

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura :	Ingeniería de Software
Carrera :	Ingeniería en Tecnologías de la Información y Comunicaciones
Clave de la asignatura :	TIC-1014
SATCA ¹	2-2-4

2.- PRESENTACIÓN

Caracterización de la asignatura.

Esta asignatura aporta al perfil del Ingeniero en Tecnologías de Información y Comunicaciones la capacidad para:

- Diseñar, desarrollar y mantener sistemas de bases de datos asegurando la integridad, disponibilidad y confidencialidad de la información almacenada.
- Desarrollar e implementar sistemas de información para el control y la toma de decisiones utilizando metodologías basadas en estándares internacionales.
- Utilizar tecnologías y herramientas actuales y emergentes acordes a las necesidades del entorno.
- Administrar proyectos que involucren tecnologías de la información y comunicaciones en las organizaciones conforme a requerimientos establecidos.
- Integrar las diferentes arquitecturas de hardware y administrar plataformas de software para incrementar la productividad en las organizaciones.

Esta materia es la introducción a la ingeniería de software, sirviendo como base para la materia Taller de Ingeniería de Software y materias propias del perfil de la carrera.

Intención didáctica.

El estudio de esta materia se divide en 5 unidades donde el estudiante podrá conocer que es la Ingeniería de Software, los sistemas de información, los modelos de la ingeniería de software, como elaborar la planificación y el análisis de proyectos de software, asegurando la calidad del software.

En la primera unidad, el estudiante conocerá los fundamentos de la Ingeniería de Software y los sistemas de información.

En la segunda unidad, se conocerán los distintos tipos modelos de la Ingeniería de Software, además el estudiante podrá seleccionar el modelo más adecuado para el desarrollo del software.

En la tercera unidad, se abordan temas para que el estudiante elabore una excelente planificación del proyecto del software, así como el uso o aplicación de las diferentes métricas de software.

En la unidad cuatro, se tratan temas relacionados con el modelado, construcción, implementación y mantenimiento del software, para que el estudiante elabore al análisis del

¹ Sistema de Asignación y Transferencia de Créditos Académicos

proyecto de software.

En la quinta unidad, se abordan los temas de calidad del software, para que el estudiante se asegure de desarrollar software de calidad.

Es importante que el profesor ubique al estudiante en el panorama general en lo que será su ocupación profesional y los aportes que se dan a la sociedad.

3.- COMPETENCIAS A DESARROLLAR

<p>Competencias específicas:</p> <ul style="list-style-type: none">▪ Aplicar técnicas de adquisición de datos (entrevistas, cuestionarios, sondeos, entre otros) para el desarrollo de su proyecto de software.▪ Aplicar los elementos y conceptos integrados en los procesos de desarrollo de software para la documentación adecuada de su proyecto de software.	<p>Competencias genéricas:</p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Conocimientos básicos de la carrera.• Comunicación oral y escrita.• Habilidades básicas de manejo de la computadora.• Habilidad para buscar y analizar información proveniente de fuentes diversas.• Solución de problemas.• Toma de decisiones. <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica.• Trabajo en equipo.• Habilidades interpersonales. <p><u>Competencias sistémicas</u></p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos en la práctica.• Habilidades de investigación.• 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	Evento
<p>Instituto Tecnológico Superior de Puerto Vallarta del 10 al 14 de agosto de 2009.</p>	<p>Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Superior de Centla, Chetumal, Ciudad Cuauhtémoc, Ciudad Madero, Comitán, Delicias, León, Superior de Misantla, Pachuca, Pinotepa, Puebla, Superior de Puerto Vallarta, Roque, Tepic, Tijuana, Tuxtla Gutiérrez y Villahermosa.</p>	<p>Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>
<p>Desarrollo de Programas en Competencias Profesionales por los Institutos Tecnológicos del 17 de agosto de 2009 al 21 de mayo de 2010.</p>	<p>Academias de Ingeniería en Tecnologías de la Información y Comunicaciones de los Institutos Tecnológicos de: Apizaco, Puebla, Villahermosa y Superior de Misantla.</p>	<p>Elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>
<p>Instituto Tecnológico de Villahermosa del 24 al 28 de mayo de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Superior de Centla, Chetumal, León, Pachuca, Puebla, Roque, Tepic, Tuxtla Gutiérrez y Villahermosa.</p>	<p>Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de la Carrera de Ingeniería en Tecnologías de la Información y Comunicaciones.</p>

5.- OBJETIVO GENERAL DEL CURSO

Aplicar técnicas de adquisición de datos (entrevistas, cuestionarios, sondeos, entre otros) para el desarrollo de su proyecto de software.

Aplicar los elementos y conceptos integrados en los procesos de desarrollo de software para la documentación adecuada de su proyecto de software.

