

CURSO:	FUNDAMENTOS DE INGENIERÍA DE SOFTWARE
---------------	--

1	SEMESTRE:	V	
2	CODIGO:	602503	
3	COMPONENTE:		
4	CICLO:		
5	AREA:	Profesional	
6	FECHA DE APROBACIÓN:		
7	NATURALEZA:	TEÓRICO – PRÁCTICO.	
8	CARÁCTER:	Obligatorio	
9	CREDITOS (RELACIÓN):	4(1-1)	
10	INTENSIDAD HORARIA:	192	Total Horas/Semestre
		64	Horas/Semestre de trabajo presencial
		128	Horas/Semestre de trabajo independiente

11. JUSTIFICACION:

La construcción de software hoy exige satisfacer requerimientos de calidad cada vez más complejos debido al creciente nivel de sofisticación de la tecnología, negocios y necesidades de la sociedad. Lidar con esta complejidad necesita de técnicas y métodos concretos que ayuden a producir software de calidad con tiempos y costos controlados. La tendencia es la obtención masiva de productos de software de la más amplia variedad y a costos comparativamente mínimos[1].

La sociedad en general espera software abundante, elaborado en forma rápida, muy eficiente y a muy bajo costo o incluso gratuito para el usuario. Organizaciones internacionales como la Free Software Foundation promueven esta tendencia y han logrado una difusión y proliferación de impacto en el mercado.

La Ingeniería de Software contribuye a estas expectativas, sino y avanza en apoyo de actividades complementarias al desarrollo de software, proporcionando metodologías, herramientas y técnicas para la planeación de la producción de software, el análisis y diseño de sistemas, apoyo a servicios como asesoría y capacitación, implantación y mantenimiento de sistemas de información y otras aplicaciones[2].

Tal como queda de manifiesto en el SWEBOK - “Software Engineering Body of Knowledge”, es preciso entonces impartir a los ingenieros de sistemas los conceptos, herramientas y habilidades que existen en la literatura acumulada durante los últimos treinta años respecto a

la construcción de software de tal suerte que los futuros profesionales se desempeñen con éxito en las cada vez más exigentes condiciones para el desarrollo de software.

12. PROPOSITOS:

Un primer propósito del curso es que el estudiante se familiarice con la trayectoria histórica del desarrollo de software, para que identifique los principales aciertos y desaciertos que a nivel mundial se conocen y comprenda la evolución de la ingeniería de software y su reconocimiento con una disciplina emergente.

En seguida es fundamental que el estudiante sea informado sobre cómo se llevan a cabo hoy en día los procesos de desarrollo de software en el sector real. Esto motiva la necesidad de adquirir los conocimientos y desarrollar las habilidades propias de la ingeniería de software, a saber:

CONCEPTOS

- Ingeniería de software
- Ciclo de vida del software
- Las cuatro dimensiones del software



- Modelos, paradigmas y filosofías de desarrollo de software.
- RUP – Rational Unified Process
 - Fases.
 - Disciplinas.
 - Iteraciones.

13. COMPETENCIAS QUE DESARROLLARA EL CURSO:

- Identificar los conceptos de la Ingeniería de Software.
- Describir las fases para el desarrollo de software.
- Determinar los requerimientos del software.
- Crear los artefactos propios de cada fase de RUP, con el uso de una herramienta.
- Diseñar, construir y ejecutar pruebas al software desarrollado.
- Producir software coherente con los requerimientos, el análisis y el diseño.

14. DIMENSIONES DE LAS COMPETENCIAS QUE DESARROLLA:

COGNITIVA	PRAXIOLOGICA	ACTITUDINAL
-----------	--------------	-------------

<p>Identifica las etapas del ciclo de vida del software, clasifica las metodologías de desarrollo de software, asocia una metodología a una necesidad particular, identifica las fases, disciplinas e iteraciones del RUP e identifica los aspectos a tener en cuenta para realizar una gestión de cambios y de proyecto.</p>	<p>Levantarlos requerimientos, les analiza y diseña una solución de software.</p> <p>Utiliza UML para crear artefactos en cada una de las fases de RUP.</p> <p>Utiliza herramientas de software libre para apoyar los procesos de desarrollo de software.</p> <p>Escribe en java el software solución.</p>	<p>Establece comunicación con los interesados para lograr acuerdos en los requerimientos y la solución de software.</p> <p>Gestiona los recursos disponibles para lograr la solución en los límites establecidos (tiempo, costos, recursos).</p>
---	--	--

15. UNIDADES TEMATICAS:

UNIDAD TEMATICA	ESTRATEGIA PEDAGOGICA	RECURSOS PEDAGOGICOS	TIEMP. PRES.	TIEMP. INDEP.
1. CONCEPTOS BÁSICOS.	Clase magistral Lectura con preguntas orientadoras.	Salón de clase. Texto Bibliografía. Presentación en powerpoint de motivación.	2	4
	Laboratorio	Editor de texto Editor de mapa de conceptos. Plataforma virtual.	2	4
2. PROCESOS DE SOFTWARE a. Modelo de procesos b. Ciclo de vida del software c. Estándar IEEE sobre Proceso Software	Clase magistral Práctica Proyecto	Salón de clase Biblioteca sala de sistemas	4	8
3. PARADIGMA OO, UML Y EL PROCESO UNIFICADO. a. FASES b. DISCIPLINAS c. ITERACIONES	Clase magistral Lectura con preguntas orientadoras.	Salón de clase. Texto Bibliografía. Presentación en powerpoint de motivación.	6	12
	Laboratorio	Editor de texto Editor de mapa de conceptos. Plataforma virtual.	6	12
4. FASE INICIO Modelado del negocio Requerimientos El proceso de requerimientos Análisis y Diseño El proceso de Análisis OO	Clase magistral Lectura con preguntas orientadoras.	Salón de clase. Texto Bibliografía. Presentación en powerpoint de motivación.	6	12
	Laboratorio	Editor de texto Editor de mapa de conceptos. Plataforma virtual.	6	12
5. FASE ELABORACIÓN Revisión de las anteriores, énfasis en: Análisis Y Diseño El proceso de Diseño OO Implementación Pruebas	Clase magistral Lectura con preguntas orientadoras.	Salón de clase. Texto Bibliografía. Presentación en powerpoint de motivación.	8	16
	Laboratorio	Editor de texto Editor de mapa de conceptos. Plataforma virtual.	8	16
6. FASE CONSTRUCCIÓN Revisión de las anteriores, énfasis en: Implementación Pruebas	Clase magistral Lectura con preguntas orientadoras.	Salón de clase. Texto Bibliografía. Presentación en powerpoint de motivación.	8	16

