

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Ingeniería de Software
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	SCD-1011
(Créditos) SATCA ¹	2 -3-5

2.- PRESENTACIÓN

Caracterización de la asignatura.

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las competencias profesionales para aplicar métodos y técnicas que permitan desarrollar soluciones de software, conforme a las normas organizacionales de manejo y seguridad de la información, utilizando tecnologías emergentes.

La importancia de esta asignatura, es que permite al alumno abordar todas las fases del ciclo de vida del desarrollo de cualquier tipo de software, aplicables a una infinidad de áreas.

Esta asignatura, es la aplicación práctica del conocimiento científico, a través de los métodos y técnicas adecuados, para el desarrollo de software.

La disciplina de Ingeniería de Software se relaciona con materias precedentes como: Fundamentos De Programación, Programación Orientada A Objetos, Estructura De Datos, Fundamentos De Ingeniería De Software, Tópicos Avanzados De Programación, Sistemas Operativos, Arquitectura De Computadoras, Telecomunicaciones.

Posteriores: Gestión de Proyectos de Software, Programación Web.

Requiere de competencias previas como: Manejo de un lenguaje de modelado, dominio en el uso de herramientas CASE, uso de algún Sistema Manejador de Bases de Datos, dominio de algún lenguaje de programación orientado a objetos, identificación de las etapas del ciclo de desarrollo de sistemas y de las diferentes plataformas operativas.

Intención didáctica.

La asignatura debe ser teórico – práctico y capaz de desarrollar en el estudiante la habilidad para la aplicación de las diferentes técnicas para el desarrollo de software, considerando siempre los principios de la ingeniería de software, para lo cual se

¹ Sistema de asignación y transferencia de créditos académicos

organiza el temario en cuatro bloques.

En el bloque uno, evolución del modelado de negocios, se muestra el apoyo para detectar oportunidades, anticipar riesgos innecesarios, obtener financiamiento y revisar periódicamente los avances para fijar nuevas metas, de manera que la evolución del modelado de negocios corresponde a los elementos que deben considerarse antes de iniciar un proyecto e involucra aspectos como espíritu emprendedor, creatividad y trabajo en equipo.

El bloque dos permitirá investigar y aplicar las diferentes metodologías, de acuerdo al ámbito del software, contemplando las actividades relativas a la especificación del software, el desarrollo, la validación y la evolución.

En el bloque tres se describe la arquitectura del software, como el proceso que comprende modelos arquitectónicos, para identificar subsistemas y establecer un marco de trabajo para su control y comunicación.

En el bloque cuatro se pretende identificar la seguridad como un requerimiento crítico, el cual debe ser considerado durante el proceso de desarrollo y al final de éste para que se garantice continuidad en el soporte al mismo, que permita minimizar los riesgos al aplicar pruebas de vulnerabilidad potencial.

3.- COMPETENCIAS A DESARROLLAR

Competencias específicas:

Desarrollar soluciones de software, considerando los aspectos del modelo de negocios, mediante la aplicación de la metodología adecuada a la naturaleza del problema.

Competencias genéricas:

Competencias instrumentales

- Capacidad de análisis y síntesis
- Capacidad de organizar y planificar
- Comunicación oral y escrita
- Habilidad para buscar y analizar información proveniente de fuentes diversas
- Solución de problemas
- Toma de decisiones.

Competencias interpersonales

- Capacidad crítica y autocrítica
- Trabajo en equipo
- Habilidades interpersonales

Competencias sistémicas

- Capacidad de aplicar los conocimientos en la práctica
- Habilidades de investigación

	<ul style="list-style-type: none"> • Capacidad de aprender • Capacidad de generar nuevas ideas (creatividad) • Habilidad para trabajar en forma autónoma • Búsqueda del logro
--	---

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Saltillo Fecha del 5 al 9 de Octubre de 2009	Representantes de los Institutos Tecnológicos de: I.T. de Ciudad Valles, I.T. de Matamoros I.T. de Mérida I.T. de Orizaba I.T.S. del Sur del Estado de Yucatán.	Reunión nacional de Diseño e innovación curricular de la carrera de Ingeniería en sistemas computacionales
I.T. de Ciudad Valles I.T. de Matamoros I.T. de Mérida I.T. de Orizaba I.T. de Toluca I.T.S. del Sur del Estado de Yucatán. Fecha del 12 de Octubre/2009 Al 15 de Febrero/2010	Representantes de las Academias de los Institutos Tecnológicos.	Análisis, enriquecimiento y elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico Superior de Poza Rica, Veracruz. Fecha del 22 al 26 de Febrero/2010	Representantes de los Institutos Tecnológicos participantes: I.T. de Ciudad Valles I.T. de Matamoros I.T. de Mérida I.T. de Orizaba I.T.S. del Sur del Estado de Yucatán.	Reunión nacional de consolidación de la carrera de Ingeniería en Sistemas Computacionales.

