



**Proceso: Formulación del Currículo
y Plan de Estudios**

Guía de Cátedra

Código:	DOC11-FO-01
Versión:	2
Fecha:	05/07/2017
Hoja:	Página 1 de 4

1. Identificación del Curso/ Módulo

Nombre del Curso/ Módulo: INGENIERÍA DE REQUERIMIENTOS Y EL DISEÑO DE SOFTWARE	Línea de conocimi ento SOFT	Código materia: 18016	Crédito: 4	Horas totales 192	Horas Clase	48
					Horas Independent es	144

Facultad/ Departamento **Facultad de Ingeniería**

Programa que Administra el curso o módulo **Maestría en Gestión, Aplicación y desarrollo de software**

Niveles de Formación	Técnico Profesional		Especialización
	Tecnológico Profesional		Maestría X
	Profesional		Doctorado

Modalidad Presencial Dual Virtual

Fecha de actualización de la guía: **21-01-2019**

2. Restricciones de:

Conocimiento	Orden
Prerrequisito	
Correquisitos	

3. Justificación

La Ingeniería de Requerimientos (IR) cumple un papel primordial en el proceso de producción de software, ya que se enfoca a un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, las necesidades de los usuarios o clientes; de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de los requerimientos en el desarrollo de sistemas.

Por otra parte, el desarrollo de software tiene que seguir un proceso de análisis y diseño que proporcione las bases bajo las cuales se va a desarrollar la aplicación. En este contexto los enfoques para el análisis y diseño de software son de total pertinencia dando importancia a aquellos que en su momento han permitido mejorar substancialmente el desarrollo del software, como lo es el de orientación a objetos, así como a otros enfoques aplicables al desarrollo de software.

4. Competencias de formación

Competencia	Resultado de aprendizaje esperado
Entiende de manera clara el problema que se solucionará con el software	Se comunica efectivamente en forma oral y escrita para lograr desarrollar entrevistas con clientes y usuarios finales, con el fin de obtener los requerimientos del software.
Aplica las fases de la ingeniería de requerimientos	Analiza y diseña software bajo el paradigma de orientación a objetos Diseña software bajo el paradigma de agentes (Mediante la aplicación de la metodología de ADOA vista en clase el estudiante obtiene productos específicos que serán evaluados por el docente)
Alinea lo aplicado en la ingeniería de requerimientos con la fase de diseño de software y su posterior desarrollo e implementación	Conceptualiza sobre el desarrollo de software el enfoque de orientación por modelos.

5. Contenido de la actividad académica*			
Unidad	Temáticas	Tiempo Semanas	Evaluación del aprendizaje
Introducción a requerimientos	<ul style="list-style-type: none"> • Problemática de Requerimientos • Qué son requerimientos • Características de Requerimientos Ingeniería de Requerimientos 	0.33	Taller
Negocio como base de los requerimientos	<ul style="list-style-type: none"> • Introducción • Modelado básico del Negocio • Diagramas de Actividades • Arquitectura de Negocio 	0.33	Ejercicio
Fundamentos de arquitectura empresarial	<ul style="list-style-type: none"> • Qué es arquitectura y arquitectura empresarial • Impactos de la arquitectura empresarial de la organización • Complejidad de la arquitectura • La esencia de la Arquitectura Empresarial • Modelos y semántica 	0.33	Ejercicio
Captura de requerimientos	<ul style="list-style-type: none"> • Proceso de Requerimientos • Identificación de necesidades 	0.33	Ejercicio
Técnicas de levantamiento de requerimientos	<ul style="list-style-type: none"> • Técnicas de levantamiento de requerimientos • Técnicas de validación de requerimientos • Roles y Responsabilidades • Lenguaje Unificado de Modelado 	0.33	Ejercicio
Conceptualización de Casos de Uso	<ul style="list-style-type: none"> • Qué son casos de uso y actores • Puntos de chequeo 	0.33	Ejercicio
Estructuración Casos de Uso	<ul style="list-style-type: none"> • Relaciones entre actores • Relaciones entre casos de uso • Empaquetamiento 	0.33	Ejercicio
Estimación de requerimientos	<ul style="list-style-type: none"> • Puntos de Casos de Uso • Estimación de cronograma 	0.33	Ejercicio
Estimación de requerimientos	<ul style="list-style-type: none"> • Administración de requerimientos • Manejo de métricas • Trazabilidad 	0.33	Ejercicio
Control de cambios	<ul style="list-style-type: none"> • Control de cambios • Buenas prácticas de requerimientos 	0.33	Ejercicio

