



PROGRAMA DE ESTUDIO POR COMPETENCIAS
MÉTRICAS DE SOFTWARE

I. IDENTIFICACIÓN DEL CURSO

ESPACIO EDUCATIVO: Facultad de Ingeniería						
LICENCIATURA: Ingeniería en Computación				ÁREA DE DOCENCIA: Programación e Ingeniería de Software		
AÑO DE APROBACIÓN POR EL CONSEJO UNIVERSITARIO:						
APROBACIÓN POR LOS HH. CONSEJOS ACADÉMICO Y DE GOBIERNO		FECHA:		PROGRAMA ELABORADO POR: Ing. Elfego Gutiérrez Ocampo		PROGRAMA REVISADO POR: Integrantes de la Academia de Programación e Ingeniería de Software
				FECHA DE ELABORACIÓN : Abril 2008		FECHA DE REVISIÓN : Mayo 2011
CLAVE	HORAS DE TEORÍA	HORAS DE PRÁCTICA	TOTAL DE HORAS	CRÉDITOS	TIPO DE CURSO	NÚCLEO DE FORMACIÓN
L41076	2	2	4	6	Curso	Integral
UNIDAD DE APRENDIZAJE ANTECEDENTE Ninguna				UNIDAD DE APRENDIZAJE CONSECUENTE Ninguna		
PROGRAMAS EDUCATIVOS O ESPACIOS ACADÉMICOS EN LOS QUE SE IMPARTE: Licenciatura en Ingeniería en Computación (Facultad. de Ingeniería, Centros Universitarios: Atlacomulco, Ecatepec, Texcoco, Valle de Chalco, Valle de México, Valle de Teotihuacán, Zumpango)						



II. PRESENTACIÓN

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a cada vez más, los procesos principales de las organizaciones dependen de los sistemas informáticos para su buen funcionamiento.

En la evolución experimentada por la calidad del software se ha pasado de un tratamiento centrado fundamentalmente en la inspección y detección de errores, a una aproximación más sistemática, dada la importancia que ha adquirido la calidad en la ingeniería del software. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora tanto de productos como de procesos de software. Todo ello ha influido de forma significativa en el papel que actualmente tiene la calidad en las organizaciones, que pasa de convertirse en una filosofía y una cultura que afecta a toda la organización.

III. LINEAMIENTOS DE LA UNIDAD DE APRENDIZAJE

DEL DOCENTE	DEL DISCENTE
<ul style="list-style-type: none">• Cumplir en tiempo y contenido el programa de la unidad de aprendizaje.• Estar presente en el horario establecido en el aula y sala de cómputo para prácticas.• Establecer tolerancia para el inicio de clases y prácticas.• Proponer y respetar las formas de evaluación.• Respetar el número de horas teóricas y prácticas de la unidad de aprendizaje.	<ul style="list-style-type: none">• Contar con el porcentaje acordado en asistencias para presentar el examen ordinario y extraordinario.• Hacer uso adecuado del laboratorio.• Respetar el reglamento de la sala de cómputo.• Tener la respectiva responsabilidad para desarrollar y entregar en tiempo los trabajos extraclase.• Tener sentido de integración en los trabajos por equipo en la sala de cómputo.

IV. PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Situar al futuro profesionista ante los conceptos básicos y más recientes de la calidad del software, familiarizándolo con la teoría de la medición del software y sus procesos interrelacionados (estimación, calidad, complejidad, etc.), a fin de que la comprensión de las técnicas de medición sea



para él un instrumento científico eficaz para el análisis crítico de la problemática inherente al desarrollo de software con calidad, a tiempo y dentro de presupuesto (crisis del software). Y al mismo tiempo contrastar de manera realista estos principios básicos con las realidades de la gestión de software en las empresas, examinándolos en el contexto de las prácticas empíricas del desarrollo de software que aún prevalecen en la actualidad.

V. COMPETENCIAS GENÉRICAS

El alumno aplicará el conocimiento adquirido en la teoría complementándolo con la investigación y aplicación de los diferentes modelos de calidad de software utilizando las técnicas vistas durante el curso, las cuales le darán la pauta acerca del tamaño, complejidad, calidad y estimación del software que construirá dentro de la universidad como estudiante y en su vida profesional.

