

<p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p style="text-align: center;">UNIDAD ACADÉMICA</p> <p style="text-align: center;">PROGRAMA DE LA UNIDAD DE APRENDIZAJE:</p> <p style="text-align: center;">INGENIERÍA DE SOFTWARE AVANZADA</p>	DES:	INGENIERÍA
	Programa Educativo	Maestría en Ingeniería en Computación
	Tipo de materia (Obli/Opta):	Optativa
	Clave de la materia:	MICOP219
	Semestre:	2
	Área en plan de estudios (G, E):	
	Total de horas por semana:	4
	<i>Teoría: Presencial o Virtual</i>	2
	<i>Laboratorio o Taller:</i>	0
	<i>Prácticas:</i>	2
	<i>Trabajo extra-clase:</i>	6
	Créditos Totales:	6
	Total de horas semestre (x 16 sem):	64
	Fecha de actualización:	Julio 2018
	<i>Prerrequisito (s):</i>	Ninguno
<i>Realizado por:</i>	Comité de Rediseño Curricular	

DESCRIPCIÓN:

Esta unidad de aprendizaje contribuye a que el estudiante conozca y aplique las mejores practicas de Ingeniería de Software; siguiendo el Proceso Personal de Software: controlando, administrando y mejorando la manera en la que desarrollan software.

COMPETENCIAS POR DESARROLLAR:

Genéricas.

Gestión de proyectos

Coordina y administra de forma responsable, proyectos que atiendan criterios de sustentabilidad y que contribuyan a mejorar la calidad de vida.

Gestión del conocimiento

Demuestra conocimientos y habilidades para la búsqueda, análisis crítico, síntesis y procesamiento de información para su transformación en conocimiento con actitud ética.

Específicas.

Aplicación de Ciencias de la Computación.

Aplica las ciencias computacionales para resolver problemas en los sectores industrial, gubernamental y académico bajo esquemas de colaboración ética y multidisciplinaria.

DOMINIOS (Se toman de las competencias)	OBJETOS DE ESTUDIO (Contenidos necesarios para desarrollar cada uno de los dominios)	RESULTADOS DE APRENDIZAJE (Se plantean de los dominios y contenidos)	METODOLOGÍA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS (Productos tangibles que permiten valorar los resultados de aprendizaje)
Gestiona, almacena, organiza y categoriza la información de	1. Introducción 1.1. Conceptos Fundamentales de Ingeniería de Software.	Enumera los diferentes conceptos básicos de la ingeniería de software y entiende	Se establecerán lecturas en las cuáles el alumno deberá escribir ensayos reflexivos y	Ensayos

manera que se traduzca en conocimiento.	<ul style="list-style-type: none"> 1.2. Origen histórico de la ingeniería de software. 1.3. La crisis del software 1.4. Necesidad de la ingeniería de software. 1.5. Mejorando la práctica de desarrollo de software 	de la importancia de esta.	críticos.	
Propone áreas de oportunidad sobre los procesos y logros del proyecto que contribuyan a implementar mejores prácticas en futuros proyectos.	2. Proceso Personal de Software <ul style="list-style-type: none"> 2.1. Estrategia del Proceso Personal. 2.2. Base del Proceso Personal 2.3. PSP0 	Define y usa un proceso personal básico, mide tiempo de desarrollo en cada fase del proceso y los defectos que inyecta, encuentra y remueve.	Conoce los fundamentos de la base del proceso personal, así como la estrategia para poder desarrollar su primer programa.	Programa 1, utilizando PSP 0.
Propone áreas de oportunidad sobre los procesos y logros del proyecto que contribuyan a implementar mejores prácticas en futuros proyectos.	3. Medición, Planeación y Estimación de Software <ul style="list-style-type: none"> 3.1. Medición del tamaño del software 3.2. Estándar de codificación y conteo. 3.3. Planeación del proceso 3.4. Estimación del software 3.5. Método de estimación PROBE 3.6. PSP1 	Mide el tamaño de sus programas de manera automática, entiende requerimientos, produce un diseño conceptual para poder hacer una estimación de tamaño y hacer una planeación que represente precisamente lo que se va a hacer.	Aprende como medir y contar el código que produce, para poder hacer una planeación estimando el tamaño y el tiempo.	Estándar de codificación y conteo. Programa 2, utilizando PSP 1.
Propone áreas de oportunidad sobre los procesos y logros del proyecto que contribuyan a implementar mejores prácticas en futuros proyectos.	4. Administración y Calidad de Software <ul style="list-style-type: none"> 4.1. Planeación y Calendario 4.2. Calidad de Software 4.3. Revisiones de diseño y código 4.4. PSP2 	Utiliza datos históricos para hacer estimaciones de tamaño y tiempo. Aprende técnicas para encontrar y arreglar defectos en etapas tempranas de desarrollo	Entiende la importancia de la calidad de software y técnicas para mejorar la calidad de sus propios desarrollos.	Programa 3 y 4 utilizando PSP2.
Propone áreas de oportunidad sobre los procesos y logros del proyecto que contribuyan a	5. Diseño de software <ul style="list-style-type: none"> 5.1. Diseño de software (Plantillas de diseño) 5.2. Verificación del diseño (Tablas de 	Define los criterios de salida de la fase de diseño, basado en un estándar de define el contenido de un diseño completo.	Aprende a utilizar herramientas para el diseño de software.	Programa 5, 6 y 7 utilizando PSP 2.1.

implementar mejores prácticas en futuros proyectos.	ejecución, tablas de rastreo y máquinas de estado) 5.3. PSP 2.1			
Construye prototipos innovadores utilizando herramientas de software y hardware adecuadas a la solución de problemas en cualquier ámbito de desempeño.	6. Aplicación del Proceso Personal 6.1. Usando el proceso personal de software 6.2. Propuesta de Proyecto de Software	Aplica las mejores prácticas del Proceso personal de Software a un proyecto externo.	Aprende a utilizar el proceso personal de software en un proyecto que el alumno propone. Se le da la estructura para la propuesta del proyecto.	Propuesta de proyecto y entrega de proyecto final.
Propone áreas de oportunidad sobre los procesos y logros del proyecto que contribuyan a implementar mejores prácticas en futuros proyectos.	7. Análisis de Datos Personales	Realiza un análisis de sus propios datos para encontrar su propio rendimiento en la planeación, calidad y proceso de desarrollo.	Analiza sus datos	Reporte de análisis de rendimiento

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos)
Roger S. Pressman. "Ingeniería de Software - Un enfoque practico" 7ma. edición, McGraw-Hill Interamericana. Watts S. Humphrey. "PSP, A Self-Improvement Process for Software Engineers", The SEI Series in software Engineering, Software Engineering Institute, Carnegie Mellon.	Se toma en cuenta para integrar la calificación final: Examen teórico: 30% Programas, tareas y trabajos: 40% Proyecto: 30% Nota: La calificación mínima aprobatoria es 80

