



1. Datos Generales de la asignatura

Nombre de la asignatura:	Arquitectura de Software
Clave de la asignatura:	DGD-2202
SATCA¹:	2-3-5
Carrera:	Ingeniería en Tecnologías de la Información y Comunicaciones

2. Presentación

Caracterización de la asignatura
<p>El propósito de esta asignatura es ofrecer los conocimientos y competencias necesarias para diseñar arquitecturas en el diseño y modelado de software. Que permitan al Ingeniero en Tecnologías de la información y comunicaciones, gestionar proyectos de software en beneficios de los diferentes sectores económicos y sociales.</p> <p>Esta asignatura aporta al perfil del Ingeniero en Tecnologías de la Información y Comunicaciones los conocimientos para manejar los roles y responsabilidades como arquitecto de software, así como aplicar los diferentes paradigmas, estilos y patrones de diseño en la implementación de arquitecturas eficientes y eficaces en el desarrollo de software.</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos



ESTÁNDAR
NACIONAL
DE CALIDAD





Intención didáctica

La asignatura cubre la necesidad que tiene un Ingeniero en Tecnologías de la Información, con perfil de desarrollador de software al enfrentarse al diseño e implementación de una arquitectura de software que se ajuste a la gestión de equipos de desarrollo de software.

Esta materia se organiza en cinco temas. El primer tema hace referencia la importancia de la Arquitectura de Software, además de familiarizar a el estudiantado en los conceptos básicos, definiciones, diferencias entre los diferentes paradigmas, modelos, patrones y estilos arquitectónicos; después, se presentan la importancia de las Vistas en la Arquitectura de Software; luego se analiza la Importancia de los atributos de calidad; posteriormente, se analizan los Modelos de Arquitectura de Software, mas importantes, y finalmente se estudia la forma correcta de realizar la Documentación de Arquitectura de Software

3. Participantes en la actualización, el diseño, consolidación y/o seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México, Campus Apizaco Marzo-2022	José Juan Hernández Mora Juan Ramos Ramos María Janai Sánchez Hernández María Lorena Roldán Flores	Reuniones de trabajo para actualización de módulos de especialidad para la carrera de Ingeniería en Tecnologías de la Información y Comunicaciones. Año 2022

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura

Diseña, implementa y evalúa una arquitectura de software en la gestión de un proyecto de desarrollo de software; considerando las vistas, los atributos de calidad, las estructuras y las partes interesadas en el desarrollo del sistema, que garantice la calidad y eficiencia del producto terminado.





5. Competencias previas

Conoce y aplica modelos y/o técnicas de desarrollo de software con la finalidad de implementar sistemas eficientes en base a requerimientos específicos bajo lineamientos y estándares para el aseguramiento de calidad de sistemas de información

Conoce y aplica y diseña modelos de bases de datos para la automatización de sistemas de información.

Desarrolla aplicaciones web dinámicas del lado cliente y del servidor, considerando la conectividad a orígenes de datos, la interconectividad entre aplicaciones y cómputo en la nube.

6. Temario

No.	Temas	Subtemas
1	Importancia de la Arquitectura de Software	<ul style="list-style-type: none"> • Introducción a la arquitectura de Software • Rol del arquitecto de software <ul style="list-style-type: none"> ○ Partes interesadas en un proyecto de software ○ Importancia de las partes interesadas ○ Aportaciones de las partes interesadas en el proyecto • Evolución de la Arquitectura de Software • Estado del Arte de la Arquitectura de Software
2	Vistas de Arquitectura de Software	<ul style="list-style-type: none"> • Vistas y puntos de vista de la arquitectura de software • Estilos de arquitecturas



SECRETARÍA DE EDUCACIÓN PÚBLICA





		<ul style="list-style-type: none"> Modelos de arquitectura de software Patrones de diseño de Arq de sw Arquitecturas de n capas
3	Importancia de los atributos de calidad	<ul style="list-style-type: none"> Calidad en los modelos de arquitectura Entendiendo los atributos de calidad Atributos de Calidad Otros Atributos de Calidad Seguridad en los modelos de arquitectura de software
4	Principales modelos de Arquitectura de Software	<ul style="list-style-type: none"> Arquitecturas basadas en componentes Arquitecturas de sistemas distribuidos Arquitecturas de sistemas en tiempo real Arquitecturas basadas en servicios (SOA) Arquitecturas basadas en microservicios Diseñando modelos de arquitectura
5	Documentación de Arquitectura de Software	<ul style="list-style-type: none"> Gestión de proyectos de arquitectura de software Importancia de la Documentación en la Arquitectura de Software <ul style="list-style-type: none"> Documentación de Interfaces Documentación y las partes interesadas Documentación del comportamiento Evaluando modelos de arquitectura de software



