


<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;">PROGRAMA ANALITICO DE LA UNIDAD DE APRENDIZAJE:</p> <p style="text-align: center;">INGENIERIA DE SOFTWARE I</p>	DES:	Ingeniería
	Programa(s) Educativo(s):	Ingeniería en Ciencias de la Computación
	Tipo de materia:	Obligatoria
	Clave de la materia:	IA778
	Semestre:	7°
	Area en plan de estudios:	Ingeniería Aplicada
	Créditos	3
	Total de horas por semana:	3
	<i>Teoría:</i>	3
	<i>Práctica</i>	0
	<i>Taller:</i>	0
	<i>Laboratorio:</i>	0
	<i>Prácticas complementarias:</i>	0
	<i>Trabajo extra clase:</i>	0
	Total de horas semestre:	48
	Fecha de actualización:	Febrero 2023
Materia requisito:		

PROPÓSITO DEL CURSO

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

COMPETENCIAS (Tipo Y Nombre de la competencias que nutre la materia y a las que contribuye)	DOMINIOS COGNITIVOS (Objetos de estudio, temas y subtemas)	RESULTADOS DE APRENDIZAJE. (Por objeto de estudio).
<p>COMPETENCIAS BÁSICAS:</p> <p>SOLUCION DE PROBLEMAS</p> <ul style="list-style-type: none"> • Aplica las diferentes técnicas de observación para la solución de problemas. • Analiza las diferentes componentes de un problema y sus interrelaciones. • Emplea diferentes métodos para establecer alternativas de solución de problemas. 	<p>I. CONCEPTOS FUNDAMENTALES DE INGENIERIA DE SOFTWARE</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>

- Aplica el enfoque sistémico en diversos contextos.
- Adapta críticamente sus propios conceptos y comportamientos a normas, ambientes y situaciones cambiantes.
- Asume una actitud responsable por el estudio independiente.
- Distingue los diversos tipos de sistemas
- Aplica la tecnología a la solución de problemáticas

TRABAJO EN EQUIPO Y LIDERAZGO

- Participa en la elaboración y ejecución de planes y proyectos mediante el trabajo en equipo.
- Interactúa en grupos multidisciplinarios.
- Desarrolla y estimula una cultura de trabajo de equipo hacia el logro de una meta común.
- Demuestra respeto, tolerancia, responsabilidad y apertura a la confrontación y pluralidad en el trabajo grupal.
- Respeto, tolera y es flexible ante el pensamiento divergente para lograr acuerdos por consenso.
- Cumple y hace cumplir las normas y leyes establecidas en un contexto social.

TRABAJO EN EQUIPO Y LIDERAZGO

- Genera y ejecuta proyectos productivos con responsabilidad social y ética.
- Adapta el conocimiento y habilidades al desarrollo de proyectos.

- Demuestra capacidad de generación de empleo y autoempleo.
- Aprovecha óptimamente los recursos existentes.
- Utiliza los principios de administración estratégica en el desarrollo de proyectos.
- Aplica métodos para promover, ejecutar y valorar el impacto de un proyecto.
- Genera y adecua nuevas tecnologías en su área.
- Revalora tecnologías tradicionales, alternativas y de punta para seleccionar la apropiada en la solución de problemas en su área, considerando el impacto que dichas tecnologías tendrán sobre el ambiente.
- Selecciona de las tecnologías a su alcance, las apropiadas para su desempeño.

COMUNICACIÓN

- Desarrolla su capacidad de comunicación escrita en forma efectiva.
- Desarrolla habilidades de lectura e interpretación de textos.
- Demuestra su habilidad de síntesis en el lenguaje verbal y escrito.
- Demuestra dominio básico en el manejo de recursos documentales y electrónicos que apoyan a la comunicación y búsqueda de información (Internet, correo electrónico, audio, conferencias, correo de voz, entre otros).
- Recopila, analiza y aplica información de diversas fuentes.
- Emplea la estadística en la interpretación de

<p>resultados y construcción de conocimiento.</p> <ul style="list-style-type: none"> • Maneja y aplica paquetes computacionales para desarrollar documentos, presentaciones y bases de información. • Desarrolla escritos a partir del proceso de investigación. • Utiliza creativamente la información para atender problemas o tareas específicas. • Genera indicadores y criterios de desempeño a partir de información relevante. 		
<p>PROFESIONALES:</p> <p>PROYECTOS DE INGENIERIA</p> <ul style="list-style-type: none"> • Analiza la factibilidad o viabilidad de un proyecto de ingeniería para justificar su desarrollo. • Administra las etapas de un proceso a fin de integrarlas para lograr los mayores beneficios sociales, profesionales y personales. • Planea un proceso desde su concepción, puesta en marcha, operatividad y control para alcanzar los objetivos que en el mismo proyecto se plantean. • 	<p>II. INGENIERIA DE REQUERIMIENTOS</p> <p>2.1 Requerimientos funcionales y no funcionales 2.2 Obtención y análisis de requerimientos 2.3 Validación de requerimientos 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"> • Integra eficientemente a su proceso los recursos humanos, materiales y financieros disponibles para alcanzar los objetivos proyectados. • Maneja riesgos en relación a recursos humanos, técnicos y financieros para su mitigación. • Organiza, desarrolla y administra proyectos 	<p>III. DISEÑO DE SOFTWARE</p> <p>3.1 Modelado de sistemas 3.1.1 Modelos de interacción 3.1.2 Modelos estructurales 3.1.3 Modelos de comportamiento 3.2 Diseño arquitectónico 3.2.1 Estilos y patrones arquitectónicos 3.3 Diseño orientado a objetos 3.3.1 Diseño orientado a objetos con</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</p>

