



**PLAN DE ESTUDIOS (PE):** Licenciatura en Ingeniería en Ciencias de la Computación

y Licenciatura en Ciencias de la Computación

**ÁREA:** Optativa Disciplinar

**ASIGNATURA:** Ingeniería de Software Avanzada

**CÓDIGO:** CCOS-601

**CRÉDITOS:** 6

**FECHA:** 30 de agosto de 2018





**1. DATOS GENERALES**

<b>Nivel Educativo:</b>	Licenciatura
<b>Nombre del Plan de Estudios:</b>	Licenciatura en Ingeniería en Ciencias de la Computación y Licenciatura en Ciencias de la Computación
<b>Modalidad Académica:</b>	Presencial
<b>Nombre de la Asignatura:</b>	Ingeniería de Software Avanzada
<b>Ubicación:</b>	Nivel Formativo
<b>Correlación:</b>	
<b>Asignaturas Precedentes:</b>	Ingeniería de Software
<b>Asignaturas Consecuentes:</b>	NA

**2. CARGA HORARIA DEL ESTUDIANTE**

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
<b>Horas teoría y práctica (16 horas = 1 crédito)</b>	<b>3</b>	<b>2</b>	<b>90</b>	<b>6</b>





**3. REVISIONES Y ACTUALIZACIONES**

Autores:	Anzures García Mario Archundia Sierra Etelvina Boone Rojas María del Rocío Carrillo Ruíz Maya Contreras González Meliza González Calleros Juan Manuel Guerrero García Josefina Molina García María del Consuelo Sánchez López Abraham Ambrosio Vázquez Alma Delia Somodevilla García María Josefa Pérez de Celis Herrero María de la Concepción
Fecha de diseño:	26 de Mayo de 2013
Fecha de la última actualización:	21 de abril de 2017
Fecha de aprobación por parte de la academia de área, departamento u otro.	30 de agosto de 2018
Revisores:	Sánchez López Abraham Sánchez Gálvez María Luz Adolfina Anzures García Mario Guerrero García Josefina González Calleros Juan Manuel Torrijos Muñoz María Teresa Molina García María del Consuelo Martínez Mirón Erika Annabel Pérez Marcial Judith Mejía Matías Hilda Torres Acuitlapa Omar
Sinopsis de la revisión y/o actualización:	Con la actualización del Mapa Curricular a semestres, el enfoque de competencias, y a bloques, y los nuevos cursos que le antecede, se ha replanteado el objetivo inicial del curso. Por tanto, se renovó el contenido y el orden de cada unidad del programa para ser impartido por semestres; así como la bibliografía en inglés.

**4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:**

Disciplina profesional:	Ciencias de la Computación, Tecnologías de la información, Informática
Nivel académico:	Maestría
Experiencia docente:	Mínima 2 años
Experiencia profesional:	Mínima 2 años



## 5. PROPÓSITO:

Al finalizar el curso, el alumno será capaz de comprender las metodologías ágiles de la ingeniería de software para realizar proyectos y productos de calidad.

- Conocer las metodologías de desarrollo derivadas del paradigma orientado a objetos y conocer las diferencias entre el desarrollo tradicional y el desarrollo ágil.
- Conocer los principios de mejoras y las áreas específicas donde se realiza el proceso de planeación y de estimación de Personal Software Process (PSP).
- Conocer la dinámica para definir objetivos de proyecto, los roles que componen un equipo Team Software Process (TSP), las partes esenciales de la calidad de software basado en TSP y la mecánica básica de mejoras del proceso.

## 6. COMPETENCIAS PROFESIONALES:

Competencias específicas de LCC:

- Formaliza y representa el conocimiento, de manera computable a través del análisis, diseño y modelado de algoritmos para la resolución de problemas en los ámbitos social, científico y profesional, de manera particular en entornos inteligentes.
- Desarrolla proyectos de investigación para la solución de problemas computacionales con el objetivo de contribuir al bienestar de la sociedad.
- Modela y diseña soluciones computacionales con base en los fundamentos matemáticos, principios algorítmicos y teorías de la Ciencia de la Computación para resolver diversas problemáticas sociales y laborales.
- Resuelve problemas complejos de computación mediante algoritmos y programas con la finalidad de hacer más eficiente cualquier sistema computacional.