ESTRATEGIA
NACIONAL
DE CALIDAD





	<ul style="list-style-type: none"> • Evaluando la documentación de la arquitectura • Herramientas de Arquitecturas de Software • Proyecto Final: Caso de Estudio.
--	--

7. Actividades de aprendizaje de los temas

Importancia de la Arquitectura de Software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Comprende la importancia de la arquitectura, identifica el rol del arquitecto y entiende el estado actual de la arquitectura en el proceso de desarrollo de software.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de investigación. • Capacidad de abstracción. análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de liderazgo de equipos de trabajo. • Conocimiento sobre el área de estudio y la profesión. 	<ul style="list-style-type: none"> • Redactar un ensayo sobre las diferentes perspectivas de la arquitectura de software. • Diseñar un mapa conceptual haciendo énfasis en la importancia del Arquitecto de Software y analiza la diferencia entre diseño y arquitectura de software. • Realizar un ensayo del estado del arte y las tendencias de la arquitectura de software.
Vistas de Arquitectura de Software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Comprende los diferentes modelos y patrones de arquitectura de software, e identifica las ventajas y desventajas de cada uno de ellos en el proceso de desarrollo de software.</p>	<ul style="list-style-type: none"> • Realizar un ensayo sobre las ventajas del uso de diferentes vistas o perspectivas en el diseño de la arquitectura de software, enfatizando en su importancia en el diseño.





<p>Genéricas:</p> <ul style="list-style-type: none"> • Conocimiento sobre el área de estudio y la profesión. • Capacidad de investigación. • Capacidad de abstracción, análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de liderazgo de equipos de trabajo. 	<ul style="list-style-type: none"> • Realizar una investigación documental de los principales modelos o patrones de arquitecturas de software, obteniendo descripción, ventajas, desventajas, y características de su aplicación en los proyectos. • Realizar un cuadro comparativo de los principales modelos y/o patrones de arquitectura de software, enfatizando en sus características y aplicación en el diseño. • Diseñar el modelo arquitectónico de un sistema propuesto por el docente como caso práctico, considerando para ello los requisitos funcionales, no funcionales y la aplicación de las diferentes vistas para el diseño de una arquitectura de software.
<p>Importancia de los atributos de calidad</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Comprende la importancia de los atributos y sus características, así como la evaluación de calidad en la arquitectura del software.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis. • Capacidad de organizar y planificar. • Comunicación oral y escrita. • Solución de problemas. • Capacidad para la toma de decisiones. 	<ul style="list-style-type: none"> • Realizar una investigación documental acerca de la importancia de los atributos de calidad que son considerados durante el diseño de un software. • Generar un listado de los principales atributos de calidad, y colocar las principales características. • Comprender y diseñar la arquitectura del proyecto asignado con base en los



100 años
de
independencia





<ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. • Habilidades de investigación. • Capacidad de aprender. • Habilidad para trabajar en forma autónoma y colaborativa. 	<p>atributos de calidad desde la perspectiva de las partes interesadas.</p> <ul style="list-style-type: none"> • Realizar una tabla con los atributos que corresponden para su aplicación y diseñar la medida o escenarios para cada atributo priorizando. • Realizar una investigación documental para elaborar un ensayo acerca de la importancia de la seguridad en los módulos de sistema. • Analizar las amenazas potenciales en materia de seguridad y diseñar casos de abuso.
<p>Principales modelos de Arquitectura de Software</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Analiza y aplica los principales modelos y patrones de arquitectura de software e identifica el más adecuado según el tipo de proyecto de software a desarrollar.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Conocimiento sobre el área de estudio y la profesión. • Capacidad de investigación. • Capacidad de abstracción, análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de liderazgo de equipos de trabajo. 	<ul style="list-style-type: none"> • Realizar una investigación documental de los modelos de arquitectura de software más utilizados en la actualidad y sus aplicaciones en el desarrollo de sistemas; obteniendo descripción y ventajas, desventajas. • Selecciona herramientas de desarrollo para el diseño de los modelos: arquitecturas basadas en componentes, arquitecturas de sistemas distribuidos, arquitecturas de sistemas en tiempo real (IoT), arquitecturas basadas en servicios (SOA) y arquitecturas basadas en microservicios. • Redactar un ensayo sobre: Las ventajas y desventajas de las diversas arquitecturas adicionales para el





