
	UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. <i>Creada por Ley N°: 1.009/96 del 03/12/96</i> Facultad de Ingeniería	
PROGRAMA DE ESTUDIOS		

Materia:	Ingeniería en Software I		Semestre:	Séptimo
Ciclo:	Ingeniería Informática			
Código de la materia:	107			
Horas Semanales:	Teóricas:	4		
	Prácticas:	2		
	Laboratorio:	-		
Horas Semestrales:	Teóricas:	68		
	Prácticas:	34		
	Laboratorio:	-		
Pre-Requisitos:	Sistemas de Gestión y Organización y RRHH			

I. OBJETIVOS GENERALES:

El objetivo de la materia es que el alumno comprenda los principios de la Ingeniería del software, conozca sus metodologías, técnicas y herramientas y esté en condiciones de aplicar las mismas a situaciones prácticas que se plantearán a lo largo del desarrollo de las clases.

II. OBJETIVOS ESPECIFICOS

Al término de este curso los alumnos deberán haber desarrollado las siguientes capacidades:

1. Distinguir las características de los sistemas software.
2. Comprender los procesos del desarrollo del software
3. Gestionar proyectos informáticos
4. Comprender funciones y organizaciones de las empresas desarrolladora de software,
5. Comprender en qué consisten los proyectos especiales de desarrollo de software

III. CONTENIDOS PROGRAMATICOS:

Unidad I

Características de los sistemas software

1. Evolución de sistemas de software: software pequeño, grande, más grande, gigante, enorme; la necesidad de ingeniería; historia de la ingeniería; qué es ingeniería; el rol del ingeniero
2. El ciclo de vida del software
3. Tipos de organizaciones que necesitan software y cómo el tipo de organización influye en el proceso de ingeniería; tipos de software (sistemas informáticos, sistemas de tiempo real, sistemas distribuidos, sistemas integrados)

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 1 de 4
--	---	------------------------	---------------

	<p>UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. Creada por Ley N°: 1.009/96 del 03/12/96 Facultad de Ingeniería</p>	
<p>PROGRAMA DE ESTUDIOS</p>		

4. Software: sus cualidades, diferencias de requerimientos de calidad en diferentes tipos de software; medición de calidad; mitos de la medición
5. Modelos de procesos de desarrollo: diferencias según el tipo de software, los requerimientos de calidad, el tipo de organización que está desarrollando el software; modelo clásico (Análisis -> Diseño -> Implementación -> Testeo / Depuración -> Mantenimiento), modelo espiral, otros modelos modernos; la importancia del prototipo

Unidad II

Procesos del desarrollo del software

1. El proceso de Análisis
2. El proceso de Diseño
3. El proceso de Implementación
4. El proceso de Testeo/Depuración
5. El proceso de mantenimiento
6. Análisis de sistemas: determinación de necesidades; balanceo de necesidades con costo; técnicas de especificación de requerimientos; metodologías formales e informales; el analista, su rol en una empresa de software, en una empresa que necesita software y en una empresa con un producto que utiliza software; encuestas, prototipos y otras formas de determinar requerimientos; riesgos que enfrenta el analista; empresas que sobreviven sin analistas

Unidad III

Gestión de proyectos software

1. Balancear iniciativa individual con orden centralizado, compartiendo información en un proyecto gigante, documentos su uso y abuso, organización vs. sobreorganización
2. Problemas que ocurren en el proceso de desarrollo: el software con demasiados defectos, el proyecto que nunca acaba
3. Herramientas del ingeniero: la base de datos de defectos (y a veces también de componentes que todavía deben desarrollarse), el sistema de control de fuente, herramientas de comunicación (email, newsgroups, web, chat, documentos), herramientas de desarrollo (depuradores, compiladores, generadores de código, 4GL, etc..), herramientas de diseño (UML, etc.), herramientas de manejo de proyectos
4. Selección de herramientas, lenguajes, sistemas operativos, etc..
5. Manejo de proyectos: planificación, control, organización (centralizada, descentralizada, mixta), manejo de riesgos
6. Principios de trabajo en grupo: la consulta, la cooperación, la autoridad, respeto por los demás, asegurar que se tomen decisiones; problemas que deben evitarse (celos, prejuicios, persecución, acoso sexual, etc.); principios en las que debe basarse un buen ambiente de trabajo (justicia, igualdad de derechos, unidad del grupo, eliminación de murmuración, privacidad, etc.); ética del ingeniero (defender los intereses de su cliente, honestidad, privacidad, etc..)