VI. ÁMBITOS DE DESEMPEÑO

Empresas públicas y privadas de los sectores industrial, comercial y educativo.

VII. ESCENARIOS DE APRENDIZAJE

Aulas, salas de cómputo, conferencias, revistas.

VIII. ESTRUCTURA DE LA UNIDAD DE APRENDIZAJE

1. Ingeniería de software
2. Teoría de la medición del software
3. Métricas del proceso y del proyecto
4. Métricas del producto
5. Modelos de estimación del software
6. Modelos de calidad
7. Calidad del producto



- 8. Calidad del proceso
- 9. Programa de métricas de software

IX. DESARROLLO DE LA UNIDAD DE APRENDIZAJE

UNIDAD DE COMPETENCIA I	ELEMENTOS DE COMPETENCIA		
	Conocimientos	HABILIDADES	ACTITUDES/ VALORES
Ingeniería de software	La crisis del software Objetivos de la ingeniería de software Proceso, producto, proyecto y personas	Observación Conceptualización Análisis	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Revistas de software Internet	TIEMPO DESTINADO 4 horas
CRITERIOS DE DESEMPEÑO I	EVIDENCIAS		
		DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito	Asistencia Examen escrito Tareas	La crisis del software Objetivos de la ingeniería de software Proceso, producto,	



		proyecto y personas
--	--	---------------------

UNIDAD DE COMPETENCIA II	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Teoría de la medición del software	Historia de la medición en el software Tecnología de la medición Precisión y exactitud Escalas de medición Medidas métricas e indicadores Toma de decisiones basada en la medición	Observación Conceptualización Análisis	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Internet	TIEMPO DESTINADO 4 horas
CRITERIOS DE DESEMPEÑO II	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito	Asistencia Examen escrito Tareas	Historia de la medición en el software Tecnología de la medición Precisión y exactitud Escalas de medición Medidas métricas e indicadores	



		Toma de decisiones basada en la medición
--	--	--

UNIDAD DE COMPETENCIA III	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Métricas del proceso y del proyecto	Modelos de procesos Clasificación de métricas Entidades Atributos Ingeniería de requerimientos Control de riesgos	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 4 horas
CRITERIOS DE DESEMPEÑO III	EVIDENCIAS		
		DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito	Asistencia Examen escrito Tareas	Modelos de procesos Clasificación de métricas Entidades	



		Atributos Ingeniería de requerimientos Control de riesgos
--	--	---

UNIDAD DE COMPETENCIA IV	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Métricas del producto	Tamaño físico del software Tamaño funcional del software Métricas de mccabe Métricas de halstead Líneas de código Puntos de función Métricas orientadas a objetos	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Manual de prácticas de conteo Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 8 horas
CRITERIOS DE DESEMPEÑO IV	EVIDENCIAS		
	DESEMPEÑO	PRODUCTOS	



<p>Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software</p>	<p>Asistencia Examen escrito Tareas Proyecto de desarrollo</p>	<p>Tamaño físico del software Tamaño funcional del software Métricas de mccabe Métricas de halstead Líneas de código Puntos de función Métricas orientadas a objetos</p>
--	--	--

UNIDAD DE COMPETENCIA V	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Modelos de estimación del software	Modelos de estimación Modelo COCOMO	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
<p>ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo</p>		<p>RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de</p>	<p>TIEMPO DESTINADO 4 horas</p>



	aplicaciones	
CRITERIOS DE DESEMPEÑO V	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software	Asistencia Examen escrito Tareas Proyecto de desarrollo	Modelos de estimación Modelo COCOMO

UNIDAD DE COMPETENCIA VI	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Modelos de calidad	Modelo de MCCALL Modelo ISO 9126	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 4 horas



CRITERIOS DE DESEMPEÑO VI	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
	Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software	Asistencia Examen escrito Tareas Proyecto de desarrollo

UNIDAD DE COMPETENCIA VII	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Calidad del producto	Atributos de calidad Vector de calidad Índice de calidad Cálculo de la calidad	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 8 horas



