

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	<b>Fundamentos de Ingeniería de Software</b>
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	SCC-1007
(Créditos) SATCA <sup>1</sup>	2-2-4

## 2.- PRESENTACIÓN

### **Caracterización de la asignatura.**

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales los conceptos básicos relacionados con el desarrollo de sistemas, los tipos de modelos para el desarrollo y gestión de software considerando la calidad, lo que permite integrar soluciones computacionales con diferentes tecnologías en diversas áreas.

Es una introducción para la planeación y el proceso de desarrollo de proyectos de software que involucra la comprensión de términos, herramientas, métodos, para crear soluciones informáticas eficientes. Esta diseñada para el logro de competencias específicas, orientada a conocer y aplicar los conceptos básicos de ingeniería de software.

Se relaciona previamente con las asignaturas de programación orientada a objetos y sirve de base para las asignaturas de ingeniería de software y gestión de proyectos.

### **Intención didáctica.**

En la unidad 1, Fundamentos de Ingeniería de Software se brindan los conceptos básicos del papel evolutivo del software para que el estudiante comprenda el contexto del desarrollo del software orientado a objetos.

En la unidad 2 denominada Ingeniería de requisitos orientada a objetos proporciona al estudiante las técnicas y herramientas para la recopilación de la información obteniendo como resultado un modelo de requisitos.

En la unidad 3, Modelo de Análisis orientado a objetos, el estudiante deberá desglosar cada uno de los requisitos y representarlos/traducirlos a una especificación técnica mediante el uso de diagramas tales como: Modelo CRC, Modelo Objeto-Relación, atributos, operaciones y colaboradores.

En la unidad 4, Modelo de Diseño orientado a objetos, el estudiante transformara el modelo de análisis en un modelo de diseño que sirve como un anteproyecto para la

---

<sup>1</sup> Sistema de asignación y transferencia de créditos académicos

construcción del software.

En la unidad 5, Modelo de Implementación orientado a objetos, el estudiante será capaz de tomar los resultados del modelo de diseño para generar el producto de software que se adapte al lenguaje de programación y/o la base de datos según las especificaciones del diseño.

### 3.- COMPETENCIAS A DESARROLLAR

<p><b>Competencias específicas:</b></p> <ul style="list-style-type: none"><li>• Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos.</li><li>• Aplicar modelos, técnicas y herramientas para cada una de las etapas del ciclo de vida de desarrollo de software.</li></ul>	<p><b>Competencias genéricas</b></p> <p><b>1- Competencias instrumentales:</b></p> <ul style="list-style-type: none"><li>• Capacidades cognitivas</li><li>• Capacidades metodológicas para manipular el ambiente</li><li>• Destrezas tecnológicas relacionadas con el uso y manejo de equipo de computo, así como de búsqueda y manejo de información</li><li>• Destrezas lingüística tales como la comunicación oral y escrita o conocimientos de una segunda lengua.</li></ul> <p><b>2-Competencias interpersonales:</b></p> <ul style="list-style-type: none"><li>• Capacidad crítica y autocrítica</li><li>• Trabajo en equipo</li><li>• Habilidades interpersonales</li><li>• Capacidad de trabajar en equipo interdisciplinario</li><li>• Capacidad de comunicarse con profesionales de otras áreas</li><li>• Habilidad para trabajar en un ambiente laboral</li><li>• Compromiso ético</li></ul> <p><b>3-Competencias sistémicas:</b></p> <ul style="list-style-type: none"><li>• Clasificar los modelos de requisitos de ingeniería de software.</li><li>• Aplicar diferentes herramientas para modelar el dominio de la información, describir su función, representar su comportamiento.</li><li>• Seleccionar al menos dos herramientas CASE por etapa de desarrollo.</li></ul>
--	--

#### 4.- HISTORIA DEL PROGRAMA

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

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

- Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos.

- Aplicar modelos, técnicas y herramientas para cada una de las etapas del ciclo de vida de desarrollo de software.

