

| | |
|---------------------------|--|
| Asignatura | Ingeniería del Software I |
| Código | IS714 |
| Créditos | 4 |
| Intensidad semanal | 6 Horas |
| Requisitos | IS663 Bases de Datos I IS463 Teoría General de Sistemas |

| | |
|-----------------------------|--|
| Justificación | <p>El Ingeniero de Sistemas y Computación debe tener la capacidad para desarrollar grandes proyectos de software en los cuales se contruyen productos de software de tamaño industrial. Una parte de esta capacidad se desarrolla en este curso.</p> <p>Primero que todo el ingeniero necesita tener un panorama de todos los temas que cubre la ingeniería del software. Este panorama se obtiene estudiando un estándar internacional llamado SWEBOK (Software Engineering Body Of Knowledge).</p> <p>En segundo lugar, el proceso de ingeniería incluye las siguientes etapas: Estudio preliminar, planeación del proyecto, requerimientos, diseño, construcción, pruebas, entrega, producción y mantenimiento. En éste curso se tratará las etapas de requerimientos y pruebas. Las otras etapas se han distribuido en otras asignaturas.</p> <p>Lo que el estudiante debe aprender en esta asignatura es un conjunto de competencias laborales, es decir, con lo aprendido podrá realizar trabajos que son bien remunerados en la industria del software tales como el levantamiento de requerimientos, la descripción de casos de uso, el análisis o modelamiento de los requerimientos mediante diagramas UML, establecer los requerimientos no funcionales de un producto de software.</p> |
| Competencias Previas | <ul style="list-style-type: none"> • Conocer sobre Teoría General de Sistemas • Conocer sobre programación de computadoras en sus diferentes paradigmas. |

| | |
|------------------------------|---|
| | <ul style="list-style-type: none"> • Competencia para leer documentos sobre esta asignatura en inglés. • Conocer sobre bases de datos relacionales y su diseño. |
| Objetivo general | Al terminar el curso, el estudiante estará en capacidad para desarrollar la fase de requerimientos de grandes proyectos de software y la fase de pruebas del software. |
| Objetivos Específicos | <ol style="list-style-type: none"> 1. El estudiante conocerá cuales son las KA (áreas de conocimiento) de la ingeniería del software especificadas en el SWEBOOK. 2. El estudiante conocerá las fases de un proceso de desarrollo de software. 3. El estudiante adquirirá la capacidad para realizar la fase de requerimientos de un proyecto de desarrollo de software. 4. El estudiante adquirirá la capacidad de realizar la fase de pruebas del software. 5. El estudiante manejará los conceptos y el lenguaje específico de la asignatura, podrá entender los textos de la materia y escribir sobre el tema. |
| Metodología | <p>El profesor orientará la asignatura con clases magistrales y planteará lecturas y temas de investigación.</p> <p>Se discutirán los documentos de apoyo, lecturas planteadas y temas de investigación en clase.</p> <p>Se realizarán tres grandes evaluaciones: Parcial 1, Parcial 2 y Final. Cada una de estas evaluaciones se puede dividir en varias evaluaciones de menor valor (quices, trabajos, exposiciones, etc.) teniendo en cuenta la autonomía de cada profesor.</p> <p>ESTRATEGIAS DE APRENDIZAJE: Constructivismo: Se construirán nuevos conocimientos teniendo como base los conocimientos que el estudiante ya ha construido.</p> |

| | |
|---------------------------------|---|
| | Aprendizaje Significativo: Se hará énfasis en los conocimientos más significativos e importantes. |
| Competencias Genéricas | <p>Requeridas:</p> <ul style="list-style-type: none"> - Lectura en Inglés. - Comprensión de lectura. - Redacción. - Pensamiento sistémico. <p>A adquirir:</p> <ul style="list-style-type: none"> - Capacidad de Síntesis. - Comprensión del proceso de la ingeniería. - Aplicación de conocimientos científicos y técnicos a la resolución de problemas. |
| Competencias específicas | <p>REQUERIDAS:</p> <p>Cognitivas:</p> <ol style="list-style-type: none"> 1. Inglés. 2. Programación de computadores. 3. Bases de datos. 4. Herramientas de desarrollo IDE. (Ambientes Integrados de Desarrollo). 5. Cómo funcionan las empresas, sus procesos y necesidades. 6. Probabilidades. <p>Procedimentales/Instrumentales:</p> <ol style="list-style-type: none"> 1. Programación Orientada a Objetos. 2. Teoría General de Sistemas y los cinco aspectos de un |

| | |
|--|--|
| | <p>sistema: Variables, actividad en el tiempo, comportamiento, estructura, estados y transiciones.</p> <p>Actitudinales:</p> <ol style="list-style-type: none">1. Pensamiento lógico matemático para afrontar los problemas.2. Actitud creativa, curiosidad por explorar nuevas posibilidades.3. Método científico. <p>A ADQUIRIR:</p> <p>Cognitivas:</p> <ol style="list-style-type: none">1. Conocer el ciclo de la ingeniería: requerimientos, diseño, construcción, pruebas, entrega, producción y mantenimiento.2. Conocer los fundamentos sobre requerimientos del software.3. Conocer los fundamentos sobre pruebas de software. <p>Procedimentales/Instrumentales:</p> <ol style="list-style-type: none">1. Modelamiento UML de los casos de uso mediante los siguientes diagramas: Casos de uso, secuencia, comunicación, estados, actividades, clases.2. Métodos de levantamiento de requerimientos.3. Elaboración de casos de uso.4. Obtención de los requerimientos no funcionales de un producto de software.5. Tipos de pruebas de software y métodos para elaborar pruebas. |
|--|--|