6.- COMPETENCIAS PREVIAS

- Construir diagramas de tiempo para determinar ruta(s) crítica(s) en la administración de un proyecto.
- Aplicar técnicas de modelado para la solución de problemas.
- Aplicar la sintaxis de un lenguaje orientado a objetos.
- Aplicar un lenguaje orientado a objetos para la solución de problemas.
- Identificar los tipos de organización así como sus estructuras y funciones correspondientes.
- Identificar y analizar necesidades de información para su representación, tratamiento y automatización para la toma de decisiones.
- Diseñar esquemas de bases de datos para generar soluciones al tratamiento de información.

7.- TEMARIO

Unidad	Temas	Subtemas
1.	Introducción a la ingeniería del software y sistemas de información	1.1. Conceptos de Ingeniería del Software: mitos, paradigma, ingeniería de software, calidad, proceso, método, herramienta, espectro de gestión. 1.2. La importancia de la ingeniería del software. 1.3. Historia de la Ingeniería del Software. 1.4. Los sistemas de información: concepto, características, estructuras, procesos, clasificación, ERP's, CRM, SCM.
2.	Modelos de la ingeniería del software	2.1. Modelo de capacidad de madurez. 2.2. Marco de trabajo para el proceso. 2.3. Modelos de la ingeniería del software: modelo de cascada, modelo de prototipos, modelo de espiral, modelo de Proceso Unificado Racional (RUP). 2.4. Tendencias modernas de modelos de la ingeniería del software.
3.	Planificación del proyecto de software	3.1. Aplicación de herramientas para estimación de tiempos y costos de desarrollo de software: GANTT, PERT/CPM, uso de software para la estimación de tiempos y costos. 3.2. Ámbito del software: recursos humanos, recursos de software reutilizables, recursos del entorno.

		3.3. Análisis y gestión del riesgo: estrategias, identificación, proyección, refinamiento, reducción, supervisión y gestión del riesgo.
4.	Análisis del proyecto de software	<p>4.1. Modelado: análisis, diseño, documentación.</p> <p>4.2. Construcción: codificación, pruebas y evaluación, manual del usuario, manual técnico.</p> <p>4.3. Medida, métrica e indicador.</p> <p>4.4. Tipos de métricas: métricas de proceso, métricas de proyecto, métricas orientadas a punto de función, métricas orientadas al tamaño, métricas para la calidad del software.</p> <p>4.5. Implementación y mantenimiento: entrega, retroalimentación del cliente.</p>
5.	Calidad del software	<p>5.1. Definición de calidad y calidad del software.</p> <p>5.2. Importancia de la calidad.</p> <p>5.3. La calidad y la globalización.</p> <p>5.4. Aseguramiento de la calidad del software (SQA): definición y propósito del SQA, problemas que resuelve el SQA, roles y responsabilidades de los equipos de desarrollo, habilidades y capacidades del personal del SQA, Actividades del SQA.</p> <p>5.5. Derecho informático aplicado al software: piratería y falsificación, autoría y creación, contratos y licencias.</p>

8.- SUGERENCIAS DIDÁCTICAS

El docente debe:

- Propiciar el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, tales como: observación, identificación manejo y control de de variables y datos relevantes, planteamiento de hipótesis, de trabajo en equipo.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Propiciar el uso adecuado de conceptos, y de terminología científico-tecnológica.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.
- En coordinación con los estudiantes elaborar instructivos, demostraciones, manuales o cualquier material didáctico que auxilie la impartición de la asignatura.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Reportes que serán entregados en tiempo y forma, cumpliendo las reglas gramaticales del idioma.
- Exposiciones por parte del estudiante.
- Participación activa y crítica en clase.
- Resultados obtenidos en los exámenes de conocimientos que se apliquen.
- Prácticas de laboratorio.
- Asistencia a asesorías para desarrollo de proyecto.
- Lineamientos para la elaboración de trabajos, tareas, proyectos, mapas, tablas comparativas entre otros: portada, índice, desarrollo, análisis, conclusiones y referencias.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción a la ingeniería del software y sistemas de información

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Identificar los elementos que integran la ingeniería de software y los sistemas de información.	<ul style="list-style-type: none">• Realizar una síntesis sobre las características, mitos, elementos y capas de la ingeniería de software.• Investigar en diferentes fuentes de información la importancia e historia de la ingeniería del software.• Formar mesas de discusión de los temas investigados con la finalidad de enriquecer el conocimiento.• Realizar una síntesis sobre el concepto, las características, y elementos de los sistemas de información.• Discutir en clase la importancia que tiene los sistemas de información para la toma de decisiones.• Realizar una investigación en diferentes fuentes sobre los elementos de los sistemas de información.• Visitar una empresa donde investigue los roles de los diferentes actores en los sistemas de información.• Discutir en clase los resultados de su investigación.