5.- OBJETIVO(S) GENERAL(ES) DEL CURSO (competencias específicas a desarrollar en el curso)

Desarrollar soluciones de software, considerando los aspectos del modelo de negocios, mediante la aplicación de la metodología adecuada a la naturaleza del problema.

6.- COMPETENCIAS PREVIAS

- Aplica modelos, técnicas y herramientas para cada una de las etapas del ciclo de vida de desarrollo de software
- Utiliza un Sistema Manejador de Bases de Datos
- Utiliza algún lenguaje de programación orientado a objetos
- Identifica diferentes plataformas operativas

7.- TEMARIO

Unidad	Temas	Subtemas
1.	Modelado de Negocios	1.1 Evolución del Modelado de Negocios 1.2 Componentes del Modelado de Negocios 1.3 Orientaciones del Modelado de Negocio 1.4 BPMN en el Modelado del Negocio
2.	Metodologías de Desarrollo	2.1 Metodologías clásicas 2.1.1 Cascada 2.1.2 Incremental 2.1.3 Evolutivo 2.1.4 Espiral 2.1.5 Prototipos 2.1.6 Desarrollo basado en componentes 2.2 Otras Metodologías 2.2.1 Ganar-ganar 2.2.2 Proceso Unificado (UP) 2.2.3 Ingeniería Web 2.2.4 Metodologías Ágiles 2.2.5 Metodologías emergentes
3.	Arquitecturas de software	2.3 Reingeniería 3.1 Descomposición modular 3.2 Patrones de Diseño 3.3 Arquitectura de dominio específico 3.4 Diseño de software de arquitectura multiprocesador 3.5 Diseño de software de arquitectura Cliente - Servidor 3.6 Diseño de software de arquitectura distribuida 3.7 Diseño de software de arquitectura de tiempo real

4.	Seguridad en Ingeniería de Software	4.1 Seguridad de software 4.2 Seguridad en el ciclo de desarrollo del software 4.3 Confiabilidad del software 4.4 Ingeniería de seguridad
----	-------------------------------------	--

8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)

El profesor debe:

Ser conocedor de la disciplina que está bajo su responsabilidad, conocer su origen y desarrollo histórico para considerar este conocimiento al abordar los temas. Desarrollar la capacidad para coordinar y trabajar en equipo; orientar el trabajo del estudiante y desarrollar en él la autonomía, el trabajo cooperativo y la toma de decisiones. Mostrar flexibilidad en el seguimiento del proceso formativo y propiciar la interacción entre los estudiantes. Tomar en cuenta el conocimiento de los estudiantes como punto de partida y como obstáculo para la construcción de nuevos conocimientos.

- Propiciar actividades de metacognición. Ante la ejecución de una actividad, señalar o identificar el tipo de proceso intelectual que se realizó: una identificación de patrones, un análisis, una síntesis, la creación de una heurística, etc. Al principio lo hará el profesor, luego será el alumno quien lo identifique. Ejemplos: reconocer la metodología para el desarrollo de un sistema de control escolar: reconocimiento de paradigmas, elaboración de un software a partir de una serie de observaciones, producto de un caso real.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes. Ejemplo: buscar y contrastar definiciones de las metodologías identificando puntos de coincidencia entre unas y otras.

- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes. Ejemplo: al socializar los resultados de las investigaciones y las experiencias prácticas solicitadas como trabajo extra clase.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional. Ejemplos: el proyecto que se realizará durante el curso.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios a las que ésta da soporte para desarrollar una visión interdisciplinaria en el estudiante.
- Propiciar el desarrollo de capacidades intelectuales relacionadas con la lectura, la escritura y la expresión oral. Ejemplos: trabajar las actividades prácticas a través de guías escritas, redactar reportes e informes de las actividades realizadas en el centro de cómputo, exponer al grupo las conclusiones obtenidas durante las observaciones.
- Facilitar el contacto directo con materiales e instrumentos, al llevar a cabo actividades prácticas, para contribuir a la formación de las competencias para el trabajo experimental como: identificación, manejo y control de variables y datos relevantes, planteamiento de hipótesis, trabajo en equipo.
- Propiciar el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, que encaminen hacia la investigación.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Relacionar los contenidos de la asignatura con el cuidado del medio ambiente; así como con las prácticas de una agricultura sustentable.
- Cuando los temas lo requieran, utilizar medios audiovisuales para una mejor comprensión del estudiante.

- Propiciar el uso de las nuevas tecnologías en el desarrollo de la asignatura (Lenguaje de programación, sistema operativo, manejador de bases de datos, procesador de texto, hoja de cálculo, graficador, Internet, etc.).