Competencias específicas de ICC:

- Elaborar soluciones de sistemas de cómputo soportadas en modelos de proceso, metodologías y herramientas para resolver problemas.
- Diseñar soluciones creativas e innovadoras por medio del análisis, síntesis e implementación en sistemas de cómputo que cumplan con los estándares de calidad.
- Interactuar con el usuario entendiendo y atendiendo sus necesidades con el fin de darle soluciones en su competencia.
- Aplicar los avances tecnológicos más recientes en las áreas de desarrollo de aplicaciones de software, tratamiento de datos, redes de computadoras, sistemas empotrados, control digital y robótico con el fin de dar soluciones innovadoras a problemas en el desarrollo científico-tecnológico del país.

Justificación LCC:

El desarrollo de software es un proceso que requiere del seguimiento de metodologías que les permita producir software de calidad, por lo que el licenciado en ciencias computacionales





requiere entender que en dicho proceso existen varias etapas que involucran el diseño, análisis, y modelado de una abstracción de la realidad apoyados por el pensamiento crítico. Además, precisa tener un conocimiento actualizado de las innovaciones tecnológicas que le permita interactuar en un ambiente multidisciplinario y ofrecer propuestas que atiendan a problemáticas de otras áreas de conocimiento.

**Justificación ICC:**

El desarrollo de soluciones informáticas es un proceso que requiere los conocimientos sobre las metodologías actuales para el análisis, diseño e implementación de sistemas informáticos, por lo que el ingeniero informático requiere adquirir y comprender la teoría para ponerla en práctica en la gestión de sistemas informáticos, analizando la infraestructura y administrando el cambio para su posterior aplicación.

**7. CONTENIDOS TEMÁTICOS**

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Metodología RUP (Rational Unified Process)	1.1. Fase de Inseminación 1.2 Fase de Elaboración 1.3 Fase de Construcción 1.4 Fase de Transición	Sánchez, S., Sicilia, M.A., Rodríguez, D. (2012). Ingeniería del Software. Un enfoque desde la guía SWEBOK. México: Alfaomega.  Sommerville, I. (2015). Ingeniería de Software, 10th ed. Prentice Hall.  Pressman, R. S. (2014). Software engineering: a practitioner's approach, 8th ed. McGraw-Hill Education.  Kruchten, P. (2004). The rational unified process: an introduction. Addison-Wesley Professional.  Osis, J., & Donins, U. (2017). Topological UML Modeling: An





Unidad de Aprendizaje	Contenido Temático	Referencias
		Improved Approach for Domain Modeling and Software Development. Elsevier.
2. Metodologías ágiles	2.1 Introducción a los métodos ágiles. 2.1.1 Crystal 2.1.2 Dynamic System Development Method (DSDM) 2.1.3 Adaptive Software Development (ASD) 2.1.4 Lean Development (LD) 2.2 Programación extrema (XP) 2.3. SCRUM 2.3.1 Principios de SCRUM 2.3.2 Fases 2.3.3 Equipos de Trabajo 2.3.4 Gestión de Tiempo 2.3.5 Administración de Proyectos 2.3.6 Gestión de Prioridades	Pope-Ruark, R. (2017). Agile Faculty: Practical Strategies for Managing Research, Service, and Teaching. University of Chicago Press.  Duhigg, C. (2016). Smarter faster better: The secrets of being productive. Random House.  Sutherland, J., & Sutherland, J. J. (2014). Scrum: the art of doing twice the work in half the time. Currency.
3. Confiabilidad y seguridad	3.1 Propiedades de confiabilidad 3.2. Disponibilidad y fiabilidad 3.3 Protección 3.4 Seguridad	Sánchez, S., Sicilia, M.A., Rodríguez, D. (2012). Ingeniería del Software. Un enfoque desde la guía SWEBOK. México: Alfaomega.  Sommerville, I. (2015). Ingeniería de Software, 10th ed. Prentice Hall.  Pressman, R. S. (2014). Software engineering: a practitioner's approach, 8th ed. McGraw-Hill Education.





