases de Datos
ASIGNATURA(S) SUBSECUENTE(S)	Desarrollo Web

Objetivo general: El alumno aplicará los métodos, técnicas y procesos de la ingeniería de software al desarrollo de soluciones automatizadas.

Unidad	Índice Temático	Horas	
		Teóricas	Prácticas
1	Introducción a la ingeniería de software	4	4
2	Administración de proyectos de software	4	4
3	Ingeniería de requerimientos	7	7
4	Modelado	5	5
5	Diseño de software	5	5
6	Verificación, validación y pruebas del software	4	4
7	Estándares y modelos de mejores prácticas	3	3
Total de horas:		32	32
Suma total de horas:		64	

HORAS		UNIDAD	CONTENIDO
T	P		
4	4	1	<p>INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE</p> <p>Objetivo particular: El alumno analizará el origen e importancia de la ingeniería de software.</p> <p>Temas:</p> <ul style="list-style-type: none"> 1.1 Conceptos fundamentales de la ingeniería de software 1.2 Procesos de producción de software (Ciclo de vida del desarrollo de software) <ul style="list-style-type: none"> 1.2.1 Cascada 1.2.2 Prototipo 1.2.3 Incremental 1.2.4 Evolutivo 1.2.5 Espiral 1.2.6 Proceso Unificado Rational (RUP) 1.3 Productos de la Ingeniería de Software <ul style="list-style-type: none"> 1.3.1 Sistema Transaccional 1.3.2 Sistema integrados de gestión 1.3.3 Sistemas de apoyo a la toma de decisiones 1.3.4 Sistemas de comercio electrónico 1.3.5 Sistemas de gestión del conocimiento
4	4	2	<p>ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE</p> <p>Objetivo particular: El alumno aplicará los procesos involucrados en la planificación de proyectos de software.</p> <p>Temas:</p> <ul style="list-style-type: none"> 2.1 Etapas en el desarrollo de proyectos <ul style="list-style-type: none"> 2.1.1 Inicio 2.1.2 Planeación 2.1.3 Ejecución 2.1.4 Control 2.1.5 Cierre 2.2 Áreas a considerar en la administración de un proyecto <ul style="list-style-type: none"> 2.2.1 Alcance 2.2.2 Tiempo: calendarización 2.2.3 Costo 2.2.4 Gestión de la calidad 2.2.5 Recursos humanos 2.2.6 Comunicación 2.2.7 Gestión del riesgo 2.2.8 Abastecimientos 2.2.9 Integración 2.3 Estimación <ul style="list-style-type: none"> 2.3.1 Puntos de función 2.3.2 Casos de uso

7	7	3	<p>INGENIERÍA DE REQUERIMIENTOS</p> <p>Objetivo particular: El alumno aplicará técnicas para la obtención y documentación de los requerimientos del usuario.</p> <p>Temas:</p> <ul style="list-style-type: none"> 3.1 Requerimientos funcionales y no funcionales 3.2 Preparación de la estrategia de recolección de requerimientos 3.3 Técnicas de licitación <ul style="list-style-type: none"> 3.3.1 Entrevista 3.3.2 Cuestionario 3.3.3 Lluvia de ideas 3.3.4 Prototipos 3.3.5 Casos de uso 3.4 Análisis del dominio y reconocimiento de patrones de análisis 3.5 Especificación de requerimientos de software <ul style="list-style-type: none"> 3.5.1 Estructura mínima de un documento de requerimientos 3.5.2 Propiedades deseables en la redacción de los requerimientos 3.5.3 Especificación de casos de uso 3.5.4 Especificación formal 3.6 Métricas de calidad para los requerimientos 3.7 Validación de requerimientos
5	5	4	<p>MODELADO</p> <p>Objetivo particular: El alumno aplicará diferentes herramientas para el modelado de software.</p> <p>Temas:</p> <ul style="list-style-type: none"> 4.1 Modelos de contexto y comportamiento 4.2 Diagrama de flujo de datos 4.3 Diagramas de actividad 4.4 Diagramas de interacción 4.5 Diagramas de clases 4.6 Diagramas de estados
5	5	5	<p>DISEÑO DE SOFTWARE</p> <p>Objetivo particular: El alumno aplicará diferentes herramientas para el diseño de software.</p> <p>Temas:</p> <ul style="list-style-type: none"> 5.1 Diseño arquitectónico <ul style="list-style-type: none"> 5.1.1 Patrones arquitectónicos 5.1.2 Arquitectura cliente servidor 5.1.3 Arquitectura en capas 5.2 Diseño de interfaz 5.3 Diseño de componentes <ul style="list-style-type: none"> 5.3.1 Principios del diseño de componentes 5.3.2 Diagrama de componentes 5.4 Diseño de la implementación: diagrama de despliegue