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación de la asignatura debe ser continua y formativa por lo que se hará con base en el siguiente desempeño para cada una de las actividades:

- Solución de casos prácticos solicitados durante las actividades, así como sus conclusiones de forma escrita.
- Reportes de investigación de campo.
- Reportes de prácticas
- Ejercicios realizados.
- Tareas
- Exposición
- Participación en clase
- Proyecto
- Exámenes teóricos y/o prácticos.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Modelado de Negocios

Competencia específica a desarrollar	Actividades de Aprendizaje
Desarrollar la habilidad para generar propuestas de modelos de negocios de proyectos de software.	<ul style="list-style-type: none"> • Investigar los conceptos de evolución y componentes del Modelado de Negocios • Discutir sobre la importancia (ventajas y desventajas) del modelado de negocios en el desarrollo de un proyecto de software. • Investigar en qué se basa el Modelado BPMN y, a partir del análisis de un caso práctico real, elaborar la propuesta del modelo de negocios del proyecto de software.

Unidad 2: Metodologías de desarrollo

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar y aplicar la metodología adecuada para el desarrollo de diferentes productos de software.	<ul style="list-style-type: none">• Investigar qué caracteriza a cada una de las metodologías planteadas en este módulo. Discutir y formalizar grupalmente lo investigado.• Identificar las metodologías que aplican las diferentes compañías productoras de software• Justificar la elección de alguna de las metodologías para el desarrollo de un proyecto de software.

Unidad 3: Arquitecturas de software

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar y establecer los lineamientos formales para el desarrollo de aplicaciones robustas, tomando decisiones técnicas relacionadas con la arquitectura de la aplicación como guía para el diseño de la misma.	<ul style="list-style-type: none">▪ Comparar y contrastar diferentes conceptos de arquitectura de software▪ Investigar la evolución de las arquitecturas▪ Identificar las funciones de un arquitecto de software▪ Identificar las características de las diferentes arquitecturas de software, así como todos los elementos que requieren unir para desarrollarlo.▪ El alumno deberá investigar ejemplos de software, que utilicen empresas de la región, que cumplan con las características de cada una de las diferentes arquitecturas.

Unidad 4: Seguridad en ingeniería de software

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar los riesgos posibles que puede enfrentar durante el proceso de desarrollo del software y aplicar medidas de seguridad para minimizarlos.	<ul style="list-style-type: none">▪ Investigar el concepto de seguridad enfocado a los productos de software y discutirlo en grupo.▪ Investigar el concepto de riesgos y elaborar un mapa mental.▪ Identificar los riesgos que se pueden presentar en el proyecto de desarrollo de software, exponerlo al grupo para

	<p>enriquecer con opiniones.</p> <ul style="list-style-type: none"> ▪ Mostrar soluciones para los riesgos que se presenten en el desarrollo del proyecto de software y analizarlo en grupo. ▪ Identificar las medidas de seguridad que refiere la ingeniería de software en el proyecto de software a desarrollar. ▪ Identificar los puntos que permiten establecer la confiabilidad del software ▪ Establecer la diferencia entre seguridad y fiabilidad del software. ▪ Investigar los algoritmos de encriptación y elaborar un programa implementando alguno de éstos. ▪ Elaborar un diagrama de contexto para la implementación de firmas digitales.
--	--

11.- FUENTES DE INFORMACIÓN

1. Borrero, L. Tecnologías de la Información En Internet. Editorial Norma. Colombia. 2003.
2. Howard, M. Puntos críticos sobre seguridad de software. McGraw-Hill interamericana. España. 2006.
3. Laudon, K.C. Sistemas de Información Gerencial. Pearson Educación. México, 2004.
4. Minguet M. J. M. La calidad del software y su medida. Editorial CERASA. Madrid, España. 2003.
5. Pressman, R. S. *Ingeniería del Software* – Un enfoque práctico. 5ta. Edición. McGraw Hill. Madrid, España. 2002.
6. Weitzenfeld, A. Ingeniería de software orientada a objetos. Cengage learning editores. México. 2005.

12.- PRÁCTICAS PROPUESTAS

1. Estudio de un caso práctico, utilizando una herramienta CASE para flujo-diagramación con la notación BPMN.
2. Elaborar para una empresa real un modelado de negocios.

3. Realizar una investigación sobre las metodologías de desarrollo mas comunes que aplican las organizaciones productoras de software.
4. Justificar la selección de una metodología adecuada para aplicarse en el desarrollo de un proyecto de software, cumpliendo con cada una de las fases que lo conforman, entregando un prototipo del mismo.
5. Realizar visitas a empresas dedicadas al desarrollo de software
6. Desarrollar una mesa de debates, en la cual se cuente con la presencia de un experto en desarrollo, a fin de conocer las experiencias con respecto a los procesos de ingeniería de software.