Unidad de Aprendizaje	Contenido Temático	Referencias
		Laudon, K.C. and Laudon, J.P. (2012). Sistemas de Información Gerencial. México: Pearson Educación.
4. Reutilización de software	4.1 Panorama de la reutilización 4.2 Frameworks de aplicación 4.3 Líneas de productos de software 4.4 Resultados de productos COTS	<p>Mor, Y., Mellar, H., Warburton, S., &amp; Winters, N. (Eds.). (2014). Practical design patterns for teaching and learning with technology. Springer.</p> <p>Sommerville, I. (2015). Ingeniería de Software, 10th ed. Prentice Hall.</p> <p>Pressman, R. S. (2014). Software engineering: a practitioner's approach, 8th ed. McGraw-Hill Education.</p> <p>Feathers, M. (2004). Working effectively with legacy code. Prentice Hall Professional.</p>
5. Calidad de Software	5.1 Modelos de Calidad 5.2 Modelos y estándares de proceso de Software 5.3 El modelo CMMI 5.4 El modelo TSP/PSP 5.5. El modelo ISO-15504. 2.4. El modelo mexicano MOPROSOFT. 5.6 Pruebas del Software. 5.7 Verificación y Validación	<p>Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer.</p> <p>Ghazali, Umer W. (2014) Software testing: Essential skills for first time testers,</p>



Unidad de Aprendizaje	Contenido Temático	Referencias
		<p>Amazon Digital Services LLC, USA.</p> <p>Watkins, John (2009). Agile testing: How to succeed in a extreme testing environment, Cambridge University Press, USA.</p>







**8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS**

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> <li>● Lectura y comprensión</li> <li>● Reflexión</li> <li>● Comparación</li> <li>● Resumen</li> </ul> <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> <li>● ABP</li> <li>● Aprendizaje activo</li> <li>● Aprendizaje cooperativo</li> <li>● Aprendizaje colaborativo</li> <li>● Basado en el descubrimiento</li> </ul> <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> <li>● Aula</li> <li>● Laboratorio</li> <li>● Simuladores</li> <li>● Lenguajes de especificación y modelado.</li> </ul> <p>Técnicas</p> <ul style="list-style-type: none"> <li>● grupales</li> <li>● de debate</li> <li>● del diálogo</li> <li>● de problemas</li> <li>● de estudio de casos</li> <li>● cuadros sinópticos</li> <li>● mapas conceptuales</li> <li>● para el análisis</li> <li>● comparación</li> <li>● síntesis</li> <li>● mapas mentales</li> <li>● lluvia de ideas</li> <li>● analogías</li> </ul> <p>Exposición.</p>	<p>Libro del área de Ingeniería de Software  Pizarrón  Materiales audiovisuales: extractos de películas  Servicios telemáticos: páginas Web, Weblog, tours virtuales, webquest, correo electrónico, chats, foros, unidades didácticas y cursos on-line  TV y vídeo interactivos</p>

**9. EJES TRANSVERSALES**

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Promover una actitud de trabajo en equipo, la utilización de foros en temas asociados de Ingeniería de software y sus aplicaciones.





Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Utilización de aplicaciones gráficas para los procesos de análisis, diseño de los proyectos. Utilización de sistemas de gestión de contenido para el alojamiento de la información y uso de e-mail, chat, foros y blogs para la comunicación entre el profesor y estudiantes. Búsqueda de información en fuentes confiables electrónicas de temas relacionados a la materia y a su proyecto.
Desarrollo de Habilidades del Pensamiento Complejo	Desarrollo de competencias del alumno para analizar las problemáticas sociales, ambientales y/o tecnológicas, proponiendo soluciones factibles, aplicando procesos cognitivos, como: comprensión, análisis y síntesis, clasificación, diseño, creación, evaluación y toma de decisiones. Desarrollo de pensamiento creativo para plantear la soluciones del problema y el pensamiento crítico para identificar la mejor propuesta.
Lengua Extranjera	Comprensión de los documentos, libros, artículos y espacios Web en inglés u otros idiomas respecto de la asignatura y especializados en el área.
Innovación y Talento Universitario	Aplicación de sus talentos para innovar en el diseño de proyectos con calidad que resuelvan problemas de la vida real utilizando Ingeniería de Software.
Educación para la Investigación	Se fomenta el hábito de investigar e implementar procesos de análisis y diseño de sistemas, así como los cambios continuos de la tecnología invita al alumno a estar en una constante investigación de los avances tecnológicos y desarrollo del software en la solución de problemas.

## 10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
Actividades Objetivo	30
Participación en clase	10
Tareas	10
Trabajos de investigación u/o intervención	10
Proyecto final	40
Total	100%





### **11. REQUISITOS DE ACREDITACIÓN**

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones para tener derecho a exentar por evaluación continua y/o presentar el examen final en ordinario o extraordinario
Asistir como mínimo al 70% de las sesiones para tener derecho al examen extraordinario
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

#### **Notas:**

- a) La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- b) La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