CRITERIOS DE DESEMPEÑO VII	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software	Asistencia Examen escrito Tareas Proyecto de desarrollo	Atributos de calidad Vector de calidad Índice de calidad Cálculo de la calidad

UNIDAD DE COMPETENCIA VIII	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Calidad del proceso	Calidad del proceso Componentes del proceso Actividades por componente Escala de valores por actividad	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo		RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 8 horas
CRITERIOS DE DESEMPEÑO VIII	EVIDENCIAS		



	DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software	Asistencia Examen escrito Tareas Proyecto de desarrollo	Calidad del proceso Componentes del proceso Actividades del proceso Escala de valores por actividad

UNIDAD DE COMPETENCIA IX	ELEMENTOS DE COMPETENCIA		
	CONOCIMIENTOS	HABILIDADES	ACTITUDES/ VALORES
Programa de métricas de software	Definición de métricas Recolección de métricas Creación de base de datos de métricas Obtención de indicadores	Análisis y diseño de aplicaciones, usando un lenguaje de programación determinado.	Responsabilidad para cumplir con las tareas asignadas Actitud prepositiva, constructivista e innovadora Tolerancia y participación Trabajo en equipo
ESTRATEGIAS DIDÁCTICAS: Revisión y análisis de textos Explicación por parte del instructor de temas específicos de mayor complejidad Solución de ejercicios Desarrollo de programas Discusión en grupo	RECURSOS REQUERIDOS Pizarrón Libros de texto Apuntes pre-elaborados Computadora Internet Software de desarrollo de aplicaciones	TIEMPO DESTINADO 4 horas	



CRITERIOS DE DESEMPEÑO IX	EVIDENCIAS	
	DESEMPEÑO	PRODUCTOS
Asistencia a clases Realizar investigaciones Consultar referencias en internet Presentar examen escrito Realizar proyecto de desarrollo de software	Asistencia Examen escrito Tareas Proyecto de desarrollo	Definición de métricas Recolección de métricas Creación de base de datos de métricas Obtención de indicadores

X. EVALUACIÓN Y ACREDITACIÓN

Para que el alumno pueda acreditar esta unidad de aprendizaje, se considerarán tres exámenes parciales obteniendo una calificación parcial final que le dará derecho a exentar o a presentar el examen ordinario o extraordinario de acuerdo al siguiente criterio:
Si la calificación parcial final es mayor o igual a 8.0, el alumno exenta el examen ordinario y se le asienta esa calificación
Si la calificación parcial final se encuentra entre 6.0 y 8.0, el alumno presenta examen ordinario
Si la calificación parcial final es menor a 6.0, el alumno presenta examen ordinario.
La calificación final del curso se obtiene de la siguiente manera:

100 % tres exámenes parciales

Las faltas se descontarán de la calificación parcial con un valor 2 décimos por faltas

Las tareas se descontarán de la calificación parcial con un valor de 3 décimos por tarea no-entregada.

Los exámenes consistirán de:

1. Un examen escrito acerca de la teoría vista durante el curso con un peso del 20 %.
2. Un proyecto de estimación de métricas de un sistema de información determinado (producto) calculando lo siguiente: tamaño, esfuerzo, tiempo, número de personas y costo con un peso del 40 %.
3. Un proyecto de medición de la calidad del producto o del proceso de software con un peso del 40 %.

Para poder obtener una calificación aprobatoria en el examen ordinario, el alumno deberá haber cumplido al menos con el 80 % de asistencias como mínimo.



Los exámenes extraordinario y a título de suficiencia consistirán de un examen teórico con un valor de un 40 % y un examen práctico con un valor de un 60 % y que consistirá de un proyecto parecido a los presentados en algunos de los exámenes parciales, a saber la estimación del tamaño de un sistema software medido en puntos de función y en líneas de código, el esfuerzo requerido y el tiempo estimado. O el proyecto puede ser medir la calidad de un sistema software mediante una de las técnicas vistas en clase.

