


<p style="text-align: center;">UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA</p>  <p style="text-align: center;">UNIDAD ACADÉMICA:</p> <p style="text-align: center;">FACULTAD DE INGENIERÍA</p> <p style="text-align: center;">PROGRAMA ANALÍTICO DE LA UNIDAD DE APRENDIZAJE:</p> <p style="text-align: center;"><u>ARQUITECTURA DE SOFTWARE</u></p>	DES:	INGENIERÍA
	Programa académico	Ingeniería en Computación
	Tipo de materia (Obli/Opta):	Obligatoria
	Clave de la materia:	CO606
	Semestre:	Sexto
	Área en plan de estudios:	Específica
	Total de horas por semana:	5
	<i>Teoría: Presencial o Virtual</i>	0
	<i>Laboratorio o Taller:</i>	4
	<i>Prácticas:</i>	0
	<i>Trabajo extra-clase:</i>	1
	Créditos Totales:	5
	Total de horas semestre (x sem):	80
	Fecha de actualización:	Octubre 2024
<i>Prerrequisito (s):</i>	N/A	
DESCRIPCIÓN:		
<p>Este curso proporciona a los estudiantes una base sólida en los principios, patrones y prácticas esenciales en el diseño de sistemas de software, preparándolos para abordar desafíos arquitectónicos en el desarrollo de software en sus diferentes facetas, combinando diversas técnicas para la resolución de problemas de origen arquitectónico integrando infraestructura tecnológica y aplicaciones informáticas.</p>		
COMPETENCIAS PARA DESARROLLAR:		
<p>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.</p>		
<p>P2. DESARROLLO DE PROYECTOS DE INGENIERÍA. Desarrolla proyectos de ingeniería complejos en sus etapas de planeación, análisis y diseño, utilizando las tecnologías y los principios de la administración para la optimización de los recursos con base en procesos de calidad, mejora continua y teniendo en cuenta la seguridad, el costo del ciclo de vida, el carbono neto cero y la salud según sea necesario, atendiendo las necesidades de sostenibilidad.}</p>		
<p>E3. GESTIÓN DE INFRAESTRUCTURA TECNOLÓGICA. Realizar el análisis, diseño, gestión de infraestructura tecnológica, aplicando conocimientos avanzados en sistemas operativos, redes de dispositivos electrónicos, administración de infraestructura tecnológica y seguridad informática. Se centra en garantizar la eficiencia, seguridad y escalabilidad, requiriendo un enfoque analítico para identificar y solucionar problemas complejos en infraestructuras de TI.</p>		

DOMINIOS	OBJETOS DE ESTUDIO	RESULTADOS DE APRENDIZAJE (Se plantean de los dominios y contenidos)	METODOLOGÍA	EVIDENCIAS
<p>B4.4 Analiza los desafíos éticos en la era digital y promueve el uso seguro y responsable de la tecnología; toma en cuenta la protección de datos personales en el entorno digital.</p> <p>P2. DESARROLLO DE PROYECTOS DE INGENIERÍA.</p> <p>1. Identifica las principales áreas de oportunidad en proyectos complejos de ingeniería para definir estrategias de solución utilizando herramientas tecnológicas y administrativas, para optimizar los procesos de calidad, mejora continua contemplando las normativas aplicables.</p> <p>E3. GESTIÓN DE INFRAESTRUCTURA TECNOLÓGICA</p>	<p>UNIDAD I: FUNDAMENTOS DE ARQUITECTURA DE SOFTWARE</p> <p>1.1 Introducción a la Arquitectura de Software. 1.1.1 Definición de arquitectura de software. 1.1.2 Importancia de la arquitectura en el desarrollo de software. 1.1.3 Tendencias y evolución de la arquitectura de software.</p> <p>1.2 Roles y responsabilidades del arquitecto de software. 2.1.1 Funciones del arquitecto de software en el ciclo de vida del desarrollo 2.2.2 Habilidades y competencias requeridas para un arquitecto de software efectivo.</p> <p>1.3 Principios básicos de diseño y patrones arquitectónicos. 1.3.1 Estilos arquitectónicos: capas, pipes and filters, eventos, etc. 1.3.2 Patrones arquitectónicos comunes (por ejemplo, Modelo-Vista-Controlador, Capas). 1.3.3 Selección de estilos y patrones para un proyecto de software.</p> <p>1.4 Modelado de Arquitecturas.</p>	<p>Se introduce a la Arquitectura de Software, comprendiendo su definición, importancia y evolución, así como los roles y responsabilidades del arquitecto de software, incluyendo sus funciones en el ciclo de vida del desarrollo y las habilidades requeridas.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales. ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

CA.