4	4	6	VERIFICACIÓN, VALIDACIÓN Y PRUEBAS DEL SOFTWARE Objetivo particular: El alumno aplicará estrategias de verificación, validación y prueba en la producción de software. Temas: 6.1 Verificación y validación 6.2 Confiabilidad y seguridad del software 6.3 Pruebas de caja blanca <ul style="list-style-type: none"> 6.3.1 Prueba de ruta básica 6.3.2 Prueba de condición 6.3.3 Prueba de ciclo 6.3.4 Prueba de flujo de datos 6.4 Pruebas de caja negra <ul style="list-style-type: none"> 6.4.1 Prueba de clases de equivalencia 6.4.2 Prueba pairwise 6.4.3 Prueba aleatoria 6.5 Pruebas de unidad e integración 6.6 Pruebas del sistema <ul style="list-style-type: none"> 6.6.1 Seguridad 6.6.2 Resistencia (estrés) 6.6.3 Desempeño 6.7 Documentación de las pruebas de software 6.8 Métricas de la calidad del software
3	3	7	ESTÁNDARES Y MODELOS DE MEJORES PRÁCTICAS Objetivo particular: El alumno identificará estándares de calidad y modelos de mejores prácticas para el desarrollo y la implementación de proyectos de software. Temas: 7.1 CMMI 7.2 MoProsoft 7.3 Modelos paracurriculares NYCE 7.4 Métodos ágiles 7.5 COBIT 7.6 ITIL

Referencias básicas:

- Hamlet y Maybee. (2001). *The engineering of software*. E.U.A: Addison Wesley.
- Jacobson, I. (1992). *Object oriented software engineering. A use case driven approach*. E.U.A.: Addison Wesley.
- Pressman, R. (2010). *Ingeniería de software, un enfoque práctico* (7 ed). México: McGraw Hill.
- Sommerville, I. (2011). *Software engineering*. E.U.A.: Addison Wesley Iberoamericana.

Referencias complementarias:

- Burch, J. (1992). *Diseño de sistemas de información: teoría y práctica*. México: Megabyte.
- Cox y Novobilski. (1991). *Object-Oriented Programming: An evolutionary approach*. E.U.A.: Addison Wesley.
- Cuevas, G. (1993). *Ingeniería del software: Práctica de la programación*. México: Serie Paradigma.
- Fuentes, A. (1995). *El enfoque de sistemas en la solución de problemas la elaboración del modelo conceptual*. México: Facultad de Ingeniería UNAM.
- Kendall y Kendall. (1991). *Análisis y diseño de sistemas*. México: Prentice Hall.
- McConnell, S. (1996). *Desarrollo y gestión de proyectos informáticos*. México: McGraw Hill
- Randolph y Posner. (1993). *Gerencia de proyectos*. México: McGraw Hill.
- Suárez, R. (1995). *Un modelo cualitativo del proceso de solución de problemas. El modelo del diamante*. México: Facultad de Ingeniería UNAM.
- Taylor, D. (1991). *Object-Oriented Technology: A manager's Guide*. E.U.A: Addison Wesley.
- Thimbleby, H. (1990). *User interface design*. E.U.A.: Addison Wesley.

Sugerencias didácticas:	Sugerencias de evaluación del aprendizaje:
Analizar y producir textos. Utilizar tecnologías multimedia. Realizar ejercicios dentro y fuera de clase. Estudiar casos prácticos. Prácticas de campo Instrumentar técnicas didácticas como exposición audiovisual, exposición oral, interrogatorio y técnicas grupales de trabajo colaborativo. Realizar visitas de observación. Usar recursos didácticos en línea.	Examen final oral o escrito Exámenes parciales Informes de prácticas Informes de investigación Participación en clase Rúbricas Solución de ejercicios Trabajos y tareas

Perfil Profesiográfico: El profesor que imparta la asignatura deberá tener el título de licenciado en Matemáticas Aplicadas y Computación o carrera afín, con experiencia profesional y docente en la materia, contar con actualización en el área y preferentemente tener estudios de posgrado.