


<p>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p> <p>POGRAMA ANALITICO DE LA UNIDAD DE APRENDIZAJE:</p> <p>PROGRAMACION ORIENTADA A OBJETOS AVANZADA</p>	DES:	Ingeniería
	Programa(s) Educativo(s):	Ingeniería en Ciencias de la Computación
	Tipo de materia:	Obligatoria
	Clave de la materia:	CI575
	Semestre:	5°
	Area en plan de estudios:	Ciencias de la Ingeniería
	Créditos	4
	Total de horas por semana:	4
	<i>Teoría:</i>	2
	<i>Práctica</i>	2
	<i>Taller:</i>	0
	<i>Laboratorio:</i>	0
	<i>Prácticas complementarias:</i>	0
	<i>Trabajo extra clase:</i>	0
	Total de horas semestre:	80
Fecha de actualización:	Febrero 2023	
Materia requisito:	Programación orientada a objetos	

PROPÓSITO DEL CURSO

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.

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>El curso promueve las siguientes competencias:</p> <p>BÁSICAS:</p> <p>SOLUCIÓN DE PROBLEMAS</p> <ul style="list-style-type: none"> • Aplica las diferentes técnicas de observación para la solución de problemas. • Distingue los diversos tipos de sistemas • Aplica la tecnología a la solución de problemáticas • Aplica el enfoque sistémico en diversos 	<p>I. 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 1.8. Identificación de características comunes (herencia) 1.9. Elaboración de clases 1.10. Asociación de clases, agregación y composición 	<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>

<p>contextos</p> <ul style="list-style-type: none"> • Crea soluciones innovadoras y utiliza formas no convencionales en la solución de problemas 		
<p>PROFESIONALES:</p> <p>CIENCIAS FUNDAMENTALES DE LA INGENIERÍA</p> <ul style="list-style-type: none"> • Utiliza las matemáticas como herramientas para solución de problemas en ingeniería 	<p>II. EL LENGUAJE DE MODELADO UNIFICADO (UML)</p> <p>2.1 Introducción a UML 2.2 Diseño con UML 2.3. Tipos de diagramas 2.3.1 Diagramas estructurales 2.3.1.1 Diagramas de clases 2.3.1.2 Diagramas de componentes 2.3.1.3 Diagramas de despliegue 2.3.1.4 Diagramas de objetos 2.3.1.5 Diagramas de paquetes 2.3.2 Diagramas de comportamiento 2.3.2.1 Diagramas de Actividades 2.3.2.2 Diagramas de Comunicación 2.3.2.3 Diagramas de Secuencia 2.3.2.4 Diagramas de Estado 2.3.2.5 Diagramas de Casos de Uso 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 computo por medio del lenguaje de modelado unificado (UML) en el ambito de la teoría orientada a objetos.</p>
<p>ESPECIFICAS:</p> <p>FUNDAMENTOS DE CIENCIAS DE LA COMPUTACIÓN</p> <ul style="list-style-type: none"> • Diseña y aplica algoritmos, estructuras y representación de datos para soluciones computacionales • Distingue los fundamentos de un sistema computacional • Aplica las bases de los lenguajes de programación para generar aplicaciones óptimas • Aplica los tópicos de paradigmas de programación: estructurado, orientado a objetos. Lógico, funcional, entre otros para la creación de aplicaciones óptimas 	<p>III. PATRONES DE DISEÑO</p> <p>3.1 Patrones creacionales 3.1.1 Concepto de patrón de diseño 3.1.2 Fabrica abstracta (Abstract Factory) 3.1.3 Constructor virtual (Builder) 3.1.4 Método de fabricación (Factory method) 3.1.5 Prototipo (Prototype) 3.1.6 Instancia única (Singleton)</p> <p>3.2 Patrones estructurales 3.2.1 Adaptador o envoltorio (Adapter o wrapper) 3.2.2 Puente (Bridge) 3.2.3 Objeto compuesto (Composite) 3.2.4 Decorador (Decorator) 3.2.5 Fachada (Facade) 3.2.6 Peso ligero (Flyweight) 3.2.7 Proxy</p> <p>3.3 Patrones de comportamiento 3.3.1 Cadena de responsabilidad (Chain of responsibility) 3.3.2 Orden (Command) 3.3.3 Interprete (Interpreter) 3.3.4 Iterador (Iterator) 3.3.5 Mediador (Mediator) 3.3.6 Recuerdo (Memento)</p>	<p>Define el concepto de patron de diseño para la solucion de problemas conocidos en el ambito de la programación orientada a objetos.</p> <p>Desarrolla programas simples empleando los diferentes patrones de diseño para solucionar problemas especificos con programación orientada a objetos.</p>

	3.3.7 Observador (Observer) 3.3.8 Estado (State) 3.3.9 Estrategia (Strategy) 3.3.10 Método Plantilla (Template method) 3.3.11 Visitante (Visitor)	
--	---	--

OBJETO DE ESTUDIO	METODOLOGIA (Estrategias, secuencias, recursos didácticos)	EVIDENCIAS DE APRENDIZAJE.
I. ANALISIS Y DISEÑO ORIENTADO A OBJETOS II. EL LENGUAJE DE MODELADO UNIFICADO (UML) III. PATRONES DE DISEÑO	Aprendizaje interactivo (exposición del profesor) Grupo de discusión. Auto aprendizaje (búsqueda y análisis de información) Inductivo <ul style="list-style-type: none"> • Observación • Comparación Deductivo <ul style="list-style-type: none"> • Aplicación • Comprobación • Demostración Sintético <ul style="list-style-type: none"> • Recapitulación • Definición • Resumen • Esquemas • Modelos matemáticos • Conclusión. Material de Apoyo didáctico: Recursos <ul style="list-style-type: none"> • Manual de Instrucción • Materiales gráficos: artículos, libros, diccionarios, etc. • Cañón • Pintarrones • Equipo de computo 	<ul style="list-style-type: none"> • Tareas y ejercicios. • Escritura de programas de computo. • Exámenes escritos.

FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas)	EVALUACIÓN DE LOS APRENDIZAJES (Criterios e instrumentos)
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.	INSTRUMENTOS: <ul style="list-style-type: none"> • Exámenes • Solución de ejercicios (aplicación de conocimientos) • Lista de cotejo (Respeto y participación al

