

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA



UNIDAD ACADÉMICA:
FACULTAD DE INGENIERÍA

PROGRAMACIÓN ORIENTADA
A OBJETOS

DES:	INGENIERÍA
Programa académico	Ingeniería en Computación. Ingeniería en Ciencias de datos y Matemáticas Aplicadas.
Tipo de materia (Obli/Opta):	Obligatoria
Clave de la materia:	MC401
Semestre:	Cuarto
Área en plan de estudios:	Específica
Total de horas por semana:	6
<i>Teoría: Presencial o Virtual</i>	0
<i>Laboratorio o Taller:</i>	4
<i>Prácticas:</i>	0
<i>Trabajo extra-clase:</i>	2
Créditos Totales:	6
Total de horas semestre (x sem):	96
Fecha de actualización:	Octubre 2024
<i>Prerrequisito (s):</i>	CO303 Programación y Análisis de Algoritmos

DESCRIPCIÓN:

El curso proporciona una comprensión acerca del paradigma orientado a objetos, enfocándose en el modelado, principios y conceptos clave, permitiendo a los estudiantes desarrollar habilidades para la solución de problemas mediante la interpretación de la realidad usando un lenguaje de programación.

COMPETENCIAS PARA DESARROLLAR:

B4. Transformación Digital

Transforma la cultura digital en la sociedad, en las organizaciones e instituciones educativas para aprovechar al máximo el potencial de las tecnologías y herramientas digitales; propiciar su uso responsable y ético que estimule la creatividad, innovación, la comunicación efectiva y el trabajo colaborativo e interdisciplinar en la solución de problemas de la sociedad digital; promoviendo la privacidad y la seguridad, así como el respeto a los derechos de autor y la propiedad intelectual.

P1. CIENCIAS E INGENIERÍA. Aplica los conocimientos y metodologías para el planteamiento y resolución de problemas complejos de las ciencias naturales y de la ingeniería, para la toma de decisiones en un contexto de responsabilidad social y del medio ambiente.

E1. DISEÑO Y DESARROLLO DE SOFTWARE. Utilizar en el diseño y desarrollo de software, integrando algoritmos avanzados y estructuras de datos para crear soluciones de software robustas y de calidad. Implica una comprensión profunda de los principios de programación, un enfoque metódico para la solución de problemas y la capacidad de adaptar y mejorar continuamente las prácticas de desarrollo para satisfacer las cambiantes necesidades tecnológicas y las demandas de los diversos sectores.

DOMINIOS	OBJETOS DE ESTUDIO	RESULTADOS DE APRENDIZAJE	METODOLOGÍA	EVIDENCIAS
<p>P1. CIENCIAS E INGENIERÍA</p> <p>3.Utiliza conceptos, métodos y leyes fundamentales de las ciencias básicas para dar soluciones a problemas complejos de ciencias e ingeniería analizando los resultados para emitir conclusiones en situaciones reales.</p> <p>E1. DISEÑO Y DESARROLLO DE SOFTWARE</p> <p>1.Desarrollar código eficiente, aplicando buenas prácticas de programación y aprovechando las características avanzadas</p>	<p><u>UNIDAD I: FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.</u></p> <p>1.1 Paradigmas de programación 1.1.1 Definición de paradigma de programación. 1.1.2 Comparación entre paradigmas. 1.2 Aspectos generales de la programación orientada a objetos. 1.2.1 Descripción de la realidad mediante objetos (principio de Abstracción). 1.2.2 Estructuras contra objetos. 1.2.3 Problemas que resuelve la programación orientada a objetos.</p> <p>1.3 Introducción a un lenguaje de programación orientado a objetos. 1.3.1 Estilo de programación de los lenguajes de programación. 1.3.2 Lenguajes interpretados, compilados y traducidos. 1.3.3 Configuración del ambiente de</p>	<p>Analiza y desarrolla los conceptos básicos de la teoría de objetos en el paradigma de programación orientado a objetos.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Uso de recursos tecnológicos institucionales ● Uso de un lenguaje de programación y su interfaz de desarrollo. ● Ejercicios prácticos del tema desarrollados en clase en forma individual o grupal. 	<p>Resolución de ejercicios de clase a través de programas computacionales desarrollados en un lenguaje de programación.</p> <p>Presentación de un programa computacional aplicando los conceptos presentados para la finalización del objeto de estudio.</p>

	desarrollo. 1.3.4 Gestores de paquetes.			
--	---	--	--	--

<p>del lenguaje, adquiriendo conocimientos sólidos en la programación utilizando lenguajes y paradigmas de programación relevantes para la industria.</p> <p>B4.1 Desarrolla habilidades digitales de forma crítica que impacten positivamente en la vida cotidiana y en las organizaciones e instituciones para la comunicación efectiva en entornos digitales.</p>	<p>1.3.5 Gestores de construcción.</p> <p>1.4. Modelado de objetos mediante clases (Abstracción).</p> <p>1.4.1 Anatomía de una clase.</p> <p>1.4.2 Propiedades.</p> <p>1.4.3 Instancias.</p> <p>1.4.3 El constructor.</p> <p>1.4.4 Sobrecarga.</p> <p>1.4.5 Métodos de instancia.</p> <p>1.4.6 Métodos de clase.</p> <p>1.4.7 Enumeradores.</p> <p>1.4.8 Concatenación vs interpolación.</p> <p>1.4.9 Recolector de basura.</p>			
---	--	--	--	--

