


<p style="text-align: center;"><b>UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</b></p>  <p style="text-align: center;"><b>UNIDAD ACADÉMICA: FACULTAD DE INGENIERÍA</b></p> <p style="text-align: center;"><b>PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE: <u>TÉCNICAS AVANZADAS DE PROGRAMACIÓN</u></b></p>	<b>DES:</b>	<b>INGENIERÍA</b>
	<b>Programa académico</b>	Ingeniería en Computación
	<b>Tipo de materia (Obli/Opta):</b>	Obligatoria
	<b>Clave de la materia:</b>	CO501
	<b>Semestre:</b>	Quinto
	<b>Área en plan de estudios:</b>	Específica
	<b>Total de horas por semana:</b>	5
	<i>Teoría: Presencial o Virtual</i>	0
	<i>Laboratorio o Taller:</i>	4
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	1
	<b>Créditos Totales:</b>	5
	<b>Total de horas semestre (x sem):</b>	80
	Fecha de actualización:	Octubre 2024.
<i>Prerrequisito (s):</i>	MC401 Programación Orientada a Objetos	

**DESCRIPCIÓN:**

El curso se enfoca en proporcionar a los estudiantes las habilidades y conocimientos necesarios para diseñar y desarrollar software robusto, mantenible y escalable aplicando patrones de diseño creacionales, estructurales y de comportamiento, así como los principios SOLID 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><b>P1. CIENCIAS E INGENIERÍA.</b></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><b><u>UNIDAD I. INTRODUCCIÓN A LOS PATRONES DE DISEÑO</u></b></p> <p><b>1.1 Principios S.O.L.I.D.</b></p> <p>1.1.1 Principio de responsabilidad única.</p> <p>1.1.2 Principio de abierto/cerrado.</p> <p>1.1.3 Principio de sustitución de Liskov.</p> <p>1.1.4 Principio de segregación de la interfaz.</p> <p>1.1.5 Principio de inversión de dependencias.</p> <p><b>1.2 Introducción a los patrones de diseño.</b></p> <p>1.2.1 Definición de patrón de diseño.</p> <p>1.2.2 Organización de los patrones de diseño.</p>	<p>Identifica y enumera los diferentes tipos de patrones de diseño, comprendiendo los conceptos básicos relacionados con el diseño de software. Además, explica la importancia de los patrones de diseño en el desarrollo de software y comprende la diferencia entre patrones de diseño creacionales, estructurales y de comportamiento.</p>	<ul style="list-style-type: none"> <li>• Clase interactiva Maestro - Alumno.</li> <li>• Recursos tecnológicos institucionales.</li> <li>• Laboratorio de cómputo.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas.</li> <li>• Proyecto.</li> <li>• Examen.</li> </ul>
<p><b>E1. DISEÑO Y DESARROLLO DE SOFTWARE.</b></p> <p>1.Desarrollar código eficiente, aplicando buenas prácticas de programación y aprovechando las características avanzadas del lenguaje, adquiriendo conocimientos sólidos en la programación utilizando lenguajes y paradigmas de programación relevantes</p>	<p><b><u>UNIDAD II. PATRONES CREACIONALES</u></b></p> <p><b>2.1 Introducción</b></p> <p>2.1.1 Problemas relacionados a los patrones creacionales.</p> <p><b>2.2 Patrón singleton.</b></p> <p>2.2.1 Descripción.</p> <p>2.2.2 Casos de uso.</p> <p>2.2.3 Implementación.</p> <p><b>2.3 Patrón prototype.</b></p> <p>2.3.1 Descripción.</p> <p>2.3.2 Casos de uso.</p> <p>2.3.3 Implementación.</p> <p><b>2.4 Patrón constructor.</b></p> <p>2.4.1 Descripción.</p> <p>2.4.2 Casos de uso.</p> <p>2.4.3 Implementación.</p> <p><b>2.5 Patrón factory.</b></p> <p>2.5.1 Descripción.</p> <p>2.5.2 Casos de uso.</p> <p>2.5.3 Implementación.</p>	<p>Aplica patrones creacionales en la resolución de problemas de diseño de software. Además, de analizar situaciones específicas donde los patrones creacionales sean apropiados y justificar su elección, así como evaluar la eficacia y eficiencia de la implementación de patrones creacionales en un proyecto de desarrollo de software.</p>	<ul style="list-style-type: none"> <li>• Clase interactiva Maestro - Alumno.</li> <li>• Recursos tecnológicos institucionales.</li> <li>• Laboratorio de cómputo.</li> </ul>	<ul style="list-style-type: none"> <li>• Tareas.</li> <li>• Proyecto.</li> <li>• Examen.</li> </ul>

