


<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 ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</p> <p style="text-align: center;">LENGUAJES DE PROGRAMACIÓN IV</p>	DES:	INGENIERÍA
	Programa Educativo	Ingeniería en Sistemas Computacionales en Hardware
	Tipo de materia (Obli/Opta):	Obligatoria
	Clave de la materia:	615
	Semestre:	6
	Área en plan de estudios (G, E):	E
	Total de horas por semana:	5
	<i>Teoría: Presencial o Virtual</i>	3
	<i>Laboratorio o Taller:</i>	2
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	0
	Créditos Totales:	5
	Total de horas semestre (x 16 sem):	80
	<i>Fecha de actualización:</i>	Enero 2023
<i>Prerrequisito (s):</i>	Lenguajes de programación III, (515)	
<i>Realizado por:</i>	Comité de Rediseño Curricular	

DESCRIPCIÓN:

El curso proporciona conceptos fundamentales para promover la adquisición de habilidades para el análisis, diseño y modelado de sistemas de cómputo, utilizando los diferentes tipos de lenguaje de modelado unificado (UML), siendo capaz de escribir programas de software simples empleando los diferentes patrones de diseño para solucionar problemas específicos con programación orientada a objetos.

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)
<p>Específicas:</p> <p>Sistemas Informáticos y Computación.</p> <p>Descripción:</p> <p>Aplica el conocimiento, metodologías, procesos y técnicas, para el análisis, diseño, modelado y desarrollo de sistemas informáticos y de cómputo.</p> <p>Dominio:</p> <p>Diseña y aplica algoritmos,</p>	<p>UNIDAD I.</p> <p>ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS</p> <ol style="list-style-type: none"> 1.1. Análisis de los requerimientos 1.2. Sustantivos, adjetivos y verbos para diseñar clases y sus comportamientos 1.3. El alcance del sistema 1.4. Definición de conceptos del sistema 1.5. Identificación de clases potenciales 1.6. Identificación de atributos potenciales 1.7. Identificación de métodos potenciales 	<p>Define y discute los conceptos fundamentales de análisis y diseño orientado a objetos.</p> <p>Diseña clases a partir de requerimientos dados en el ámbito del modelo orientado a objetos.</p>	<ul style="list-style-type: none"> • Lecturas • Auto aprendizaje (búsqueda y análisis de información) • Aprendizaje interactivo (exposición del profesor) • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos 	<ul style="list-style-type: none"> • Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen

<p>estructuras y representación de datos para soluciones computacionales.</p> <p>Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas.</p>	<p>1.8. Identificación de características comunes (herencia)</p> <p>1.9. Elaboración de clases</p> <p>1.10. Asociación de clases, agregación y composición</p>			
<p>Aplica los tópicos de paradigmas de programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas.</p>	<p>UNIDAD II. EL LENGUAJE DE MODELADO UNIFICADO (UML)</p> <p>2.1. Introducción a UML</p> <p>2.2. Diseño con UML</p> <p>2.3. Tipos de diagramas</p> <p>2.3.1. Diagramas estructurales</p> <p>2.3.1.1. Diagramas de clases</p> <p>2.3.1.2. Diagramas de componentes</p> <p>2.3.1.3. Diagramas de despliegue</p> <p>2.3.1.4. Diagramas de objetos</p> <p>2.3.1.5. Diagramas de paquetes</p> <p>2.3.2. Diagramas de comportamiento</p> <p>2.3.2.1. Diagramas de Actividades</p> <p>2.3.2.2. Diagramas de Comunicación</p> <p>2.3.2.3. Diagramas de Secuencia</p> <p>2.3.2.4. Diagramas de Estado</p> <p>2.3.2.5. Diagramas de Casos de Uso</p> <p>2.4. Casos de estudio y ejemplos</p>	<p>Identifica y explica los diferentes tipos de diagramas del lenguaje de modelado unificado (UML).</p> <p>Diseña y modela sistemas de cómputo por medio del lenguaje de modelado unificado (UML) en el ámbito de la teoría orientada a objetos.</p>	<ul style="list-style-type: none"> • Lecturas • Auto aprendizaje (búsqueda y análisis de información) • Aprendizaje interactivo (exposición del profesor) • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos 	<ul style="list-style-type: none"> • Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen
	<p>UNIDAD III. PATRONES DE DISEÑO</p> <p>3.1. Patrones creacionales</p> <p>3.1.1. Concepto de patrón de diseño</p> <p>3.1.2. Fabrica abstracta (Abstract Factory)</p> <p>3.1.3. Constructor virtual (Builder)</p>	<p>Define el concepto de patrón de diseño para la solución de problemas conocidos en el ámbito de la programación orientada a objetos.</p> <p>Desarrolla programas simples empleando los</p>	<ul style="list-style-type: none"> • Lecturas • Auto aprendizaje (búsqueda y análisis de información) • Aprendizaje interactivo (exposición del profesor) 	<ul style="list-style-type: none"> • Tareas de Investigación • Exposiciones • Prácticas de Laboratorio • Actividad integradora. • Examen

	<p>3.1.4. Método de fabricación (Factory method)</p> <p>3.1.5. Prototipo (Prototype)</p> <p>3.1.6. Instancia única (Singleton)</p> <p>3.2. Patrones estructurales</p> <p>3.2.1. Adaptador o envoltorio (Adapter o wrapper)</p> <p>3.2.2. Puente (Bridge)</p> <p>3.2.3. Objeto compuesto (Composite)</p> <p>3.2.4. Decorador (Decorator)</p> <p>3.2.5. Fachada (Facade)</p> <p>3.2.6. Peso ligero (Flyweight)</p> <p>3.2.7. Proxy</p> <p>3.3. Patrones de comportamiento</p> <p>3.3.1. Cadena de responsabilidad (Chain of responsibility)</p> <p>3.3.2. Orden (Command)</p> <p>3.3.3. Interpretador (Interpreter)</p> <p>3.3.4. Iterador (Iterator)</p> <p>3.3.5. Mediador (Mediator)</p> <p>3.3.6. Recuerdo (Memento)</p> <p>3.3.7. Observador (Observer)</p> <p>3.3.8. Estado (State)</p> <p>3.3.9. Estrategia (Strategy)</p> <p>3.3.10. Método Plantilla (Template method)</p> <p>3.3.11. Visitante (Visitor)</p>	<p>diferentes patrones de diseño para solucionar problemas específicos con programación orientada a objetos.</p>	<ul style="list-style-type: none"> • Técnicas de enseñanza demostrativa • Aprendizaje basado en prácticas de laboratorio (ABPL) • Recursos tecnológicos 	
--	--	--	--	--

FUENTES DE INFORMACIÓN (Bibliografía, direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios, ponderación e instrumentos)
<ol style="list-style-type: none"> 1. Análisis y diseño orientado a objetos. Grady Booch. 2. UML y patrones – Introducción al análisis y diseño orientado a objetos. Larman. Pearson – Prentice Hall. 3. Design Patterns. Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison Wesley. 	<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)