	<p><u>UNIDAD II. OTROS PRINCIPIOS Y ESTRUCTURAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.</u></p> <p>2.1 Encapsulamiento. 2.1.1 El alcance (scope). 2.1.2 Métodos get y set. 2.1.3 Clases envoltorio (wrapper). 2.1.4 Composición. 2.1.5 Delegación.</p> <p>2.2 Herencia. 2.2.1 Concepto de herencia. 2.2.2 Jerarquía de clases. 2.2.3 Sobreescritura. 2.2.4 Utilidad de la herencia.</p>	<p>Emplea los conceptos de herencia, polimorfismo e interfaces en el desarrollo de programas computacionales.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Uso de recursos tecnológicos institucionales ● Uso de un lenguaje de programación y su interfaz de desarrollo. ● Ejercicios prácticos del tema desarrollados en clase en forma individual o grupal. 	<p>Resolución de ejercicios de clase a través de programas computacionales desarrollados en un lenguaje de programación.</p> <p>Presentación de un programa computacional aplicando los conceptos presentados para la finalización del objeto de estudio.</p>
--	---	---	--	---

	<p>2.3 Polimorfismo. 2.3.1 Concepto de polimorfismo. 2.3.2 Propiedades contra comportamientos. 2.3.3 Interfaces. 2.3.4 Clases abstractas. 2.3.5 Reflexión de propiedades y métodos. 2.3.6 Generics. 2.3.5 Utilidad del polimorfismo.</p> <p>2.4 Manejo de excepciones 2.4.1 Concepto de excepciones. 2.4.2 Excepciones predefinidas. 2.4.3 Excepciones definidas por el programador.</p>			
	<p><u>UNIDAD III. EL LENGUAJE UNIFICADO DE MODELADO (UML).</u></p> <p>3.1 Introducción a UML. 3.1.1 Conceptos de diseño de programas orientados a objetos. 3.1.2 UML y su función en el modelado y diseño orientados a objetos. 3.1.3 Conceptos de modelado especificados por UML.</p> <p>3.2 Tipos de diagramas.</p>	<p>Aplica técnicas de modelado para crear diagramas que expresan el diseño de componentes de software.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Uso de recursos tecnológicos institucionales ● Uso de un lenguaje de programación y su interfaz de desarrollo. ● Ejercicios prácticos del tema desarrollados en clase en forma individual o grupal. 	<p>Resolución de ejercicios de clase a través de programas computacionales desarrollados en un lenguaje de programación.</p> <p>Presentación de un programa computacional aplicando los conceptos presentados para la finalización del objeto de estudio.</p>

	3.2.1 Diagrama de clases. 3.2.2 Diagrama de componentes. 3.2.3 Diagrama de estructura compuesta. 3.2.4 Diagrama de estructura paquetes. 3.2.5 Diagrama de estructura de objetos. 3.2.6 Diagrama de actividades. 3.2.7 Diagrama de secuencia. 3.2.8 Diagrama de casos de uso.			
--	---	--	--	--

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
<p>Blanco Fernández, Y. (2020). <i>Introducción a programación orientada a objetos</i>. ISBN 978-8412150902.</p> <p>Guagliano, C. (2019). <i>Programación en Python II: Programación orientada a objetos</i>. RedUsers. ISBN 978-9874958198.</p> <p>Blasco, F. (2019). <i>Programación orientada a objetos en JAVA</i>. RA-MA Editorial. ISBN 978-8499648651.</p> <p>Sebesta, R. W. (2012). <i>Concepts of programming languages</i>. Pearson Education, Inc.</p> <p>Van Roy, P., & Haridi, S. (2004). <i>Concepts and techniques of computer programming</i>.</p> <p>Oviedo Regino, E. M. (2015). <i>Lógica De programación Orientada A Objetos</i>. Ecoe Ediciones.</p>	<p>Primer parcial 30%</p> <ul style="list-style-type: none"> • Desarrollo y revisión de ejercicios en clase 40% • Tareas diversas 20% • Presentación de programa computacional 40% <p>Segundo parcial: 30%</p> <ul style="list-style-type: none"> • Desarrollo y revisión de ejercicios en clase 30% • Tareas diversas 20% • Presentación de programa computacional 50% <p>Tercer Parcial: 40%</p> <ul style="list-style-type: none"> • Desarrollo y revisión de ejercicios en clase 20% • Tareas diversas 20% • Presentación de programa computacional 60% <p>Se usará rúbrica para la entrega y presentación de programa computacional realizado.</p> <p>El lenguaje de programación a utilizar se indicará por medio de la academia correspondiente a la materia.</p> <p>Calificación mínima 7.0</p>

CRONOGRAMA

Objetos de estudio	Semanas															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
UNIDAD I: FUNDAMENTOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.																
UNIDAD II. OTROS PRINCIPIOS Y ESTRUCTURAS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS.																
UNIDAD III. EL LENGUAJE UNIFICADO DE MODELADO (UML).																