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 2 de 4
--	---	------------------------	---------------

	<p>UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. Creada por Ley N°: 1.009/96 del 03/12/96 Facultad de Ingeniería</p>	
	<p>PROGRAMA DE ESTUDIOS</p>	

Unidad IV

Empresas y desarrollo de software

1. La Empresa desarrolladora de software: cómo suelen organizarse (desarrolladores, testers, ingenieros de configuración, ingenieros de instalación, documentación, arquitecto, soporte técnico, integradores del sistema, jefes y otros roles); el mercado competitivo y su influencia en el proceso de desarrollo; divisiones organizacionales del proyecto; el ambiente de trabajo y su influencia en productividad; problemas que ocurren; el cronograma de producción, sus complejidades; incentivos al programador; el proceso que permite que se puedan utilizar técnicas efectivas de desarrollo; modelos típicos de desarrollo y cómo difieren de los modelos clásicos; consideraciones técnicas del ambiente del shrink-wrapped software (no depender de otras empresas, código simple y mantenible, estandarización)
2. La Empresa que necesita de software: cómo suelen organizarse (analistas, diseñadores, arquitectos, programadores, testers, administración y otros roles); organización con contratistas; el costo y riesgo para la empresa al desarrollarse software y cómo influye al proceso; el ambiente de trabajo y su influencia a la productividad; conflictos entre teorías antiguas de administración de empresas y el proceso de desarrollo; los documentos de requerimientos, de diseño, teoría del manual antes que el producto, y otros documentos; documentos con contratistas (el RFP – pedido de propuestas, propuestas y contratos); consideraciones del ambiente informático de la empresa, resistencia al cambio, el problema de las “modas” en el software; uso de 4GLs y otras herramientas de desarrollo rápido

Unidad V

Proyectos especiales de desarrollo de software


3. Proyectos gubernamentales y gigantes: necesitan de mayor organización; coordinación de cientos o miles de programadores; contratistas y subcontratistas; peligros al proyecto
4. Software crítico: software de las que dependen vidas humanas o en las que una falla puede ser desastroso; técnicas organizacionales que se utilizan para asegurar calidad; el rol de documentación y rigor; el rol del hardware y del software en la seguridad del sistema
5. Consideraciones especiales para sistemas distribuidos
6. Consideraciones especiales para sistemas integrados y sistemas de tiempo real
7. Seguridad en el software: hackers, crackers, cómo trabajan y cómo nos protegemos de ellos; el proceso administrativo de protección del sistema; auditorías de seguridad
8. Trabajo práctico en grupo

IV. METODOLOGÍA

Las actividades de los alumnos comprenderán:

- Clases teóricas
- Estudio de casos
- Practicas supervisadas en laboratorio
- Elaboración de Trabajos prácticos
- Elaboración y presentación de trabajos
- Investigaciones

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 3 de 4
--	---	------------------------	---------------

	<p>UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. Creada por Ley N°: 1.009/96 del 03/12/96 Facultad de Ingeniería</p>	
	<p>PROGRAMA DE ESTUDIOS</p>	

V. CRITERIOS DE EVALUACIÓN

Conforme al Reglamento Académico y Reglamento de Cátedra vigentes.

VI: BIBLIOGRAFÍA

McConnell, S. (1997). Desarrollo y gestión de proyectos informáticos. McGraw-Hill Interamericana de España.

Edward, K. (1995). Software Testing in the real world. Pearson Education India.

Booch, G., Rumbaugh, J., Jacobson, I., & Molina, J. J. G. (1999). El lenguaje unificado de modelado (Vol. 1). Addison-Wesley.

Kroll, P., & Kruchten, P. (2003). The rational unified process made easy: a practitioner's guide to the RUP. Addison-Wesley Professional.

Jacobson, I., Booch, G., & Rumbaugh, J. (2000). El proceso unificado de desarrollo de software (Vol. 7). Reading: Addison Wesley.

Craig, L. (2003). UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado.

Page-Jones, M., & Constantine, L. L. (2000). Fundamentals of object-oriented design in UML. Addison-Wesley Professional.

Pressman, R. S. . Mc Grah-Hill, 7a. edición, 2010. Ingeniería del software, un enfoque práctico.

Sommerville, I., & Galipienso, M. I. A. (2005). Ingeniería del software. Pearson Educación.

Apuntes de cátedra

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 4 de 4
--	---	------------------------	---------------