

## PROGRAMA DE CURSO

Código	Nombre			
CC4401	<b>Ingeniería de Software</b>			
Nombre en Inglés				
Software Engineering				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3002 Metodologías de Diseño y Programación, CC3201 Base de Datos			Obligatorio para Licenciatura en Computación.	
Resultados de Aprendizaje				
Al término del curso, el alumno demuestra que identifica enfoques de desarrollo cuando el software es de gran tamaño o complejidad, reconociendo las etapas involucradas en el desarrollo de software. Además, maneja técnicas de especificación de requisitos funcionales, operacionales y de datos, conoce estrategias de diseño basadas en patrones arquitectónicos y técnicas de verificación y validación de software.				

Metodología Docente	Evaluación General
<ul style="list-style-type: none"> <li>• Clases de cátedra que incluyen: <ul style="list-style-type: none"> <li>○ Ejemplos motivadores</li> <li>○ Presentación de materia para su resolución</li> </ul> </li> <li>• Clases auxiliares: <ul style="list-style-type: none"> <li>○ Resolución de problemas</li> <li>○ Apoyo a las tareas</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Dos controles de materia</li> <li>• Dos controles de lectura</li> <li>• Un Examen</li> <li>• Cuatro o cinco tareas: <ul style="list-style-type: none"> <li>▪ Conceptos generales IS</li> <li>▪ Especificación de requisitos</li> <li>▪ Diseño de interfaces</li> <li>▪ Programación y pruebas</li> </ul> </li> </ul>

### Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas	
1	Conceptos de Ingeniería de Software	2,5	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> <li>Conceptos generales de ingeniería de software.</li> </ul>		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> <li>- Maneje conceptos de software, ingeniería, ciencias de la computación e ingeniería de software.</li> <li>- Identifique propiedades de un software de calidad.</li> <li>- Reconozca principios generales para el desarrollo de software de calidad.</li> </ul>	[1]

Número	Nombre de la Unidad	Duración en Semanas	
2	Metodologías de Desarrollo Software	2,5	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> <li>Modelos de ciclo de vida del desarrollo de software.</li> <li>Modelo de cascada, iterativo-incremental, espiral, XP.</li> </ul>		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> <li>- Identifique las etapas en el desarrollo de software.</li> <li>- Organice estas etapas en los distintos modelos de ciclo de vida.</li> </ul>	[2][3]

Número	Nombre de la Unidad	Duración en Semanas	
3	Requisitos del Software	4	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> <li>Concepto de requisitos del software.</li> <li>Técnicas para especificación de requisitos.</li> </ul>		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> <li>- Identifique conceptos de requisitos del software.</li> <li>- Utilice técnicas para especificar: funcionalidad, operaciones y datos.</li> </ul>	[1][2][3]

Número	Nombre de la Unidad	Duración en Semanas	
4	Diseño del Software	3	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> <li>Diseño arquitectónico del software.</li> <li>Patrones de arquitectura</li> <li>Diseño de interfaces con el usuario.</li> <li>Diseño orientado a objetos.</li> </ul>		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> <li>Abarque el diseño del software paso a paso.</li> <li>Aplique distintas estrategias de diseño para distintas porciones del software.</li> </ul>	[1][2]

Número	Nombre de la Unidad	Duración en Semanas	
5	Verificación y Validación del Software	3	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> <li>Conceptos de verificación y validación de software.</li> <li>Pruebas funcionales, estructurales y de integración.</li> <li>Verificación estática del software.</li> </ul>		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> <li>Maneje la utilidad de la verificación del software.</li> <li>Maneje Modelo V de verificación del software.</li> <li>Maneje pruebas de caja negra, pruebas de caja blanca, pruebas de integración.</li> <li>Utilice verificación estática.</li> </ul>	[1][2][3]

Bibliografía	
[1]	Fundamentals of Software Engineering (2nd Edition), Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Prentice Hall; September 29, 2002.
[2]	Software Engineering: (Update) (8th Edition), Ian Sommerville; Addison Wesley, June 4, 2006.
[3]	Software Engineering: A Practitioner's Approach /(7 edition), Roger Pressman; McGraw-Hill Science/Engineering/Math, January 20, 2009.

Vigencia desde:	Marzo de 2010
Elaborado por:	María Cecilia Bastarrica
Revisado por:	ADD (noviembre 2009)