| | |
|--|--|
| | <p>Actitudinales:</p> <ol style="list-style-type: none"> 1. Pensamiento sistémico para analizar un problema. 2. Actitud investigadora en la búsqueda de requerimientos. 3. Actitud empresarial y de negocios. |
|--|--|

| Contenido de la asignatura | |
|-----------------------------------|--|
| | <p>INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE</p> <ol style="list-style-type: none"> 1 ¿Qué es ingeniería? 2 El proceso de la ingeniería <ol style="list-style-type: none"> 2.1 Requerimientos 2.2 Diseño 2.3 Construcción 2.4 Pruebas 2.5 Entrega 2.6 Producción 2.7 Mantenimiento 3 Metodologías de desarrollo de software <ol style="list-style-type: none"> 3.1 El proceso <ol style="list-style-type: none"> 3.1.1 Ciclo de vida en cascada 3.1.2 Ciclo de vida estructurado 3.1.3 Ciclo de vida RAD 3.1.4 Ciclo de vida en espiral 3.2 El modelamiento 4 Conceptos sobre teoría general de sistemas <ol style="list-style-type: none"> 4.1 Variables del problema 4.2 Actividad en el tiempo 4.3 Comportamiento 4.4 Estructura 4.5 Estados y Transiciones 5 El SWEBOK <ol style="list-style-type: none"> 5.1 La ingeniería del software como una profesión 5.2 Las áreas de conocimiento |
| Unidad 1 | |

| | |
|-----------------|--|
| Unidad 2 | <p>REQUERIMIENTOS DEL SOFTWARE</p> <ol style="list-style-type: none"> 1 El capítulo de requerimientos del SWEBOK 2 Levantamiento de requerimientos (elicitación) <ol style="list-style-type: none"> 2.1 Entrevistas 2.2 Prototipos 2.3 Encuestas y Muestreo 2.4 Observación directa 2.5 Reuniones 3 Definición de los requerimientos mediante casos de uso. 4 Análisis de casos de uso: <ol style="list-style-type: none"> 4.1 Diagramas de casos de uso 4.2 Diagramas de secuencia 4.3 Diagramas de comunicación (antes colaboración) 4.4 Diagramas de estados 4.5 Diagramas de actividades 4.6 Diagramas de clases 4.7 Diccionario de clases 5 Los requerimientos no funcionales |
| Unidad 3 | <p>PATRONES DE DISEÑO GRASP</p> <ol style="list-style-type: none"> 1 Experto en información 2 Creador 3 Bajo acoplamiento 4 Alta cohesión 5 Controlador |
| Unidad 4 | <p>PRUEBAS DEL SOFTWARE</p> <ol style="list-style-type: none"> 1 Introducción 2 Pruebas de desarrollo <ol style="list-style-type: none"> 2.1 Pruebas de unidad 2.2 Elección de los casos de prueba de unidad 2.3 Prueba de componentes 2.4 Pruebas del sistema 3 Prueba de versiones <ol style="list-style-type: none"> 3.1 Pruebas basadas en requerimientos 3.2 Pruebas de escenario 3.3 Pruebas de rendimiento 4 Pruebas de usuario |

| | |
|--|------------------------|
| | 5 La prueba del camino |
|--|------------------------|

| | |
|-------------------|---|
| Texto Guía | No hay un solo texto, se siguen varios. |
|-------------------|---|

| Referencia | Bibliografía |
|-------------------|---|
| <i>BOU14</i> | Bourque, Pierre. Fairley, Richard. SWEBOK. Versión 3.0, 2014. IEEE Computer Society. ISBN 978-0-7695-5166-1. |
| <i>KLI81</i> | Klir, George. Teoría General de Sistemas (Un enfoque Metodológico). Editoria Ice, 1981. ISBN 84-7085-104-7. |
| <i>MAS75</i> | Maserovic. Takahara. General Systems Theory: Mathematical Foundations. Academic Press Inc. New York. 1975. ISBN 0-12-491540-X. |
| <i>KEN97</i> | Kendall, Keneth E. Kendall, Julie E. Análisis y Diseño de Sistemas. 3ª Edición, México, Prentice Hall Hispanoamericana, 1997, ISBN: 968-880-694-4. |
| <i>LAR04</i> | Larman, Craig. Applying UML and Patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development. 3ª Ed. Prentice Hall, 2004. ISBN 0-13-148906-2. |
| <i>SOM11</i> | Sommerville, Ian. Software Engineering. 9ª Ed. 2011. Pearson Education. ISBN 978-607-32-0603-7. |
| <i>PRE01</i> | Pressman, Roger. Software Engineerion: A Practitioner's Approach. 5ª Ed. Mc Graw Hill Higher Education. 2001. ISBN 0-07-365578-3. |
| <i>BOO07</i> | Booch, Grady et Al. Object Oriented Analysis and Design with Applications. 3ª Ed. Addison Wesley. 2007. ISBN 0-201-89551-X. |
| <i>GOR06</i> | Gorton, Ian. Essential Software Architecture. 2006. Springer Verlag Berlin Heidelberg. 2006. ISBN 973-3-540-28713-1. |



UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERIAS
Programa Ingeniería de Sistemas y Computación

