

Datos de identificación			
Nombre del EE ¹ : Ingeniería de Software II		Área Formativa: Vocacional	
Departamento que da el servicio: Departamento de Ingeniería Industrial			
Clave:	Modalidad: Presencial/En línea		Idioma: Español
Horas totales al semestre: 80	Valor en créditos: 5		Semestre en que se cursa: VI
Carácter: Obligatoria	EE Antecedente: Ingeniería de software I	EE subsecuente: Ingeniería de software III	
Opciones de promoción: Calificación		Mecanismos alternativos de promoción: Equivalencia	
Presentación			
Este espacio educativo proporciona los conocimientos, habilidades y/o destrezas que forman al estudiante para el desarrollo de software en una forma disciplinada, siguiendo protocolos de trabajo, contrastando metodologías existentes contra las emergentes.			
Desempeños			
Competencias genéricas que se ejercitan		Unidades de competencia profesionales	
● Utiliza con eficiencia las tecnologías digitales para la comunicación y la gestión de información académica y profesional, en un entorno de trabajo colaborativo		5.3 Diseñar componentes del sistema (bases de datos, interfaces, estructuras de datos y procesos) en base a las necesidades del cliente.	
Resultados de Aprendizaje			
● Distinguir los diferentes principios básicos del desarrollo de software. ● Modelar los posibles escenarios de un software. ● Crear los diagramas de UML necesarios para el diseño de sistemas de software. ● Comparar las metodologías de desarrollo de software tanto las emergentes como las metodologías clásicas.			
Orientación didáctica			
El estudiante asistirá 80 horas semestrales (5 horas semanales) a clases presenciales dirigidas por el profesor donde se desarrollarán los conceptos y prácticas propias de un segundo curso de ingeniería de software y se evidenciará la flexibilidad y utilidad de este para modelar y resolver problemas de desarrollo de sistemas. Desde la perspectiva de organismo acreditador, la materia pertenece al área programación e ingeniería de software donde se busca darle al alumno, las capacidades y conocimientos en el desarrollo de software, utilizando distintos lenguajes y técnicas de programación. Tiene atributos de diseño, codificación y configuración de piezas o componentes de software. Propone soluciones innovadoras. Propone maneras de solucionar un problema o desarrollar un proyecto en equipo. Definiendo un curso de acción con pasos específicos.			
Actividades del estudiante		Actividades del profesor	
Horas/ semestre	Actividades	Horas/ semestr e	Actividades
80	● Asistencia y participación en clase.	80	● Impartir clases presenciales ● Definir las reglas de uso de algoritmos generativos de IA (GitHub, Copilot, ChatGPT, Gemini, etc.) en actividades de la materia que permitan utilizarlas como complemento para incrementar la

¹ Espacio Educativo = EE

			productividad, más no como la solución total de los problemas a resolver o tareas por entregar.
Evaluación del aprendizaje			
Criterios de cumplimiento	Evidencias de desempeño		Evidencias de conocimiento
1. Cumplir con la asistencia, puntualidad (Presencial o Virtual). 2. Entrega de trabajos (investigación, tareas, exámenes) y/o practicas a tiempo y siguiendo las especificaciones descritas.	- Realización de exámenes (en línea y / o en papel). - Entrega de tareas y trabajos en plataforma electrónica. - Exposiciones para profundizar o complementar temas en idioma inglés. - Desarrollo y presentación de proyecto final que integre todos los conceptos, métodos, técnicas y procesos revisados.		1. Realizar trabajos de análisis y modelado avanzado de requisitos de software (basado en escenarios, en objetos y clase responsabilidad-colaborador). 2. Realizar trabajos de modelado de un sistema aplicando diagramas UML. 3. Desarrollar un proyecto aplicando un modelo de proceso de desarrollo.
Técnicas e instrumentos de evaluación	Rúbricas para los exámenes, trabajos, investigación y proyecto.		
Recursos para la formación			
Contenidos básicos		Materiales	
1. Modelado de software 2. Modelado de los requerimientos en base a escenarios, información y clases de análisis 3. El lenguaje de modelado unificado – UML 4. Metodologías Emergentes Vs Metodologías Clásicas 5. Modelos Importantes 6. Reutilización de software		● Diagramas UML del sistema ● Documento de Especificación de Requerimientos ● Documento Final de análisis ● Documentos de análisis ● Documentos Electrónicos ● Material Audiovisual ● Recursos en la Nube	
Bibliografía			
● Kendall, K. E., & Kendall, J. E. (2023). Systems analysis and design (11th ed.). Prentice Hall. ● Dennis, A., Wixom, B., Roth, R.M. (2021) Systems Analysis and Design (8th ed.). Wiley ● Dennis, A., Wixom, B., Tegarden, D., (2020). Systems Analysis and Design: An Object-Oriented Approach with UML. Wiley. ● Orosz, G. (2024). The Software Engineer's Guidebook. Pragmatic Engineer BV. ● Wiegers, K., & Hokanson, C. (2024). Software Requirements Essentials: Core Practices for Successful Business Analysis. Addison-Wesley Professional. ● Hyde, R. (2020). Write Great Code, Volume 3: Engineering Software. No Starch Press.			
Perfil deseable del profesor que lo conduce o lo coordina			
Grado académico: Licenciatura. De preferencia con estudios de Maestría o Doctorado.		Área de formación: Ingeniería de Software o afines. Se recomienda un nivel intermedio-alto del idioma inglés para impartir esta materia.	
Experiencia docente: 1 año		Experiencia profesional en el campo: 1 año	

Elaboró: Mery Helen Pesantes Espinoza	Fecha: 9 de octubre de 2024
---------------------------------------	-----------------------------