

PROGRAMA DE CURSO

Código	Nombre			
CC55C	Ingeniería de Software			
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	
Autor sólo Postgrado			Electivo para Magister y Doctorado en Ciencias mención Computación	
Resultados de Aprendizaje				
<p>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.</p>				

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

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"> Conceptos generales de ingeniería de software. 		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> - Maneje conceptos de software, ingeniería, ciencias de la computación e ingeniería de software. - Identifique propiedades de un software de calidad. - Reconozca principios generales para el desarrollo de software de calidad. 	[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"> Modelos de ciclo de vida del desarrollo de software. Modelo de cascada, iterativo-incremental, espiral, XP. 		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> - Identifique las etapas en el desarrollo de software. - Organice estas etapas en los distintos modelos de ciclo de vida. 	[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"> Concepto de requisitos del software. Técnicas para especificación de requisitos. 		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> - Identifique conceptos de requisitos del software. - Utilice técnicas para especificar: funcionalidad, operaciones y datos. 	[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"> Diseño arquitectónico del software. Patrones de arquitectura Diseño de interfaces con el usuario. Diseño orientado a objetos. 		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> - Abarque el diseño del software paso a paso. - Aplique distintas estrategias de diseño para distintas porciones del software. 	[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"> Conceptos de verificación y validación de software. Pruebas funcionales, estructurales y de integración. Verificación estática del software. 		Al término de la unidad se espera que el alumno: <ul style="list-style-type: none"> - Maneje la utilidad de la verificación del software. - Maneje Modelo V de verificación del software. - Maneje pruebas de caja negra, pruebas de caja blanca, pruebas de integración. - Utilice verificación estática. 	[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:	Primavera 2013
Elaborado por:	María Cecilia Bastarrica
Revisado por:	