

PROGRAMA DE CURSO

Código	Nombre			
CC4401	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	
CC3002, CC3201			Obligatorio	
Resultados de Aprendizaje				
<ul style="list-style-type: none"> • Conocer enfoques de mantenimiento y evolución de sistemas de software de gran tamaño y/o complejidad. • Conocer las etapas involucradas en el desarrollo y la manutención de software. • Manejar técnicas de especificaciones de requisitos funcionales, operacionales y de datos. • Manejar herramientas importantes para la evolución del software (Git, GitHub, JRipples, Ant/Maven/Gradle/Make, Eclipse/Netbeans/IntelliJ). • Conocer técnicas de mejoramiento de la calidad de software (code smells, refactorings). • Conocer técnicas de verificación y validación de software (testing, revisión de código). • Conocer estrategias de diseño basadas en patrones de diseños y arquitectónicos. 				

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 ○ Seguimiento de proyecto de evolución de software • Clases auxiliares: <ul style="list-style-type: none"> ○ Resolución de problemas ○ Apoyo al proyecto 	<ul style="list-style-type: none"> • Un control de materia • Un Examen • Un proyecto de evolución software: <ul style="list-style-type: none"> ○ Proyecto existente, open source ○ Tamaño: 200 + clases ○ Basado en una propuesta de proyecto ○ Trabajo grupal coordinado con GitHub ○ Evaluación individual

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas	
1	Introducción a la Ingeniería de Software	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Conceptos generales de ingeniería de software. 		<ul style="list-style-type: none"> • Conceptos de software, ingeniería, ciencias de la computación e ingeniería de software. • Propiedades de un software de calidad. • Principios generales para el desarrollo de software de calidad. 	(1)

Número	Nombre de la Unidad	Duración en Semanas	
2	Especificación de Software	1	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Especificación de Software. • Concepto de requisitos del software. • Técnicas para especificación de requisitos. 		<ul style="list-style-type: none"> • Importancia de la especificación de software. • Requisitos de Usuarios. • Requisitos de Software. • El documento de requisitos. • Técnicas de Elicitation de requisitos. 	(1)

Número	Nombre de la Unidad	Duración en Semanas	
3	Mantenimiento y Evolución de Software	5	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Importancia de la evolución de software. • Fases del proceso de evolución de software. 		Introducción del cambio Locación de concepto Análisis de impacto Actualización y propagación del cambio Refactoring, Pre y Post-Factoring Conclusión	(2) (3) (4)

Número	Nombre de la Unidad	Duración en Semanas	
4	Verificación y Validación del Software	2	
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. • Revisión de Código. 		<ul style="list-style-type: none"> • Utilidad de la verificación del software. • Pruebas de caja negra, pruebas de caja blanca, pruebas de integración. • Verificación estática. • Revisión de Código. 	(1) (2) (5)

Número	Nombre de la Unidad	Duración en Semanas	
5	Diseño de Software	2	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Diseño orientado a objetos. • Diseño arquitectónico del software. • Patrones de diseño y de arquitectura 		<ul style="list-style-type: none"> • Poder abordar el diseño del software paso a paso. • Aplicar distintas estrategias de diseño para distintas porciones del software. 	(1) (4) (5)

Número	Nombre de la Unidad	Duración en Semanas	
6	Metodologías de Desarrollo Software	1	
Contenidos		Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Modelo de cascada, iterativo-incremental, espiral XP. • Modelos de ciclo de vida del desarrollo de software. 		<ul style="list-style-type: none"> • Etapas en el desarrollo de software. • Organizaciones de estas etapas en los distintos modelos de ciclo de vida. 	(1) (2)

Bibliografía

1. Fundamentals of Software Engineering (2nd Edition). Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Prentice Hall; September 29, 2002. ISBN: 0133056996
2. Software Engineering: The Current Practice (1st edition). Vaclav Rajlich; Chapman and Hall/CRC Innovations in Software Engineering and Software Development Series); November 17, 2011. ISBN: 1439841225
3. Refactoring: Improving the Design of Existing Code (1st Edition). Martin Fowler, Kent Beck; Addison-Wesley Professional; July 8, 1999. ISBN: 0201485672
4. Design Patterns: Elements of Reusable Object-Oriented Software (1st Edition); Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides; Addison-Wesley Professional; November 10, 1994. ISBN: 0201633612
5. Making Software: What Really Works, and Why We Believe It (1st Edition); Andy Oram, Greg Wilson; O'Reilly Media; October 30, 2010. ISBN: 0596808321

Vigencia desde:

Semestre Primavera 2016

Elaborado por:

Romain Robbes