|  |  |  |  |
| --- | --- | --- | --- |
| **Espacio Pedagógico:** INGENIERÍA DE SOFTWARE | | | |
| **Código:** | | | |
| **Unidad Académica Responsable:** Vicerectoría Académica | | | |
| **Carrera según grado:** Profesor de informática Con orientación en Robótica o Diseño Gráfico en el grado de Licenciatura. | | | |
| **Requisitos (código, nombre y unidades valorativas):**  Sistemas de información | | | |
| **Distribución de la actividad académica del espacio pedagógico:** | | | |
| **Total de unidades valorativas:** 4  **Teóricas:** 3  **Práctica:** 1 | **Número de semanas:** 13  **Horas por semana Teóricas:** 45  **Horas por semana Prácticas:** 45 | Horas de clase frente al profesor en el período académico y durante la **semana:**  H.T.: 45/3  H.P.: 45/1 | Horas de Trabajo Independiente en el período académico y durante la **semana:** 90 |
| **Modalidad en la que se presenta el proceso de aprendizaje:**   1. **Presencial**  * Herramienta de apoyo a la presencialidad (plataforma)  1. **Distancia**  * Semipresencial (e-Learning) * Semipresencial con mediación virtual (b-Learning) * Virtual | | | |
| **Descripción del espacio pedagógico (Naturaleza y propósito):**  Este espacio se caracteriza por introducir al estudiante a un conocimiento fundamental y riguroso para la construcción de software; además les permitirá a los estudiantes conocer enfoques de mantenimiento y evolución de sistemas de software de gran tamaño y/o complejidad, haciendo uso de herramientas especializadas para el desarrollo de software. Resaltando las diferentes etapas involucradas en el desarrollo y la manutención de software.  La propuesta de este espacio propone vinculaciones teórica-prácticas, en el cual el estudiante incrementará sus habilidades y conocimientos de manera efectiva y estratégica; misma que se traducirá en la posibilidad de gestionar y participar en proyectos de software. La integración de los conocimientos previos con el conocimiento adquirido se hará en un ambiente colaborativo. | | | |
| **Capacidades previas (conocimientos, habilidades, destrezas, valores adquiridos por los estudiantes):**   1. Plantea acciones estratégicas utilizando los sistemas de información dentro de la institución 2. Identifica los elementos de la infraestructura actual de las organizaciones y es capaz de, siempre que se requiera, sugerir mejoras a la misma para que los servicios tecnológicos se encuentren disponibles con la mejor de las prestaciones. 3. Analiza y diseña un sistema de información para ejecutar los procesos de la gestión empresarial. 4. Promueve, bajo un plan de trabajo y metodología de desarrollo adecuados, el desarrollo de sistemas de información para mejorar situaciones actuales dentro de las instituciones. | | | |
| **Competencias genéricas:**   1. Conocer y aplicar los conceptos fundamentales asociados a la Ingeniería de Software. 2. Desarrollar una cultura de aplicación de los principios de ingeniería en proyectos de software. 3. Crear, diseñar, desarrollar y operar aplicaciones informáticas basándose en principios de Ingeniería y estándares de 4. calidad. 5. Aplicar estándares de calidad en el desarrollo y evaluación de soluciones informáticas. | | | |
| **Competencias específicas:**   1. Reconoce y describe los distintos tipos de componentes electrónicos básicos (diodo, diodo zener, transistor bipolar, etc.). 2. Profundiza en los elementos que integran los circuitos electrónicos. 3. Desarrolla los métodos de cálculo de la ganancia, impedancia de entrada, impedancia de salida y ancho de banda. 4. Comprende el funcionamiento del transistor bipolar y polarización de transistores para responder a las necesidades planteadas. 5. Categoriza y construye circuitos con transistores permitiendo conocer los diferentes sensores que dan solución a problemas industriales. 6. Identifica los sistemas y reglas o principios medulares que subyacen a una serie de fenómenos. | | | |
| **Subcompetencias:**   1. Diseñar e implementar aplicaciones informáticas 2. Seleccionar la metodología, entre las diversas opciones, la más apropiada para hacer la construcción; optimizando el 3. equilibrio entre diversos aspectos: capacidad técnica, costo, calidad, entre otros. 4. Representar, a través de diagramas, las necesidades del cliente, interacciones del usuario, arquitecturas a emplear, 5. entre otros. Para facilitar la comunicación con los otros miembros del equipo de desarrollo. 6. Identificar un marco de referencia común para validar los modelos con usuarios representativos, en función de los 7. modelos de desarrollo (por ejemplo, enfoque iterativo). 8. Optimizar el desarrollo, el mantenimiento y el rendimiento de las aplicaciones mediante el empleo de patrones de 9. diseño y la reutilización de soluciones probadas 10. Conceptualizar la calidad como un eje transversal a todas las actividades de la ingeniería de software | | | |
