

Asignatura	Laboratorio de Software
Código	IS873
Créditos	3
Intensidad semanal	4 horas
Requisitos	IS714
Tipo de asignatura	Práctica

Justificación	<p>El gobierno nacional declaró el software como un sector de clase mundial y es compromiso de las Universidades, de las organizaciones públicas y privadas y en sí, del talento humano que lo componen, aportar al fortalecimiento de este sector.</p> <p>Es importante resaltar que en el perfil profesional del ingeniero de Sistemas y Computación se considera que el egresado podrá “Identificar y resolver retos y problemas informáticos y administrativos” y “Utilizar herramientas computacionales para solucionar problemas de Ingeniería”, así como, en el perfil ocupacional está definido que estará en capacidad de “Enfrentar desafíos que conlleven investigación tecnológica en diversas especialidades de la Ingeniería”¹; y en los objetivos de la asignatura está definido “Diseñar, implementar y probar el software mediante la utilización de metodologías, estándares y herramientas computacionales de actualidad.”, lo que justifica que en la asignatura Laboratorio de software la cual es práctica, los estudiantes la utilicen metodologías y herramientas tecnológicas acordes con las requeridas en el campo empresarial.</p>
Objetivo general	Utilizar los conocimientos teóricos y prácticos adquiridos durante la formación profesional para resolver un problema del entorno, aplicando la ingeniería de software y

¹ <http://isc.utp.edu.co/informacion/ingenieria-de-sistemas-y-computacion-.html>

	llevando a cabo las fases que conforman el ciclo vida del software.
Objetivos Específicos	<ul style="list-style-type: none"> • Utilizar la ingeniería de software y las herramientas computacionales para resolver problemas del entorno. • Obtener, analizar y especificar requerimientos funcionales y no funcionales de un proyecto de software. • Diseñar la arquitectura del software considerando los objetivos de negocio y los requerimientos no funcionales, mediante la utilización una metodología. • Diseñar, implementar y probar el software mediante la utilización de metodologías, estándares y herramientas computacionales de actualidad. • Gestionar un proyecto de software bajo una metodología de desarrollo utilizada en el contexto empresarial, asumiendo con responsabilidad y compromiso cada uno de los roles definidos.

Competencias Genéricas	<ul style="list-style-type: none"> - Aprendizaje autónomo - Capacidad de análisis y síntesis - Capacidad de aplicar los conocimientos a la práctica - Resolución de problemas - Trabajo individual y en equipo - Comunicación oral y escrita - Iniciativa y participación para la toma de decisiones
Competencias específicas	<p>Cognitivas (Saber):</p> <ul style="list-style-type: none"> - Metodologías aplicadas al TI - Idioma inglés - Nuevas tecnologías TIC - Conocimientos de informática - Conocimientos en las áreas de la ingeniería del software

	<ul style="list-style-type: none"> - Conocimientos en la gestión de proyectos. <p>Procedimentales / Instrumentales (Saber hacer):</p> <ul style="list-style-type: none"> - Redacción en interpretación de documentación técnica - Estimación y programación del trabajo - Planificación, organización y estrategia. - Capacidad de aplicación de metodologías - Dirección de equipos de trabajo - Dirección de proyectos - Capacidad para buscar información - Capacidad para resolver problemas y no crearlos <p>Actitudinales (Ser):</p> <ul style="list-style-type: none"> - Liderazgo - Compromiso - Responsabilidad - Respeto - Tolerancia - Tolerancia al stress - Trabajar bajo stress - Iniciativa y participación - Facilidad de comunicación - Actitud positiva - Administración el tiempo - Facilidad para adaptarse a los cambios y a nuevas situaciones - Compromiso de trabajar con calidad
--	---

Contenido de la asignatura	
Unidad 1	Contextualización e importancia de la práctica en los proyectos de software
Unidad 2	Definición, delimitación y planeación general del proyecto de software.
Unidad 3	Ejecución y gestión del proyecto del proyecto de software

Bibliografía

- Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. 1ª Edición, España, Addison Wesley, 1999. ISBN: 84-7829-036-2.
- Booch, Grady. Rumbaugh, James. Jacobson, Ivar. El Lenguaje Unificado de Modelado UML. 1ª Edición, España, Addison-Wesley Iberoamericana, 1999. ISBN: 84-7829-028-1.
- Larman, Craig. UML y Patrones: Introducción al análisis y diseño orientado a objetos. 1ª Edición, México, Prentice-Hall, 1999. ISBN: 970-17-0261-1.
- Sommerville, Ian. Ingeniería de Software. 6ª Edición, México, Pearson Educación de México S. A. De C. V., 2002. ISBN: 970-26-0206-8.
- Booch, Grady. Análisis y Diseño Orientado a Objetos con aplicaciones. 2ª Edición, EEUU, Addison-Wesley/Diaz de Santos, 1996. ISBN: 0-201-60122-2.
- Rumbaugh, James. Blaha, Michael. Premerlani, William. Eddy, Frederick. Lorensen, William. Modelado y Diseño Orientado a Objetos: Metodología OMT. 1ª Edición, España, Prentice-Hall Interamericana, 1996. ISBN: 013-240698-5.
- Pfleeger, Shari Lawrence. Ingeniería de Software: Teoría y Práctica. 2ª Edición, Argentina, Pearson Education S. A., 2002. ISBN: 987-9460-71-5.
- Cohn, Mike. Morgan, Bryan. Morrison, Michael. Nygard, Michael t. Joshi, Dan. Trinko, Tom. JAVA Developer's Reference. 1ª Edición, EEUU, Sams Net, 1996. ISBN: 1-57521-129-7.
- Meyer, Bertrand. Construcción de Software Orientado a Objetos. 2ª Edición, España, Prentice-Hall, 1999. ISBN: 84-8322-040-7.
- Arquitectura de software. Software Engineering Institute
- PMBOK
- SWEBOK