

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
COORDINACIÓN DE FORMACIÓN BÁSICA
COORDINACIÓN DE FORMACIÓN PROFESIONAL Y VINCULACIÓN UNIVERSITARIA
DEPARTAMENTO DE ACTUALIZACIÓN CURRICULAR Y FORMACIÓN DOCENTE

DESCRIPCIÓN GENÉRICA DE UNIDADES DE APRENDIZAJE

Descripción Genérica

Nombre: Ingeniería de Software

Etapas: *Disciplinaria*

Área de conocimiento: Ingeniería del Software

Plan 2008-1

Competencia:

Analizar, diseñar, e implementar productos de software con calidad como solución a un problema real, eligiendo las normas de calidad, métodos y tecnologías que permitan especificar, diseñar y generar el código, comprometiéndose con su equipo trabajo.

Evidencia de desempeño:

Se desarrollará y presentará un producto de software terminado resultado de un proyecto final. Además de exposición oral y escrita de artefactos generados durante la fase de desarrollo. Se aplicarán exámenes de la teoría presentada.

Distribución	HC	HL	HT	HPC	HCL	HE	CR	Requisito
	2		4			2	8	Metodología de la Programación, Programación O-O, Introducción a la Programación

Contenidos Temáticos

1. Introducción a la Ingeniería de Software (IS)

- 1.1. ¿Qué es la Ingeniería de software?
- 1.2. Historia de la IS
- 1.3. Naturaleza y cualidades del software
- 1.4. Administración de proyectos
- 1.5. Plan de administración de proyecto
- 1.6. Herramientas y documentación en IS

2. Procesos formales de desarrollo de Software

- 2.1. Proceso de Software
- 2.2. Modelo de ciclo de vida del software
- 2.3. Conceptos de Modelado Orientado a Objetos y Patrones
- 2.4. Diagramas de UML

- 2.5. Modelado del Proceso de Software con UML
- 2.6. Modos y mecanismos de comunicación del equipo de desarrollo
- 2.7. Modelos de referencia

3. Análisis de requerimientos y especificación

- 3.1. Requerimientos vs especificación
- 3.2. Conceptos de la obtención de requerimientos
- 3.3. Actividades para la obtención de requerimientos
- 3.4. Administración de requerimientos

4. Análisis Orientado a Objetos (AOO)

- 4.1. Refinamiento del modelo de casos de uso
- 4.2. Conceptos de análisis
- 4.3. Actividades de AOO
- 4.4. Administración del análisis

5. Diseño Orientado a Objetos (DOO)

- 5.1. Conceptos del DOO
- 5.2. Actividades del DOO para obtener el modelo completo
- 5.3. Administración del diseño del sistema
- 5.4. Modelo de diseño de objetos
- 5.5. Diseño de patrones
- 5.6. Programación Orientada a Objeto (POO)

6. Pruebas Orientada a Objetos y Control de Calidad

- 6.1. Conceptos de Control de Calidad
- 6.2. Pruebas de los modelos AOO y DOO
- 6.3. Estrategias de de Pruebas Orientadas a Objetos (POO)
- 6.4. Diseño de casos de POO
- 6.5. Métodos de pruebas aplicables al nivel de clases
- 6.6. Diseño de caso de pruebas inter-clases.

7. Métricas de la calidad y administración de configuraciones

- 7.1. Modelos de referencia para la calidad en la IS
- 7.2. Administración de Configuración de Software
- 7.3. Control de Versiones
- 7.4. Control de cambios
- 7.5. Auditoria de la configuración
- 7.6. Informe de estado

Referencias bibliográficas actualizadas

Básica	Complementaria
<ul style="list-style-type: none"> * Pressman, R., Ingeniería del software: un enfoque práctico. S.5ª ed. McGraw Hill. 2002, ISBN 844813214-9. * Bernd Bruegge and Allen H. Dutoit, Object-Oriented Software Engineering, Prentice Hall Inc., 2000 * Rumbaugh, J., et al., Object-Oriented Modeling and Design, Prentice Hall Inc., 1991 * Notas del curso, disponibles en el sitio del curso. * Artículos diversos disponibles en Internet. 	<ul style="list-style-type: none"> * G. Booch, J.Rumbaugh, I.Jacobson, The Unified Modeling Language User Guide, Addison Wesley, 1999. * P. Stevens, R. Pooley, Using UML Software Engineering with Objects and componets, Addison Wesley, 1999. * Sommerville, Software Engineering, Fifth Edition, Addison Wesley, 1995. * R. Fairley, Ingeniería del Software, Mc.Graw-Hill.