| **Áreas temáticas (unidades de aprendizaje o bloques):**  1. UNIDAD I: INTRODUCCIÓN A LA INGENIERÍA DE SW  1.1. El software y la ingeniera de software  1.2. Modelos del proceso  1.3. Mitos del software  1.4. Selección del modelo que mejor se ajuste a las condiciones de negocio y al problema a tratar.  2. UNIDAD II: MODELADO Y DISEÑO  2.1. UML  2.2. Comprensión de requerimientos  2.3. Modelos del software  2.4. Diseño de la arquitectura  2.5. Diseño de la interfaz  3. UNIDAD III: PATRONES DE DISEÑO  3.1. Definición de patrón y su aplicación  3.2. Análisis y diseño de patrones  3.3. Catálogo de patrones (creacionales, estructurales, de comportamiento, entre otros)  3.4. Aplicación de los patrones de diseño  4. UNIDAD IV: GESTIÓN DE LA CALIDAD  4.1. Calidad de proceso y producto  4.2. Planeación de la calidad  4.3. Administración de la configuración de software  4.4. Medición y Métricas de SW | | | |
| **Estrategias metodológicas de aprendizaje-enseñanza:**   1. Presencial:   Este espacio será desarrollado de manera participativa y reflexiva de acuerdo a los resultados del aprendizaje esperados, haciendo uso entre otras de las estrategias metodológicas siguiente: clases magistrales, lecturas y debate de investigaciones bibliográficas, trabajos individuales y en grupo, presentaciones en individuales y en grupo, uso de software afín a la temática, informes. Como complemente a la formación se realizan uso de recursos textuales (instructivos, guías, u otros), audio y video; además de las herramientas que las Tecnologías de la Información y Comunicación (TIC) permitan.   1. Distancia:   Se parte del aprendizaje autónomo, donde se utilizarán diversas estrategias de aprendizaje asignadas en la guía didáctica que se elabora por periodo académico y que los estudiantes desarrollan individualmente para cada una de los encuentros de acompañamiento tutorial de acuerdo a los resultados del aprendizaje esperados, haciendo uso entre otras de las estrategias metodológicas siguientes: seminarios, aprendizaje basado en problemas o proyectos, estudio de casos, elaboración de proyectos e informes, análisis de casos, análisis y crítica de textos, uso de software afín a la temática, informes de externos, clases magistrales tras trabajos prácticos y viceversa, tutorías sobre trabajos, trabajo en grupos colaborativos, exposiciones magistrales dialogadas. La asistencia académica individual o grupal de forma sincrónica y asincrónica que considera tanto herramientas, físicas o tecnológicas; como cognitivas se realiza por parte del Docente-Tutor a través de las diferentes Tecnologías de la Información y Comunicación (TIC), como mediación pedagógica que la UPNFM establezca. | | | |
| **Resultados de aprendizaje:**   1. Comprende el papel de la ingeniería de software y los retos que conlleva su aplicación en proyectos de desarrollo de sistemas. 2. Modela, diseña y construye sistemas de información aplicando principios de ingeniería. 3. Comprende y utiliza los distintos estándares de diseño para aplicarlos en sus actividades laborales, tomando como base el lenguaje UML y los principios de patrones de diseño de software 4. Desarrolla, mantiene y evalúa servicios y sistemas software que satisfaga los requerimientos funcionales y no funcionales. | | | |
| **Estrategias de evaluación de los aprendizajes (diagnóstica, formativa, sumativa):**   1. Presencial   Evaluación Diagnóstica: Se aplicará cuando el docente considere debe hacer una vinculación sobre los saberes previos y los nuevos aprendizajes de la unidad a la cual abordará, con el objetivo de detectar fortalezas y debilidades en los estudiantes y tomar las acciones pertinentes sea el caso presentado.  Evaluación Formativa: el docente debe monitorizar el aprendizaje del estudiante para proporcionar retroalimentación continua enfocada tanto en los saberes temáticos como en las destrezas y dominios que permitan al estudiantes mejorar su aprendizaje.  Evaluación Sumativa: ésta tiene la función de asegurar que las características de los estudiantes respondan a los resultados de aprendizaje esperados, por lo que cada docente debe establecer mecanismos fiables de la medición de los conocimientos y habilidades a evaluar.   1. Distancia   Las evaluaciones Diagnósticas, Formativa y Sumativa (antes descritas) son aplicables a la modalidad a Distancia; destacando que es fundamental el aprovechamiento de las TIC (plataformas, redes sociales, mensajería, entre otros) que permita una comunicación eficiente y oportuna con los estudiantes. | | | |
| **Referencias bibliográficas sugeridas:**   1. Básicas   Ingeniería en software. Un enfoque práctico Pressman, Roger S. Edición 7ma. 2010 McGraw-Hill   1. Complementarias   Análisis y diseño Orientado a Objetos de Sistemas. Usando UML. McGrawHill. Simon Bennett. ISBN: 978-844-815-640-4.  Análisis y diseño de sistemas Kendall, Kennth E - 2011 spa México: Prentice Hall | | | |
| **Recursos adicionales (revistas, Web, videos, películas, otros):** | | | |