## 6.- COMPETENCIAS PREVIAS

- Utiliza algún lenguaje de programación orientado a objetos
- Aplica razonamiento lógico
- Identifica conceptos básicos de Bases de Datos
- Aplica técnicas de representación algorítmica

## 7.- TEMARIO

Unidad	Temas	Subtemas
1	Fundamentos Ingeniería de software	1.1. Conceptos básicos 1.2. El papel evolutivo del software 1.3. Etapas del desarrollo software 1.4. Clasificación de la tecnología en el desarrollo de software (Tecnología Estructurada y Orientada a Objetos) 1.5. Definición e historia de las herramientas CASE 1.6. Clasificación de las herramientas CASE
2	Ingeniería de requisitos	2.1. Tareas de la Ingeniería de Requisitos 2.2. Técnicas de la Ingeniería de Requisitos 2.3. Modelado de requisitos 2.4. Herramientas CASE para la Ingeniería de requisitos.
3	Modelo de Análisis	3.1. Arquitectura de clases 3.2. Identificación de clases según Estereotipos. 3.3. Clases 3.4. Diagramas de secuencias 3.5. Diccionario de clases según Módulos 3.6. Herramientas CASE para el análisis
4	Modelo de Diseño	4.1. Estrategias de diseño 4.2. Diseño de objetos 4.3. Diseño de sistema 4.4. Revisión del diseño 4.5. Diagramas de secuencias del Diseño. 4.6. Herramientas CASE para el diseño

5	Modelo Implementación	de	5.1. Diagrama de componentes 5.2. Diagrama de despliegue 5.3. Modelos de pruebas
---	--------------------------	----	--

## **8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)**

- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- 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, 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.
- Relacionar los contenidos de la asignatura con el cuidado del medio ambiente; así como con las prácticas de una ingeniería con enfoque sustentable.
- 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.

## 9.- SUGERENCIAS DE EVALUACIÓN

La evaluación de la asignatura se hará con base en siguiente desempeño:

- Solución de casos prácticos solicitados durante las actividades, así como sus conclusiones.
- Reportes de investigación de campo.
- Reportes de prácticas
- Ejercicios realizados.
- Tareas
- Exposición
- Participación en clase
- Proyecto semestral
- Exámenes para comprobar el manejo de aspectos teóricos y prácticos.

## 10.- UNIDADES DE APRENDIZAJE

### Unidad 1: Fundamentos de la Ingeniería de Software

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar la historia del software, así como las características de los paradigmas estructurado y orientado a objetos. Conocer e identificar las herramientas CASE.	<ul style="list-style-type: none"><li>• Investigar los conceptos básicos de la Ingeniería del software generando un glosario.</li><li>• Elaborar una línea del tiempo sobre la historia del software.</li><li>• Diseñar un cuadro comparativo de las diversas etapas del desarrollo de software.</li><li>• Investigar sobre la historia de las herramientas CASE.</li><li>• Elaborar una taxonomía de las herramientas CASE.</li></ul>

### Unidad 2: Ingeniería de requisitos

Competencia específica a desarrollar	Actividades de Aprendizaje
Desarrollar las habilidades para identificar las diferentes técnicas que se aplican para la obtención de requerimientos de software.	<ul style="list-style-type: none"><li>• Elaborar un mapa mental de la Ingeniería de requisitos.</li><li>• Investigar las diferentes tareas que se</li></ul>

	<p>realizan en la ingeniería de requerimientos para la documentación de proyectos de desarrollo.</p> <ul style="list-style-type: none"> <li>• Documentar en un caso de desarrollo las distintas tareas de la ingeniería de requerimientos.</li> <li>• Investigar y documentar sobre las distintas técnicas que se implementan dentro de las tareas de la ingeniería de requerimientos.</li> <li>• Desarrollar y aplicar las distintas técnicas para cada tarea dentro del caso propuesto a desarrollar.</li> <li>• Investigar sobre las aplicaciones del modelado y sus especificaciones.</li> <li>• Aplicar al menos una herramienta CASE para la identificación de requerimientos.</li> </ul>
--	---