X. REFERENCIAS

BIBLIOGRAFIA BÁSICA

- ✓ Fenton, Norman E. Y Shari Lawrence Pleegeer. (1998) Software Metrics: A Rigorous and Practical Aproach. Toronto. INTERNATIONAL THOMPSON PUBLISHERS.
- ✓ Pandian, C. Ravindranath. (2004) Software Metrics: A Guide to Planning, Analisys, and Application. London. AUERBACH PUBLICATIONS.
- ✓ Kitchenham, Barbara. (1996) Software Metrics: Measurement for Software Process Improvement. Cambridge. THE NATIONAL COMPUTING CENTRE.
- ✓ Goodman, Paul. (2004) Software Metrics: Best Parctices for Succesful IT Managemenet. Conneticut. ROTHSTEIN ASSOCIATES PUBLISHER.

BIBLIOGRAFIA COMPLEMENTARIA

- ✓ Oman, Paul y Shari Lawrence Pleegeer. (1997) Applying Software Metrics. California. IEEE COMPUTER SOCIETY.
- ✓ Kan, Stephen H. (2003) Metrics and Models in Software Quality Engineering. 2da. Edición. Boston. ADDISON-WESLEY.
- ✓ Garmus, David y David Herron. (2001) Function Point Analysis: A Measurement Practices for Succesful Software Projects. Boston. ADDISON-WESLEY.
- ✓ Putnam, Lawrence H. y Ware Myers. (2003) Five Core Metrics: The Intelligence Behind Succesful Software Management. New York. DORSET HOUSE PUBLISHING.
- ✓ Jones, Capers. (1996) Applied Software Measurement: Assuring Productivity and Quality. 2da. Edición. New York. MCGRAW-HILL.
- ✓ Stutzke, Richard D. (2005) Estimating Software Intensive Systems: Projects, Products and Processes. Boston. ADDISON-WESLEY.
- ✓ Humphrey, Watts S. (1998) Managing the Software Process. Boston. ADDISON-WESLEY.
- ✓ Tian, Jeff. (2005) Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. Dallas. WILEY.
- ✓ Coombs, Paul. (2003) IT Project Estimation: A Practical Guide to the Costing of Software. Cambridge. CAMBRIDGE UNIVERSITY PRESS.
- ✓ Lorenz, Mark y Jeff Kidd. (1994) Object-Oriented Software Metrics. New Jersey. PRENTICE-HALL.
- ✓ Henderson-Sellers, Brian. (1996) Object-Oriented Metrics: Measures of Complexity. New Jersey. PRENTICE-HALL.



- ✓ Bohem, Barry W. (1981) Software Engineering Economics. New Jersey. PRENTICE-HALL.
- ✓ Bohem, Barry W. (2000) Software Cost Estimation with COCOMO II. New Jersey. PRENTICE-HALL.
- ✓ Dolado Cosín, José Javier y Luis Fernando Sanz. (2000) Medición para la Gestión en la Ingeniería de Software. Madrid. RA-MA.
- ✓ Piattini, Mario G. Y Félix O. García. (2003) Calidad en el Desarrollo y Mantenimiento del Software. México. AlfaOmega RA-MA.
- ✓ Humphrey, Watts S. (2001) Introducción al Proceso Personal de Software. Madrid. ADDISON-WESLEY.
- ✓ Ejiogu, Lem O. (2005) Software Metrics: The Discipline of Software Quality. San Diego. SOFTMETRIX, INC.
- ✓ Florac, William A. y Anita D. Carleton. (1999) Measuring the Software Process: Statistical Process Control for Software Process Improvement. Boston. ADDISON-WESLEY.
- ✓ Humphrey, Watts S. (2000) Introduction to the Team Software Process. Boston. ADDISON-WESLEY.
- ✓ Beth Chrissis, Mike Konrad y Sandy Shrum. (2003) CMMI: Guidelines for Process Integration and Product Improvement. Boston, ADDISON-WESLEY.
- ✓ Donaldson, Scott E. y Stanley G. Siegel. (2001) Successful Software Development. 2da. Edición. New York. Prentice may.
- ✓ Galin, Daniel. (2004) Software Quality Assurance: From theory to implementation. London. ADDISON-WESLEY.