Entrega	Laboratorio	Editor de texto Editor de mapa de conceptos. Plataforma virtual.	8	16
TOTAL (Horas):			64	128

16. SISTEMA DE EVALUACION

Por su naturaleza de curso teórico-práctico, además del dominio conceptual, el estudiante deberá exhibir habilidades instrumentales consecuentes con las prácticas de laboratorio realizadas. Por lo anterior, el curso tendrá **valoraciones teóricas y prácticas**.

Dada la tendencia en el desarrollo del software, que obliga a la entrega pronta de software para su revisión por los interesados y a que se repitan estas entregas en iteraciones que conducen a la solución completa, es indispensable que durante el periodo académico el estudiante desarrolle un proyecto de desarrollo de software que simule en alguna medida este proceso real de varias iteraciones y así se habilite para enfrentar el mundo real profesional.

TRABAJO INDIVIDUAL 10%

Tareas y Quices

PARCIALES – presentación individual.

Parcial 1 15%

Parcial 2 15%

LABORATORIOS – presentación individual.

Global 20%

Entrega completa 5

Entrega incompleta 3

No entrega. 0

PROYECTO DE SEMESTRE 20%

Aplicación de escritorio en arquitectura de tres capas. Su desarrollo es grupal.

FASE 1 – Iniciación en una sola iteración. 5%

FASE 2 – Elaboración en dos iteraciones.

Iteración 1 5%

Iteración 2 5%

FASE 3 – Construcción en dos iteraciones. 5%

EXAMEN FINAL

Consiste en una ampliación de proyecto de semestre, su presentación es individual.

TEÓRICO. 10%

PRÁCTICO. 10%

17. FUENTES BIBLIOGRAFICAS:

- [1] R. Pressman S., Ingeniería de Software, enfoque práctico, 5 ed.
- [2] M. Piattini, J. Calvo-Manzano , J. Cervera y L. Fernández, Análisis y diseño de aplicaciones informáticas de gestión Una perspectiva de la ingeniería de software., Alfaomega, 2006.
- [3] R. J., Object-Oriented Modeling and Design.
- [4] Pfleeger, Ingeniería de Software.
- [5] I. Sommerville, Ingeniería de Software Orientado a Objetos, Pearson Education.
- [6] Bruegge, Ingeniería de Software Orientado a Objetos, Pearson - Addison Wesley Longman.

17.1 FUENTES DE CONSULTA BÁSICA

- [6] Bruegge, Ingeniería de Software Orientado a Objetos, Pearson - Addison Wesley Longman.

17.2 FUENTES DE PROFUNDIZACIÓN

- [1] R. Pressman S., Ingeniería de Software, enfoque práctico, 5 ed.
- [2] M. Piattini, J. Calvo-Manzano , J. Cervera y L. Fernández, Análisis y diseño de aplicaciones informáticas de gestión Una perspectiva de la ingeniería de software., Alfaomega, 2006.
- [3] R. J., Object-Oriented Modeling and Design.
- [4] Pfleeger, Ingeniería de Software.
- [5] I. Sommerville, Ingeniería de Software Orientado a Objetos, Pearson Education.

18. RECURSOS Y MEDIOS TECNOLÓGICOS:

- Sala de Sistemas
- Software libre
- STARUML para el modelado.
- JAVADOC para la documentación
- JUNIT – para las pruebas.

19. RECURSOS HUMANOS (Equipo profesional que participará en el desarrollo del curso)

TIPO	PERFIL	FUNCIÓN
Profesor	Ingeniero de sistemas, posgrado en ingeniería de software, preferiblemente maestría o doctorado.	Dirigir la teoría y la práctica, Mantener vigente el micro-curriculum del curso. Administrar al monitor y al auxiliar docente. Diseñar estrategias de aprendizaje y crear, publicar y evaluar los instrumentos para ellas. Participar en la construcción permanente de la área y línea de profundización en ingeniería de software.
Estudiante Monitor	Estudiante de 6° semestre o más, con desempeño sobresaliente en el curso y habilidades logísticas.	Apoyar el proceso de formación en lo referente a la preparación de las salas, acompañar al profesor en los laboratorios y en los procesos de instalación de herramientas de software requeridas en el curso.
Auxiliar Docente	Estudiante de 8° semestre o más, con desempeño sobresaliente en el curso y habilidades en gestión del conocimiento.	Apoyar el diseño, elaboración, puesta en uso y evaluación de los instrumentos requeridos para la práctica docente. Elaborar casos de estudio que sirvan de muestra para el desarrollo de tareas, hojas de trabajo, laboratorios y proyectos de semestre. Apoyar al profesor titular en el seguimiento al curso.