### Unidad 3: Modelo de Análisis

<b>Competencia específica a desarrollar</b>	<b>Actividades de Aprendizaje</b>
<p>Identificar a través de un modelo de requisitos la arquitectura de clases que participarán en el diseño del producto.</p>	<ul style="list-style-type: none"> <li>• Investigar los diferentes modelos orientado a objetos como base para la identificación de clases.</li> <li>• Desarrollar casos de uso y modelos CRC que permitan tener una comprensión de la manera en que el sistema se utilizará.</li> <li>• Aplicar el modelo objeto-relación-comportamiento que indique como responderá el sistema OO a eventos.</li> <li>• Aplicar al menos una herramienta CASE para el análisis.</li> <li>• Parte 1 del proyecto: <ul style="list-style-type: none"> <li>A. Identificación y delimitación del problema</li> <li>B. Propuesta de solución</li> </ul> </li> </ul>

### Unidad 4: Modelo de Diseño

<b>Competencia específica a desarrollar</b>	<b>Actividades de Aprendizaje</b>
<p>Aplicar modelos, técnicas y herramientas para la etapa de diseño del software.</p>	<ul style="list-style-type: none"> <li>• A partir del modelo de análisis del proyecto propuesto se realizarán las siguientes actividades: <ul style="list-style-type: none"> <li>• Refinamiento a clases.</li> <li>• Refinamiento a subsistemas</li> <li>• Refinamiento a diagramas de colaboración.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Refinamiento a diagramas de componentes.</li> <li>• Refinamiento a diagramas de actividades.</li> <li>• Refinamiento a diagrama de secuencia.</li> <li>• Realizar una tabla comparativa que muestre las inconsistencias detectadas.</li> <li>• Reporte de la estructura del sistema después de haber realizado el modelo de diseño en el caso de estudio.</li> <li>• Aplicar al menos una herramienta CASE para el diseño.</li> </ul>
--	--

## Unidad 5: Modelo de Implementación

Competencia específica a desarrollar	Actividades de Aprendizaje
Identificar procesos de la fase de implementación.	<ul style="list-style-type: none"> <li>• Aplicar al menos una herramienta CASE para generar código en algún lenguaje de programación a partir del diseño previo.</li> <li>• Investigar sobre las técnicas de pruebas y su clasificación.</li> <li>• Discutir sobre los métodos de implementación de las empresas de desarrollo de software de su entorno.</li> </ul>

## 11.- FUENTES DE INFORMACIÓN

1. Booch G. El lenguaje Unificado de Modelado, UML 2.0, Guia de Usuario. 1ª. Edición. Ed. Pearson ADDISON-WESLEY. España. 2006.
2. Cota, A. Ingeniería de Software: Soluciones Avanzadas. 2ª. Edición. Ed. Oxford. México. 2000.
3. Fowler M. UML Gota a Gota. 1ª. Edición. Ed. Pearson. México. 2000.
4. Jacobson ,I.,Booch,G.,Rumbaugh,J. El proceso unificado de desarrollo de software. Addison Wesley. España. 2003.
5. Kendall E. K., Análisis y Diseño de sistemas. 1ª. Edición. Prentice Hall. México. 2005.
6. Pressman, R.S. Ingeniería del Software un enfoque práctico. México. Mc Graw-Hill. Madrid, España. 2008.
7. Senn J.A. Análisis y Diseño de sistemas. 2ª Edición, Mc Graw Hill, México, 1996.

## **12.- PRÁCTICAS PROPUESTAS**

1. Realizar investigación de campo en diferentes empresas en donde identificarán los tipos de software que se utilizan.
2. Realizar investigación de campo en diferentes organizaciones en donde identificarán qué modelo de desarrollo utilizan.
3. Desarrollar un proyecto semestral que cubra las siguientes fases: Ingeniería de requisitos, Modelo de análisis, Modelo de diseño, Modelo de implementación.
4. Instalación y configuración de al menos 1 herramienta CASE
5. Aplicar la Herramienta CASE en cada una de las etapas del proyecto.