Análisis y diseño orientado a objetos	<ul style="list-style-type: none"> • Ejemplos aT3, Bogor, GMoDS • Lectura libros Google • Patrones de Diseño • Lectura libros Google 	0.33	Ejercicio
Recursos: <ul style="list-style-type: none"> • 1) Start UML. • 2) Eclipse 3.7 • 3) Visual Paradigm for Eclipse (Community Edition) • 4) JPA (Dali Java Persistency Tools) • 5) MySQL Server 			
6. Estrategias Pedagógicas			
<ol style="list-style-type: none"> 1. Para el desarrollo personal y de pensamiento. 2. Para facilitar el aprendizaje y desarrollo de habilidades. 			
Otra:			
7. Evaluación y Registro de resultados			
Evaluar: Evaluación por parte del profesor: Trabajo aplicado en clase Autoevaluación y Coevaluación Participación en Trabajo aplicado en clase Trabajo aplicado en clase			
Calificar: La forma de calificación es numérica (de 0.0 a 5.0) considerando rúbricas preestablecidas para cada momento de evaluación. Taller 20% Ejercicios de clase 40% Trabajo aplicado 40%			
Registro: Se hace registro de la calificación desde COSMOS una vez finalizado el módulo y valorados los entregables de los estudiantes.			
8. Referencias Bibliográficas			
Bibliografía UNAB			Notación topográfica
PRESSMAN, Roger S. Ingeniería del Software un enfoque práctico. 6ª. Edición. Editorial McGraw Hill. 2002.			005.1 P935i
B.S.Ainapure. Object Oriented Modeling And Design, Technical Publications, 2010 - 244 páginas			

Gandharba Swain. Object-Oriented Analysis and Design Through Unified Modeling Language. Laxmi Publications, Ltd., 2010 - 225 páginas	
Bernd Oestereich. Developing software with UML: objectoriented Analysis and design in practice. Addison-Wesley, 2002 - 299 páginas	
PIATTINI, Mario. Análisis y Diseño detallado de Aplicaciones Informáticas de Gestión. Alfa Omega Grupo Editor. 2000	005.1 A532
David Ruble. Análisis y Diseño de Sistemas. Editorial Prentice Hall 1997/98.	
Jeffrey Whitten. Análisis de Sistemas de Información. Editorial IRWIN, 1996	
Steve McConnell. Desarrollo y Gestión de Proyectos Informáticos. McGrawHill. 2000.	
Eduard Yourdon. Análisis Estructurado moderno. Editorial Prentice Hall. 1989.	
Donald G. Firesmith, Brian Henderson-Sellers. The OPEN process framework: an introduction. Addison-Wesley, 2002 - 330 páginas	
Brian Henderson-Sellers, Paolo Giorgan. Agent-oriented methodologies. Idea Group Inc (IGI), 2005 - 413 páginas	
Federico Bergenti, Marie-Pierre Gleizes, Franco Zambonelli. Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook. Springer, 2004 - 505 páginas	
Anneke G. Kleppe, Jos B. Warmer, Wim Bast. MDA explained: the model driven architecture : practice and promise Addison-Wesley Professional, 2003 - 170 páginas	
David Frankel. Model driven architecture: applying MDA to enterprise computing. Wiley, 2003 - 328 páginas.	
9. Observaciones	