Unidad 2: Modelos de la ingeniería del software

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Identificar los diferentes modelos de proceso que pueden aplicarse en el desarrollo de software.	<ul style="list-style-type: none">• Discutir, en equipo, las características que diferencian a cada uno de los modelos de desarrollo de proceso de software: modelo de cascada, modelo de espiral, modelo

	<p>incremental, proceso de desarrollo unificado, proceso software personal.</p> <ul style="list-style-type: none"> • Exponer en clase por equipos las características, actividades, ventajas, desventajas, recomendaciones de uso de un modelo de desarrollo de software. • Determinar el modelo más adecuado para un tipo de sistema.
--	--

Unidad 3: Planificación del proyecto de software

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Planificar el desarrollo de un proyecto de software.	<ul style="list-style-type: none"> • Planificar el tiempo requerido para las actividades involucradas en el desarrollo del proyecto, así como el protocolo para desarrollar la gestión • Realizar el estudio del costo beneficio para evaluar la viabilidad del proyecto desde el punto de vista económico. • Realizar un análisis del entorno en el cual se planea implantar el proyecto de software para determinar la viabilidad técnica y operativa. • Realizar un análisis de los riesgos que tendría el proyecto de software.

Unidad 4: Análisis del proyecto de software

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Sintetizar los resultados del análisis. Identificar y aplicar métricas estandarizadas para la evaluación del análisis. Identificar técnicas para la implantación y mantenimiento del software.</p>	<ul style="list-style-type: none"> • Ilustrar con el caso práctico las fases que integran la propuesta del análisis del sistema. • Presentar la propuesta del proyecto final. • Elegir las métricas a aplicar en el proyecto planteado. • Medir los requerimientos del proyecto de acuerdo a las métricas seleccionadas. • Proporcionar ejemplos resueltos de aplicación de las diferentes métricas. • Realizar una síntesis sobre las técnicas para la implantación y el mantenimiento del software. • Discutir, en equipo, las características que diferencian a cada uno de las técnicas. • Realizar una síntesis sobre el mantenimiento del software, discutir, en equipo, las características del mantenimiento de software.

Unidad 5: Calidad del software

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Identificar la importancia de la calidad para el desarrollo de software. Identificar la relación entre ingeniería de software y el aseguramiento de la calidad de software.</p>	<ul style="list-style-type: none">• Realizar un análisis comparativo de las definiciones de calidad en distintas fuentes de información y elaborar mediante una lluvia de ideas una definición propia.• Debatir sobre la importancia de la calidad en la vida y en el entorno.• Investigar qué empresas han implementado en sus procesos de desarrollo de software estándares de calidad.• Investigar en diferentes fuentes de información, la definición y propósito del SQA y elaborar un documento.• Realizar ejercicios donde aplique las actividades del SQA aplicando los métodos y herramientas vistas en clase.

11.- FUENTES DE INFORMACIÓN

1. Braude. *Ingeniería del Software*. Ed. Alfa-Omega. Primera edición. México. 2003.
2. Presuman, Roger S. *Ingeniería del Software*. Ed. McGraw-Hill. Sexta edición. México. 2005.
3. Sommerville, Ian. *Ingeniería del Software*. Ed. Pearson. Séptima edición. España. 2005.
4. Weitzenfeld, Alfredo. *Ingeniería del Software*. Ed. Thomson. Primera edición. México. 2004.
5. Gómez, A.; Suárez, C. *Sistemas de Información. Herramientas prácticas para la gestión empresarial*. Ed. Ra-ma. Segunda edición. 2006.

12.- PRÁCTICAS PROPUESTAS

- Utilizar alguna herramienta de software para administración de proyectos y así realizar el estudio de factibilidad de un caso práctico.
- Utilizar técnicas de recolección de requerimientos para identificar y clasificar requerimientos de sistemas.
- Utilizar alguna herramienta de software para el análisis de requerimientos y complementar el análisis y documentación del sistema.
- Utilizar alguna herramienta de software para realizar el prototipo del sistema.
- Medir los requerimientos del proyecto de acuerdo con las diferentes métricas de evaluación del análisis.
- Diseñar un sistema, identificando y aplicando la metodología a la cual se pueda adaptar.
- Transcribir dicho sistema a un lenguaje de modelado.
- Investigar las métricas de evaluación y los estándares que las rigen.
- Recabar la documentación correspondiente a esta etapa de diseño.
- Que los equipos expongan los avances del proyecto en el grupo para que el profesor asesore y marque los ajustes necesarios. De esta manera se comparten ideas y experiencias fomentando que los proyectos al final del curso queden terminados al 100%.