


<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></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 II</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>	618
	<b>Semestre:</b>	6
	<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>	Ingeniería de software I, (618)	
<i>Realizado por:</i>	Comité de Rediseño Curricular	

**DESCRIPCIÓN:**

La ingeniería de software es una disciplina altamente cambiante por lo que el estudio de las diferentes herramientas ideológicas y tecnológicas que ayudan a mantener la calidad del software actual es necesario en el contexto actual complejo y demandante de desarrollo de software.

<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> Aplica la ingeniería electrónica y de sistemas computacionales para dar soporte tecnológico a otros campos y resolver problemas en distintos sectores y áreas del conocimiento.</p> <p><b>Dominio:</b> Aplica los principios de modelado de sistemas de cómputo para su análisis y desarrollo.</p>	<p><b>UNIDAD I.</b> <b>ADMINISTRACION DE PROYECTOS DE SOFTWARE</b></p> <p>1.1. Planificación del proyecto. 1.2. Gestión de riesgos 1.2.1. Identificación de Riesgos 1.2.2. Análisis y Plan de Riesgos 1.2.3. Monitoreo de Riesgos 1.3. Gestión de recursos del proyecto 1.4. Calendario del proyecto 1.5. Administración de la calidad 1.5.1. Estándares del software 1.5.2. Revisiones e inspecciones 1.6. Métricas de software</p>	<p>Enumera las etapas de planificación de proyectos de software para la industria actual.</p> <p>Determina los riesgos en las distintas etapas del desarrollo de software y selecciona las técnicas para su corrección.</p> <p>Describe los conceptos básicos de calidad de software y enlista las diferentes herramientas como métricas y estándares para su administración durante la vida del proyecto.</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> 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>Dominio:</b> Aplica los principios de modelado de sistemas de cómputo para su análisis y desarrollo.</p>	<p><b>UNIDAD II.</b> <b>MADUREZ Y MEJORA DE PROCESOS</b></p> <p>2.1. CMMI 2.1.1. Niveles de CMMI 2.2. MOPROSOFT 2.3. El proceso personal de software (Personal Process Software - PSP) 2.3.1. Proceso Actual y medición de proceso. 2.3.2. Estimación y planeación. 2.3.3. Introducción a diseño y administración de calidad. 2.4. El proceso en equipo de software (Team Process Software -TSP) 2.4.1. Proceso general de TSP</p>	<p>Identifica los distintos modelos de evaluación y administración de la madurez de las fábricas de software como mecanismo de certificación de calidad en la industria dl software actual.</p> <p>Entiende los conceptos básicos estadísticos de los procesos de desarrollo de software personal y por equipos como certificación de calidad de los recursos humanos asociados a proyectos de escritura de software.</p> <p>Aplica los conceptos básicos del Proceso Personal de Software en la elaboración de proyectos de clase.</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 III.</b> <b>ADMINISTRACION DE LA CONFIGURACION</b></p> <p>3.1. Gestión del cambio 3.1.1. Ley de Lehman 3.2. Procesos de evolución 3.2.1. Sistemas legacy 3.2.2. Mantenimiento de Software 3.2.3. Reingeniería de sistemas 3.2.4. Refactoring 3.3. Manejo de versiones</p>	<p>Entiende los mecanismos que detonan el cambio en los productos de software en la industria actual.</p> <p>Conoce las técnicas de administración de los cambios de software heredado y actual como mecanismo de administración de la vigencia del producto de software en la industria actual.</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.</b> <b>TOPICOS SELECTOS DE INGENIERIA DE SOFTWARE</b></p> <p>4.1. Reutilización del software 4.1.1. Líneas de productos 4.1.2. COTS 4.2. Ingeniería de software basada en componentes 4.2.1. Modelos de Componentes 4.3. Ingeniería de software distribuida 4.3.1. Sistemas distribuidos 4.3.2. Patrones arquitectónicos para sistemas distribuidos 4.4. Ingeniería de Software orientada a servicio. 4.4.1. Arquitecturas orientadas a servicios 4.5. Ingeniería de Software en tiempo real 4.5.1. Software embebido 4.6. Ingeniería de software orientada a aspectos</p>	<p>Conoce técnicas modernas de administración y desarrollo de productos de software para el entorno cambiante de la industria de software actual.</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>• Proyecto Final</li> <li>• Examen</li> </ul>
--	---	---	---	--

<b>FUENTES DE INFORMACIÓN</b> (Bibliografía, direcciones electrónicas)	<b>EVALUACIÓN DE LOS APRENDIZAJES</b> (Criterios, ponderación e instrumentos)
<ol style="list-style-type: none"> <li>1. Ingeniería del software. Ian Sommerville. Novena edición. Pearson – Addison Wesley.</li> <li>2. Ingeniería del software. Un enfoque práctico. Roger S. Pressman. Sexta edición. McGraw Hill.</li> <li>3. PSP A self-Improvement Process for Software Engineers. Watts S. Humphrey. 2012. Addison Wesley</li> </ol>	<p><b>INSTRUMENTOS:</b></p> <p><b>Actividades:</b></p> <ul style="list-style-type: none"> <li>• Tareas de Investigación</li> <li>• Exposiciones</li> <li>• Prácticas de Laboratorio y/o actividad integradora.</li> <li>• Proyecto</li> <li>• Examen</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> <li>• <b>Exámenes:</b> Se realizan 3 exámenes durante el semestre y las fechas se establecerán por la secretaría académica. Se toma en cuenta para integrar <b>calificaciones parciales:</b> Exámenes 60% Actividades 40%</li> </ul> <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. ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE	■	■	■													
II. MADUREZ Y MEJORA DE PROCESOS				■	■	■	■	■	■	■						
III. ADMINISTRACIÓN DE LA CONFIGURACIÓN											■	■				
IV. TÓPICOS SELECTOS DE INGENIERÍA DE SOFTWARE													■	■	■	■