

<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p style="text-align: center;"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</b></p> <p style="text-align: center;"><b>INGENIERIA DE SOFTWARE I</b></p>	<b>DES:</b>	<b>INGENIERÍA</b>
	<b>Programa Educativo</b>	Ingeniería en Sistemas Computacionales en Hardware
	<b>Tipo de materia (Obli/Opta):</b>	Obligatoria
	<b>Clave de la materia:</b>	518
	<b>Semestre:</b>	5
	<b>Área en plan de estudios (G, E):</b>	Ingeniería aplicada
	<b>Total de horas por semana:</b>	4
	<i>Teoría: Presencial o Virtual</i>	3
	<i>Laboratorio o Taller:</i>	1
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	0
	<b>Créditos Totales:</b>	4
	<b>Total de horas semestre (x 16 sem):</b>	64
	Fecha de actualización:	Enero 2023
	<i>Prerrequisito (s):</i>	Lenguajes de programación II, (415)
<i>Realizado por:</i>	Comité de Rediseño Curricular	

**DESCRIPCIÓN:**

La industria de software actual requiere una interacción profunda entre la fábrica de software y el cliente, de tal forma que los procesos, métodos y mejores prácticas probadas se pongan de manifiesto en los proyectos de escritura de software, fundamentales para desarrollar proyectos de software medibles, repetibles y principalmente de calidad.

<b>DOMINIOS</b> (Se toman de las competencias)	<b>OBJETOS DE ESTUDIO</b> (Contenidos necesarios para desarrollar cada uno de los dominios)	<b>RESULTADOS DE APRENDIZAJE</b> (Se plantean de los dominios y contenidos)	<b>METODOLOGÍA</b> (Estrategias, secuencias, recursos didácticos)	<b>EVIDENCIAS</b> (Productos tangibles que permiten valorar los resultados de aprendizaje)
<p><b>Específicas.</b></p> <p><b>Sistemas Informáticos y Computación.</b></p> <p><b>Descripción:</b></p> <p>Aplica el conocimiento, metodologías, procesos y técnicas, para el análisis, diseño, modelado y desarrollo de sistemas informáticos y de cómputo.</p> <p><b>Dominio:</b></p> <p>Aplica los principios de modelado de sistemas de cómputo para su análisis y desarrollo.</p>	<p><b>UNIDAD I. CONCEPTOS FUNDAMENTALES DE INGENIERIA DE SOFTWARE.</b></p> <p>1.1 El desarrollo profesional del software 1.2 Origen histórico de la ingeniería de software 1.3 La crisis del software 1.4 Necesidad de la ingeniería de software</p>	<p>Valora las tareas del desarrollo profesional del software en el mercado actual.</p> <p>Analiza los aspectos que dieron origen a la ingeniería de software como solución a los problemas de entonces y actuales.</p>	<ul style="list-style-type: none"> <li>• Lecturas</li> <li>• Auto aprendizaje (búsqueda y análisis de información)</li> <li>• Trabajo colaborativo</li> <li>• Grupos de discusión</li> <li>• Aprendizaje interactivo (exposición del profesor)</li> <li>• Aprendizaje basado en prácticas de laboratorio (ABPL)</li> <li>• Recursos tecnológicos</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas de Investigación</li> <li>• Exposiciones</li> <li>• Prácticas de Laboratorio y / o actividad integradora.</li> <li>• Examen</li> </ul>

<p><b>Profesionales.</b></p> <p><b>Proyectos de Ingeniería.</b></p> <p><b>Descripción:</b></p> <p>Utiliza los conocimientos necesarios para la planeación, análisis, diseño y desarrollo de proyectos de ingeniería, utilizando las tecnologías y los principios de la administración para la optimización de los recursos, considerando su impacto ambiental.</p>	<p><b>UNIDAD II. INGENIERIA DE REQUERIMIENTOS.</b></p> <p>2.1 Requerimientos funcionales y no funcionales</p> <p>2.2 Obtención y análisis de requerimientos</p> <p>2.3 Validación de requerimientos</p> <p>2.4 Documento de requerimientos de software</p>	<p>Identifica los tipos de requerimientos de un producto de software en un ambiente de desarrollo comercial.</p> <p>Entiende las técnicas actuales para la recolección, validación e integración de los requerimientos de un proyecto de software.</p>	<ul style="list-style-type: none"> <li>• Lecturas</li> <li>• Auto aprendizaje (búsqueda y análisis de información)</li> <li>• Trabajo colaborativo</li> <li>• Grupos de discusión</li> <li>• Aprendizaje interactivo (exposición del profesor)</li> <li>• Aprendizaje basado en prácticas de laboratorio (ABPL)</li> <li>• Recursos tecnológicos</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas de Investigación</li> <li>• Exposiciones</li> <li>• Prácticas de Laboratorio y / o actividad integradora.</li> <li>• Examen.</li> </ul>
<p><b>Dominio:</b></p> <p>Define el perfil del personal que estará a Cargo del proyecto.</p>	<p><b>UNIDAD III. DISEÑO DE SOFTWARE.</b></p> <p>3.1 Modelado de sistemas:</p> <p>3.1.1 Modelos de interacción.</p> <p>3.1.2 Modelos estructurales.</p> <p>3.1.3 Modelos de comportamiento.</p> <p>3.2 Diseño arquitectónico:</p> <p>3.2.1 Estilos y patrones arquitectónicos.</p> <p>3.3 Diseño orientado a objetos:</p> <p>3.3.1 Diseño orientado a objetos con UML.</p> <p>3.3.2 Patrones de diseño.</p> <p>3.4 Interacción hombre – máquina:</p> <p>3.4.1 Prototipos.</p> <p>3.4.2 Diseño de interfaces (usabilidad).</p>	<p>Entiende los tipos de modelos existentes para la construcción de prototipos de software en la etapa de diseño del mismo.</p> <p>Comprende los distintos tipos de patrones arquitectónicos y su aplicación para determinadas soluciones de problemas en la industria del software.</p> <p>Diseña arquitecturas de software por medio del lenguaje de modelado unificado (UML) como instrumento de diseño. Diseña interfaces de usuario con un alto nivel de usabilidad para los usuarios finales del sistema.</p>	<ul style="list-style-type: none"> <li>• Lecturas</li> <li>• Auto aprendizaje (búsqueda y análisis de información)</li> <li>• Trabajo colaborativo.</li> <li>• Grupos de discusión.</li> <li>• Aprendizaje interactivo (exposición del profesor).</li> <li>• Aprendizaje basado en prácticas de laboratorio (ABPL).</li> <li>• Recursos tecnológicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas de Investigación</li> <li>• Exposiciones</li> <li>• Prácticas de Laboratorio y / o actividad integradora.</li> <li>• Examen</li> </ul>