para la industria.	<p><b>2.6 Patrón abstract factory.</b></p> <p>2.6.1 Descripción.</p> <p>2.6.2 Casos de uso.</p> <p>2.6.3 Implementación.</p>			
<p><b>B4.2</b> Utiliza de forma responsable las tecnologías de la información, comunicación, conocimiento y aprendizaje (TICCA), en el proceso de construcción de saberes y el desarrollo de proyectos sociales innovadores en el ámbito digital.</p>	<p><b><u>UNIDAD III. PATRONES DE DISEÑO ESTRUCTURALES</u></b></p> <p><b>3.1 Introducción</b></p> <p>3.1.1 Problemas relacionados a los patrones estructurales.</p> <p><b>3.2 Patrón adapter.</b></p> <p>3.2.1 Descripción.</p> <p>3.2.2 Casos de uso.</p> <p>3.2.3 Implementación.</p> <p><b>3.3 Patrón bridge.</b></p> <p>3.3.1 Descripción.</p> <p>3.3.2 Casos de uso.</p> <p>3.3.3 Implementación.</p> <p><b>3.3 Patrón composite.</b></p> <p>3.3.1 Descripción.</p> <p>3.3.2 Casos de uso.</p> <p>3.3.3 Implementación.</p> <p><b>3.4 Patrón decorador.</b></p> <p>3.4.1 Descripción.</p> <p>3.4.2 Casos de uso.</p> <p>3.4.3 Implementación.</p> <p><b>3.5 Patrón facade.</b></p> <p>3.5.1 Descripción.</p> <p>3.5.2 Casos de uso.</p> <p>3.5.3 Implementación.</p> <p><b>3.6 Patrón flyweight.</b></p> <p>3.6.1 Descripción.</p> <p>3.6.2 Casos de uso.</p> <p>3.6.3 Implementación.</p> <p><b>3.7 Patrón proxy.</b></p> <p>3.7.1 Descripción.</p> <p>3.7.2 Casos de uso.</p> <p>3.7.3 Implementación.</p>	<p>Analiza problemas de diseño de software en términos de estructuras y relaciones entre objetos. Identifica oportunidades para aplicar patrones estructurales para mejorar la mantenibilidad y escalabilidad del software.</p>	<ul style="list-style-type: none"> <li>● Clase interactiva Maestro - Alumno.</li> <li>● Recursos tecnológicos o institucionales.</li> <li>● Laboratorio de cómputo.</li> </ul>	<ul style="list-style-type: none"> <li>● Tareas.</li> <li>● Proyecto.</li> <li>● Examen.</li> </ul>
	<p><b><u>UNIDAD IV. PATRONES DE COMPORTAMIENTO</u></b></p> <p><b>4.1 Introducción</b></p>	<p>Diseña e implementa soluciones utilizando patrones de comportamiento en el desarrollo de</p>	<ul style="list-style-type: none"> <li>● Clase interactiva Maestro - Alumno.</li> </ul>	<ul style="list-style-type: none"> <li>● Tareas.</li> <li>● Proyecto.</li> <li>● Examen.</li> </ul>

	<p>4.1.1 Problemas relacionados a los patrones de comportamiento.</p> <p><b>4.2 Patrón cadena de responsabilidad.</b>  4.2.1 Descripción.  4.2.2 Casos de uso.  4.2.3 Implementación.</p> <p><b>4.3 Patrón command.</b>  4.3.1 Descripción.  4.3.2 Casos de uso.  4.3.3 Implementación.</p> <p><b>4.3 Patrón iterador.</b>  4.3.1 Descripción.  4.3.2 Casos de uso.  4.3.3 Implementación.</p> <p><b>4.4 Patrón mediador.</b>  4.4.1 Descripción.  4.4.2 Casos de uso.  4.4.3 Implementación.</p> <p><b>4.5 Patrón recuerdo.</b>  3.5.1 Descripción.  3.5.2 Casos de uso.  3.5.3 Implementación.</p> <p><b>4.6 Patrón observador.</b>  3.6.1 Descripción.  3.6.2 Casos de uso.  3.6.3 Implementación.</p> <p><b>4.7 Patrón estado.</b>  3.7.1 Descripción.  3.7.2 Casos de uso.  3.7.3 Implementación.</p> <p><b>4.8 Patrón strategy.</b>  3.8.1 Descripción.  3.8.2 Casos de uso.  3.8.3 Implementación.</p> <p><b>4.9 Patrón del método plantilla.</b>  3.9.1 Descripción.  3.9.2 Casos de uso.  3.9.3 Implementación.</p> <p><b>4.10 Patrón visitante.</b>  3.10.1 Descripción.  3.10.2 Casos de uso.  3.10.3 Implementación.</p>	<p>sistemas software, creando modelos de comportamiento que incorporen múltiples patrones para resolver problemas complejos de manera efectiva.</p>	<ul style="list-style-type: none"> <li>● Recursos tecnológicos institucionales.</li> <li>● Laboratorio de cómputo.</li> </ul>	
--	--	---	---	--