<p>específicos, incluida la presupuestación, supervisión y evaluación para la ejecución exitosa del proyecto.</p> <p>ESPECIFICAS:</p> <p>FUNDAMENTOS DE CIENCIAS DE LA COMPUTACIÓN</p> <ul style="list-style-type: none"> • Implementa y diseña interfaces humano-computadora. 	<p>UML</p> <p>3.3.2 Patrones de diseño</p> <p>3.4 Interacción hombre - maquina</p> <p>3.4.1 Prototipos</p> <p>3.4.2 Diseño de interfaces (usabilidad)</p>	<p>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.</p> <p>Diseña interfaces de usuario con un alto nivel de usabilidad para los usuarios finales del sistema.</p>
<p>MODELADO Y ANALISIS DE SISTEMAS COMPUTACIONALES</p> <ul style="list-style-type: none"> • Aplica los principios de modelado de sistemas de cómputo para su análisis y desarrollo. • Interpreta los modelos matemáticos y lenguajes de especificación para la creación de prototipos. • Interpreta las representaciones de modelos (sintáctica y semántica) para la implementación de sistemas de cómputo. • Aplica el modelado del sistema para predecir y validar sus especificaciones de comportamiento. • Interpreta la definición de necesidades, así como suproceso, sus niveles de abstracción y características para el desarrollo de modelos fiables. • Aplica y elabora técnicas para validar los modelos. 	<p>IV. CONSTRUCCION DEL SOFTWARE</p> <p>4.1 Modelos de procesos de software</p> <p>4.1.1 RUP</p> <p>4.2 Métodos ágiles de desarrollo de software</p> <p>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 de él como solución a la complejidad y demanda actual de software.</p>
<p>DISEÑO Y DESARROLLO DE SISTEMAS COMPUTACIONALES</p>	<p>V. PRUEBAS DE SOFTWARE</p> <p>5.1 Tipos de pruebas de software</p> <p>5.1.1 Pruebas de caja negra</p> <p>5.1.2 Pruebas de caja blanca</p>	<p>Comprende los distintos tipos de pruebas de software según el aspecto que se desea probar del software como instrumento</p>

<ul style="list-style-type: none"> • Utiliza las herramientas para el diseño de sistemas de cómputo. • Analiza el desempeño del sistema computacional para su validación y optimización. • Aplica los fundamentos del diseño de la arquitectura de sistemas para el desarrollo de soluciones computacionales adecuadas. • Analiza, desarrolla, aplica y valida modelos de diseño para la implementación de sistemas computacionales. • Propone opciones para mejora del desempeño del sistema de cómputo. 	<p>5.2 Jerarquía de pruebas</p> <p>5.2.1 Pruebas unitarias</p> <p>5.2.2 Pruebas de integración</p> <p>5.2.3 Pruebas de sistema</p> <p>5.2.4 Pruebas de regresión</p> <p>5.2.5 Otras pruebas</p> <p>5.3 Verificación y validación</p>	<p>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>
--	--	---

OBJETO DE ESTUDIO	METODOLOGIA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS DE APRENDIZAJE.
<p>I. CONCEPTOS FUNDAMENTALES DE INGENIERIA DE SOFTWARE</p> <p>II. INGENIERIA DE REQUERIMIENTOS</p> <p>III. DISEÑO DE SOFTWARE</p> <p>IV. CONSTRUCCION DEL SOFTWARE</p> <p>V. PRUEBAS DE SOFTWARE</p>	<p>Aprendizaje interactivo (exposición del profesor)</p> <p>Grupo de discusión.</p> <p>Auto aprendizaje (búsqueda y análisis de información)</p> <p>Inductivo</p> <ul style="list-style-type: none"> • Observación • Comparación <p>Deductivo</p> <ul style="list-style-type: none"> • Aplicación • Comprobación • Demostración <p>Sintético</p> <ul style="list-style-type: none"> • Recapitulación • Definición • Resumen • Esquemas • Modelos matemáticos • Conclusión. <p>Material de Apoyo didáctico:</p> <p>Recursos</p> <ul style="list-style-type: none"> • Manual de Instrucción 	<ul style="list-style-type: none"> • Tareas y ejercicios. • Escritura de programas de cómputo. • Exámenes escritos.

	<ul style="list-style-type: none"> • Materiales gráficos: artículos, libros, diccionarios, etc. • Cañón • Pintarrones • Equipo de computo 	
--	---	--

FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos)
<ol style="list-style-type: none"> 1. Ingeniería del software. Ian Sommerville. Novena edición. Pearson – Addison Wesley. 2. Ingeniería del software. Un enfoque practico. Roger S. Pressman. Sexta edición. McGraw Hill. 	<p>INSTRUMENTOS:</p> <ul style="list-style-type: none"> • Exámenes • Solución de ejercicios (aplicación de conocimientos • Lista de cotejo (Respeto y participación al trabajo dentro del salón de clase, interés por la asignatura <p>CRITERIOS DE DESEMPEÑO:</p> <ul style="list-style-type: none"> • Los programas desarrollados deberán estar completos. • Las tareas se deberán entregar en tiempo y forma <ul style="list-style-type: none"> • Exámenes escritos: Se realizan 3 exámenes escritos durante el semestre y las fechas se establecen por la secretaría académica <p>Se toma en cuenta para integrar calificaciones parciales: Exámenes 60% Tareas y practicas 40%</p> <p>Fecha de exámenes parciales: 1º. Parcial: 2º. Parcial: 3º. Parcial:</p> <p>LAS ACTIVIDADES NO REALIZADAS EN TIEMPO Y FORMA SE CALIFICAN CON CERO.</p> <p>Nota: para acreditar el curso se deberá tener calificación aprobatoria tanto en la teoría como en las prácticas.</p>

Cronograma del Avance Programático

Unidades de aprendizaje	S e m a n a s																
	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																	