

Nombre de la asignatura: **INGENIERÍA DE SOFTWARE**

Línea de Trabajo: **Ingeniería de Software/ Inteligencia Artificial**

Tiempo de dedicación del estudiante a las actividades de:

DOC	TIS	TPS	Horas Totales	Créditos
48	20	100	168	6

**DOC:** Docencia; **TIS:** Trabajo independiente significativo; **TPS:** Trabajo profesional supervisado

### 1. Historial de la Asignatura.

Lugar y Fecha de Elaboración o Revisión	Participantes	Observaciones (cambios y justificaciones)
30 junio 2010	MC. Esmeralda Covarrubias Flores MC. Bertha Karina Leyto Delgado	Versión preliminar
21 noviembre 2011	MC. Pedro Luis Sánchez Orellana	Adecuación/Actualización

### 2. Pre-requisitos

Ninguno

### 3. Objetivo de la Asignatura

Aplicar técnicas de ingeniería para el desarrollo de productos de software, con el propósito de hacer más eficientes los recursos de cómputo disponibles, así como determinar el esfuerzo dedicado en la producción del software.

### 4. Aportación del Perfil del Graduado

Mediante el aprendizaje de las estrategias para incrementar la calidad del software el alumno será capaz de plantear, analizar y resolver en forma original problemas existentes en el sector público y privado.

Además, el alumno empleará y aplicará las diferentes metodologías de ingeniería en el desarrollo de sistemas de software.

### 5. Contenido Temático

Unidad	Temas	Subtemas
1	Lenguaje orientado a objetos	1.1. Introducción 1.2. Aspectos del diseño 1.3. Análisis de requerimientos 1.4. El paradigma orientado a objetos 1.5. Modelos de ciclos de vida del software.
2	Administración de Proyectos de Software	2.1 Planeación y estimación 2.1.1 Métricas de producción 2.1.2 Modelos de estimación

		2.1.3 LOC, FP, COCOMO, etc. 2.2 Administración de proyectos 2.2.1 Planeación 2.2.2 Calendarización 2.2.3 Seguimiento
3	Ingeniería de requerimientos	3.1. Análisis de requerimientos 3.1.1.Las necesidades del cliente 3.1.2.Requerimientos iniciales 3.1.3.Comprensión del dominio de las necesidades 3.2. El modelo de negocios 3.3. Requerimientos orientados a objetos 3.4. Herramientas CASE para requerimientos 3.5. Métricas de los requerimientos 3.6. Desafíos de los requerimientos
4	Calidad y medición de los sistemas de software	4.1. Métricas del software 4.2. Estándares de calidad 4.3. Problemas en la calidad 4.3.1.Software Quality Assurance 4.3.2.Managerial Independence 4.4. FTR 4.4.1.Unidades de prueba 4.4.2.Pruebas de integración 4.4.3.Prueba de aceptación
5	Procesos de software	5.1. Planeación y el proceso de software 5.2. Estimación de duración y costo 5.2.1.Métricas para evaluar el tamaño de un producto 5.2.2.Técnicas para estimación de costo 5.2.3.COCOMO, COCOMO II 5.3. Componentes de un proyecto de software 5.4. Administración vía IEEE estándar 5.5. Planeación de pruebas 5.6. Herramientas CASE para planeación
6	Diseño de software	6.1 Diseño orientado a objetos 6.2 Diseño de flujos de trabajo 6.2.1 Análisis estático 6.2.2 Identificación de objetos-métodos 6.2.3 Análisis dinámico 6.2.4 Identificación de interacciones-colaboraciones 6.3 Concepto de diseño de sistemas 6.4 Estilos arquitecturales 6.5 Patrones de diseño
7	Validación de software	7.1 Elementos de validación 7.1.1 Utilidad 7.1.2 Robustes 7.1.3 Desempeño 7.1.4 Validez 7.2 Unidades de verificación 7.2.1 Métricas de verificación

		7.2.2 Inspección 7.2.3 Fortalezas y debilidades en la verificación 7.3 Verificación de la aceptación del usuario 7.4 Pruebas vs Pruebas de exactitud
--	--	---

## 6. Metodología de Desarrollo del Curso

El profesor expondrá los conceptos teóricos del curso y ejemplos prácticos del mismo. El alumno realizara las prácticas de lo aprendido.

## 7. Sugerencias de Evaluación

- Exámenes parciales escritos al final de cada unidad.
- En la unidad II el alumno desarrollará un ejercicio simulando la gestión de proyecto de ingeniería de software.
- De la unidad III en adelante el alumno realizará ejemplos de un sistema de software con la finalidad de probar las estrategias de validación y Se evaluará al alumno con un proyecto de desarrollo de un sistema, donde utilice los esquemas de procesos, diseño y validación del software.

## 8. Bibliografía y Software de Apoyo

- Bernd Bruegge, "Object oriented software engineering", Second Edition, Pearson Education.
- Stephan R. Schach, "Object oriented software engineering", Tata McGraw Hill.
- Roger Pressman, "Software Engineering", sixth edition, Tata McGraw Hill.
- Timothy C. Lethbridge, Robert Laganieri " Object-Oriented Software Engineering – A practical software development using UML and Java", Tata McGraw-Hill, New Delhi
- Fabio Q.B. da Silva, André L.M. Santos, Sérgio Soares, A. César C. França, Cleviton V.F. Monteiro, Felipe Farias Maciel, Six years of systematic literature reviews in software engineering: An updated tertiary study, Information and Software Technology, Volume 53, Issue 9, September 2011, Pages 899-913, ISSN 0950-5849.
- Emilia Garcia, Adriana Giret, Vicente Botti, Evaluating software engineering techniques for developing complex systems with multiagent approaches, Information and Software Technology, Volume 53, Issue 5, May 2011, Pages 494-506, ISSN 0950-5849.

## 9. Actividades Propuestas

Unidad	Actividades
1	Realizar una exposición acerca de las ventajas del modelado orientado a objetos y un ejemplo práctico del mismo.
2	El alumno modelará un problema (proporcionado por el docente) desde el punto de vista de gestión de proyectos de software.

3	Realizar equipos de 2 integrantes (1 desde el punto de vista del cliente y el segundo desde el punto de vista del desarrollador) para ejemplificar el proceso de requerimientos de software.
4,5	Utilizando las metodologías para medición de calidad del software, proponer una estrategia que ayude a mejorar la calidad en el software.
6,7	Como proyecto final, realizar la implementación del problema asignado haciendo uso de las estrategias de diseño y validación de software.

#### 10. Nombre y Firma del Catedrático Responsable



M.C. en CC Pedro Luis Sánchez Orellana