4. Implementar
y/o administrar
soluciones de
cómputo
basados en la
nube,
utilizando

--

--

--

--

<p>diferentes arquitecturas de software e introducirlo al mundo del gobierno de la infraestructura tecnológica.</p>	<p>1.4.1 Diagramas de arquitectura. 1.4.2 Lenguajes de descripción de arquitectura. 1.4.3 Herramientas para el modelado de arquitecturas.</p>			
	<p><u>UNIDAD II:</u> <u>ARQUITECTURAS MODERNAS</u></p> <p>2.1 Estilos Arquitectónicos. 2.1.1 Monolitos vs. Arquitecturas distribuidas. 2.1.2 Cliente-servidor 2.1.3 Balanceo de cargas. 2.1.4 Arquitectura basada en microservicios. 2.1.5 Arquitecturas orientadas a eventos.</p> <p>2.2 Clean Architecture. 2.2.1 Principios y beneficios. 2.2.2 Separación de preocupaciones. 2.2.3 Dependencias inversas y regla de la arquitectura limpia. 2.2.4 Ventajas de la arquitectura limpia en términos de mantenibilidad y escalabilidad.</p> <p>2.3 Arquitecturas emergentes. 2.3.1 Arquitectura hexagonal. 2.3.2 Arquitectura de puertos y adaptadores.</p> <p>2.4 Principios de diseño detallado en arquitecturas modernas.</p>	<p>Implementa arquitecturas modernas mediante el uso de estructuras que permiten la comprensión de los efectos de una arquitectura de software en el desempeño de aplicaciones con énfasis en la separación de preocupaciones, coherencia y cohesión.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales. ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

	<p>2.4.1 Estrategias para lograr una separación de preocupaciones efectiva.</p> <p>2.4.2 Coherencia y cohesión en el diseño de componentes y módulos</p>			
	<p><u>UNIDAD 3:</u> <u>TENDENCIAS EN ARQUITECTURA DE SOFTWARE</u></p> <p>3.1 Arquitecturas para Big Data. 3.1.1 Herramientas. 3.1.2 Arquitecturas de Lambda y Kappa. 3.1.3 Data lakes y data warehouses.</p> <p>3.2 Arquitecturas para Internet de las Cosas (IoT) 3.2.1 Edge computing y fog computing. 3.2.2 Arquitecturas RESTful para IoT. 3.2.3 Seguridad y privacidad en IoT.</p> <p>3.3 Arquitecturas para Inteligencia Artificial y Machine Learning. 3.3.1 Herramientas. 3.3.2 Arquitecturas para deep learning.</p>	<p>Explora las tendencias en arquitectura de software, incluyendo arquitecturas para tecnologías emergentes, junto con sus herramientas asociadas y consideraciones de seguridad y privacidad.</p>	<ul style="list-style-type: none"> ● Clase interactiva Maestro - Alumno. ● Recursos tecnológicos institucionales. ● Laboratorio de cómputo. 	<ul style="list-style-type: none"> ● Tareas. ● Proyecto. ● Examen.

FUENTES DE INFORMACIÓN	EVALUACIÓN DE LOS APRENDIZAJES
<ul style="list-style-type: none"> ● Woods, E., Erder, M., & Pureur, P. (2021). <i>Continuous architecture in practice: Software architecture in the age of agility and DevOps</i>. Addison-Wesley Professional. ISBN 9780136523567. ● Laplante, P. A., & Kassab, M. (2022). <i>What every engineer should know about software engineering</i> (2nd ed.). CRC Press. ISBN 9781003218647. ● Englander, I., & Wong, W. (2021). <i>The architecture of computer hardware, systems software, and networking: An information technology approach</i> (6th ed., Illustrated). John Wiley & Sons. ISBN 978-1119495208. ● Bass, L., Clements, P., & Kazman, R. (2013). <i>Software architecture in practice</i> (3rd ed.). Boston, MA: Addison- Wesley. ● Garlan, D., & Shaw, M. (2016). <i>Software architecture: a modern approach</i> (2nd ed.). New York, NY: Taylor & Francis. ● Sommerville, I. (2015). <i>Software engineering</i> (9th ed.). Harlow, UK: Pearson Education. ● Rhoton, J. (2016). <i>Cloud computing concepts, technology & architecture</i>. Upper Saddle River, NJ: Pearson Education. 	<p>Parcial 1 (30%)</p> <ul style="list-style-type: none"> ● Tareas. (20%) ● Proyecto. (50%) ● Examen. (30%) <p>Parcial 2 (30%)</p> <ul style="list-style-type: none"> ● Tareas. (20%) ● Proyecto. (50%) ● Examen. (30%) <p>Parcial 3 (40%)</p> <ul style="list-style-type: none"> ● Tareas. (20%) ● Proyecto. (50%) ● Examen. (30%) <p>La calificación mínima es 7.0.</p> <p>Se usará rúbrica para la entrega de actividades o tareas a realizar.</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 ARQUITECTURA DE SOFTWARE																	
UNIDAD II: ARQUITECTURAS MODERNAS																	

