



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERÍA MOCHIS LICENCIATURA EN INGENIERÍA DE SOFTWARE



PROGRAMA DE ESTUDIO

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE O MÓDULO:	INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE		
Clave:			
Ubicación:	Semestre I	Área: Básico disciplinar	
Horas y créditos:	Teóricas: 60	Prácticas: 20	Estudio Independiente: 80
	Total de horas: 160		Créditos: 10
Competencia(s) del perfil de egreso al que aporta:	<p>CE1. Comprende conocimientos, normas, experiencias y motivaciones que hacen posible la buena integración de las unidades de informática y su personal en las organizaciones y en la sociedad en general.</p> <p>CE2. Desarrolla habilidades de abstracción y la expresión de formalismos, además de proporcionar conocimientos específicos fundamentales para la informática y la computación.</p> <p>CE6. Comprende los sistemas digitales y las computadoras, así como de los principios físicos que los sustentan, con el objetivo de analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.</p> <p>CE7. Conoce y comprende el funcionamiento básico de un sistema computacional mediante el análisis de cada uno de sus componentes, así como la revisión de los antecedentes históricos a partir de los cuales operan los sistemas modernos.</p> <p>CE19. Distingue los conceptos de accesibilidad y usabilidad que posibiliten un mejor diseño del software y este sea fácilmente accedido por personas con discapacidad dentro de entornos digitales siempre aplicando reglas de heurísticas.</p>		
Unidades de aprendizaje relacionadas:	Auditoría Informática, Ingeniería de software, Metodologías tradicionales de desarrollo de software, Metodologías ágiles de desarrollo de software, Diseño y experiencia de usuario, Administración de proyectos de software, Modelado de procesos.		
Responsable(s) de elaborar el programa:	Ing. Diana Cecilia Camacho Flores		Fecha: Agosto de 2023
Responsable(s) de actualizar el programa:			Fecha:



PROGRAMA DE ESTUDIO

2. PROPÓSITO	
Proporcionar al estudiante los conocimientos y elementos para entender la disciplina de la ingeniería de software por medio de la: Selección del modelo de ciclo de vida que se adapte a un caso de estudio planteado. Que pueda identificar los requerimientos y su representación en modelos de domino e interacción y generar lineamientos de diseño y codificación e identificar diversos tipos de pruebas y conocer el proceso de mantenimiento.	
3. SABERES	
Teóricos:	Conoce e identifica los conceptos y elementos relacionados con el desarrollo de software de calidad y comprende la importancia del uso de diversas técnicas de la ingeniería de software.
Prácticos:	Elabora mapas mentales, exposiciones, debates, análisis de lecturas, cuestionarios e intercambio de ideas, resúmenes, síntesis, ensayos.
Actitudinales:	<ul style="list-style-type: none">• Adecuada comunicación oral y escrita• Sentido crítico• Responsabilidad en entrega de actividades• Trabajo en equipo• Compromiso con el aprendizaje• Habilidades para buscar y analizar información de diversas fuentes
4. CONTENIDOS	
1. Introducción a la ingeniería de software. 1.1. Descripción del desarrollo profesional de software 1.2. Descripción de la ingeniería de software 1.3. Ética en la ingeniería de software 2. Procesos de software 2.1. Descripción de los modelos de procesos de software 2.2. Descripción de las actividades del proceso: especificación del software, diseño e implementación, validación y evolución. 2.3. Cómo hacer frente al cambio: prototipado, entrega incremental y mejora del proceso. 2.4. Desarrollo ágil de software: descripción de los métodos ágiles, técnicas ágiles de desarrollo, gestión ágil de proyectos y escalando los métodos ágiles. 3. Ingeniería de requisitos 3.1. Requerimientos funcionales y no funcionales. 3.2. Proceso de ingeniería de requisitos: obtención de requisitos, especificación de requisitos, validación de requisitos y cambio en los requisitos. 4. Diseño de software 4.1. Diseño de la arquitectura: decisiones, vistas, patrones y arquitecturas de aplicaciones. 4.2. Diseño orientado a objetos usando UML. 4.3. Patrones de diseño. 5. Construcción de software 5.1. Código robusto mediante manejo de excepciones 5.2. Prácticas de programación defensiva 5.3. Estrategias de integración e implementación de cambios. 6. Verificación, validación y evolución del software	



PROGRAMA DE ESTUDIO

- 6.1. Calidad del software.
- 6.2. Revisiones e inspecciones.
- 6.3. Descripción de pruebas durante el desarrollo, desarrollo guiado por las pruebas, pruebas de liberación y pruebas de usuario.
- 6.4. Evolución del software: Proceso de evolución y Mantenimiento de software.

7. Gestión de proyectos de software

- 7.1. Gestión de riesgos del proyecto: identificación, análisis, planificación y monitoreo.
- 7.2. Gestión de las personas: motivación, trabajo en equipo, selección, organización y comunicación.
- 7.3. Planificación y seguimiento del proyecto: precios del software, desarrollo basado en planes, cronograma del proyecto, planificación ágil y técnicas de estimación.

8. Herramientas y ambientes

- 8.1. Gestión de la configuración del software y control de versiones.
- 8.2. Gestión de liberaciones.
- 8.3. Herramientas de modelado de análisis de requisitos y diseño.
- 8.4. Herramientas para pruebas (dinámicas y estáticas).
- 8.5. Herramientas para armado de versiones.

9. Confiabilidad del software

- 9.1. Disponibilidad y confiabilidad.
- 9.2. Requisitos de confiabilidad.
- 9.3. Arquitecturas tolerantes a fallas.
- 9.4. Programación de software confiable.
- 9.5. Medición de la confiabilidad.

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del docente:

- Exposición y explicación de los temas
- Coordinar lluvia de ideas en relación a los temas
- Propiciar el trabajo en equipo
- Asignar actividades a los alumnos para retroalimentar los temas presentados en clase
- Realizar la evaluación de los temas en base a los criterios definidos
- Utilizar herramientas tecnológicas que faciliten a los alumnos la comprensión de los temas
- Presentar ejemplos relacionados con los temas
- Resolver dudas y asesorar a los alumnos sobre los temas de la asignatura

Actividades del estudiante:

- Participación activa
- Realizar las actividades asignadas
- Trabajo en equipo
- Consulta y análisis de diferentes fuentes para realizar las actividades
- Busca la aclaración de dudas al docente

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Criterios de desempeño	6.2 Portafolio de evidencias
● Participación	<ul style="list-style-type: none">● Portafolio de actividades



PROGRAMA DE ESTUDIO

• Entrega de actividades en tiempo y forma • Evaluación parcial	• Exposiciones • Evaluaciones parciales			
6.3. Calificación y acreditación:				
Parcial: Portafolio de Actividades 60% Evaluación 40%	Final: Promedio de las evaluaciones parciales			
7. RECURSOS DIDÁCTICOS				
Bibliografía digital, proyector, aula virtual UAS, correo electrónico, internet, videos explicativos y páginas web oficiales.				
8. FUENTES DE INFORMACIÓN				
<i>Bibliografía básica</i>				
Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Roger S. Pressman	Ingeniería del software, un enfoque práctico (7ma Edición)	McGrawHill	2010	
Ian Sommerville	Software Engineering (10th Edition)	Pearson	2015	
9. PERFIL DEL DOCENTE				
Experiencia en ingeniería de software, con grado en algún área de tecnologías de la información.				