	<p><b>UNIDAD IV. CONSTRUCCIÓN DEL SOFTWARE.</b></p> <p>4.1 Modelos de procesos de software. 4.1.1 RUP. 4.2 Métodos ágiles de desarrollo de software. 4.2.1 Manifiesto ágil.</p>	<p>Conoce los distintos procesos de desarrollo de software y su utilización en la industria.</p> <p>Entiende el manifiesto ágil y los diferentes métodos que se desprenden del como solución a la complejidad y demanda actual de software.</p>	<ul style="list-style-type: none"> <li>• Lecturas</li> <li>• Auto aprendizaje (búsqueda y análisis de información).</li> <li>• Trabajo colaborativo.</li> <li>• Grupos de discusión.</li> <li>• Aprendizaje interactivo (exposición del profesor).</li> <li>• Aprendizaje basado en prácticas de laboratorio (ABPL).</li> <li>• Recursos tecnológicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas de Investigación.</li> <li>• Exposiciones.</li> <li>• Prácticas de Laboratorio y / o actividad integradora.</li> <li>• Examen.</li> </ul>
	<p><b>UNIDAD V. PRUEBAS DE SOFTWARE.</b></p> <p>5.1 Tipos de pruebas de software: 5.1.1 Pruebas de caja negra. 5.1.2 Pruebas de caja blanca. 5.2 Jerarquía de pruebas: 5.2.1 Pruebas unitarias. 5.2.2 Pruebas de integración. 5.2.3 Pruebas de sistema. 5.2.4 Pruebas de regresión. 5.2.5 Otras pruebas. 5.3 Verificación y validación.</p>	<p>Comprende los distintos tipos de pruebas de software según el aspecto que se desea probar del software como instrumento de aseguramiento de la calidad.</p> <p>Entiende los distintos tipos de pruebas de software según el nivel de integración del sistema como instrumento de aseguramiento de la calidad.</p> <p>Verifica y valida los requerimientos iniciales del sistema contra su implementación en el producto final como instrumento de aseguramiento de la calidad.</p>	<ul style="list-style-type: none"> <li>• Lecturas.</li> <li>• Auto aprendizaje (búsqueda y análisis de información).</li> <li>• Trabajo colaborativo.</li> <li>• Grupos de discusión.</li> <li>• Aprendizaje interactivo (exposición del profesor).</li> <li>• Aprendizaje basado en prácticas de laboratorio (ABPL).</li> <li>• Recursos tecnológicos.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas de Investigación.</li> <li>• Exposiciones.</li> <li>• Prácticas de Laboratorio y / o actividad integradora y/o Proyecto Final.</li> <li>• Examen.</li> </ul>

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos)
<ol style="list-style-type: none"> <li>Ingeniería del software. Ian Sommerville. Novena edición. Pearson – Addison Wesley.</li> <li>Ingeniería del software. Un enfoque práctico. Roger S. Pressman. Sexta edición. McGraw Hill.</li> </ol>	<p><b>INSTRUMENTOS:</b></p> <ul style="list-style-type: none"> <li><b>Actividades:</b> <ul style="list-style-type: none"> <li>Tareas de Investigación</li> <li>Exposiciones</li> <li>Prácticas de Laboratorio y/o actividad integradora y/o Proyecto.</li> </ul> </li> <li><i>Examen</i></li> </ul> <p><b>CRITERIOS DE DESEMPEÑO:</b></p> <ul style="list-style-type: none"> <li>Las actividades deberán estar completas y entregadas en tiempo y forma.</li> </ul> <p><b>Exámenes:</b></p> <p>Se realizan 3 exámenes durante el semestre y las fechas se establecerán por la secretaría académica.</p> <p>Se toma en cuenta para integrar <b>calificaciones parciales:</b></p> <p>Exámenes 60% Actividades 40%</p> <p><b>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</b></p> <p><b>Nota:</b> para acreditar el curso se deberá tener calificación aprobatoria tanto en los exámenes y actividades.</p>

### CRONOGRAMA

Objetos de estudio	Semanas																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
I. Conceptos fundamentales de ingeniería de software																	
II. Ingeniería de requerimientos																	
III. Diseño de software																	
IV. Construcción del software																	
V. Pruebas de software																	