	<p>desarrollo de sistemas, enriqueciendo los comentarios de al menos dos fuentes distintas.</p> <ul style="list-style-type: none"> • Diseña al menos dos arquitecturas de los modelos analizados y redacta un informe técnico
Documentación de Arquitectura de Software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Realiza la documentación de la arquitectura correspondiente a interfaces, partes interesadas y comportamiento de un proyecto de software, incluyendo su evaluación.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de abstracción, análisis y síntesis. • Capacidad para identificar, plantear y resolver problemas. • Capacidad de trabajo en equipo. 	<ul style="list-style-type: none"> • Investigar acerca de la importancia de documentar la arquitectura de un proyecto de software. • Analizar los elementos esenciales para estructurar un reporte de Arquitectura de Software y proponer la estructura de documentación de un proyecto de curso. • Diseñar la documentación de la arquitectura del proyecto de software propuesto en el curso.

8. Práctica(s)

<p>Diseñar el modelo arquitectónico utilizando el modelo de las 4+1 vistas de Krunchten y UML, para el sistema asignado en clase, considerando para ello los requisitos funcionales, no funcionales y atributos de calidad.</p> <p>Diseñar el modelo arquitectónico de un sistema asignado en clase, como caso práctico de aplicación de una arquitectura de software, considerando para ello los requisitos funcionales, no funcionales y atributos de calidad.</p>
--





Diseñar el modelo arquitectónico de un sistema asignado en clase como caso práctico de la Arquitectura Cliente-Servidor, considerando para ello los requisitos funcionales, no funcionales y atributos de calidad.

Diseñar el modelo arquitectónico de un sistema asignado en clase con el Modelo Vista Controlador o Presentador, considerando para ello los requisitos funcionales, no funcionales y atributos de calidad.

Diseñar el modelo arquitectónico de un sistema asignado en clase con alguno de los siguientes modelos: arquitecturas basadas en componentes, arquitecturas de sistemas distribuidos, arquitecturas de sistemas en tiempo real (IoT), arquitecturas basadas en servicios (SOA) y arquitecturas basadas en microservicios; considerando para ello los requisitos funcionales, no funcionales y atributos de calidad.

De acuerdo con la arquitectura del proyecto que ha venido desarrollando en clase, y considerando los requerimientos funcionales y no funcionales, realice la propuesta de atributos que puede medir y priorícelos. Ejecute la propuesta de medición o escenarios de los atributos seleccionados, por ejemplo, para medir la disponibilidad del producto considere el tiempo de actividad que tendrá el sistema. Documente sus resultados.

Realizar la documentación del proyecto desarrollado en cada una de sus respectivas etapas. Describiendo con claridad la arquitectura del sistema, dónde como mínimo deberá mostrar la descripción jerárquica, diagrama de módulos, descripción individual de los módulos, dependencias externas, modelo de datos, descripción de servicios y procesos, y otras que se consideren.

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los





estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.

- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

Para evaluar las actividades de aprendizaje se recomienda solicitar las evidencias de su ensayo acerca de la importancia de los atributos de calidad en la arquitectura de software, además de considerar las medidas o escenarios que ejecutó en las métricas seleccionadas.

Se sugiere considerar como una evidencia importante la documentación presentada del proyecto y la información contenida en el mismo.





11. Fuentes de información

1. Bass, Len; Clements, Paul & Kazman, Rick (2021), "Software Architecture in Practice", 4a Ed. Addison-Wesley Professional.
2. Cervantes, Humberto & Kazman, Rick (2016) "Designing Software Architectures: A Practical Approach". Addison-Wesley Professional. USA.
3. Ingeno Joseph (2018), "Software Architect's Handbook: Become a successful, software architect by implementing effective architecture concepts", Packt Publishing.
4. Gorton, Ian (2011), "Essential Software Architecture", Springer, 2ed. USA.
5. Martin, Robert C (2018), "Arquitectura Limpia", Anaya Multimedia
6. Ozkaya M & Erata F. (2020), "A survey on the practical use of UML for different software architecture viewpoints", Information and Software Technology, Elsevier, Volume 121.
7. Richards, Mark (2015), "Software Architecture Patterns, Understanding Common Architecture Patterns and When to Use Them", O'Reilly, USA.
8. Richards, Mark & Ford Neal (2020), "Fundamentals of Software Architecture", O'Reilly Media, Inc.
9. Vance William (2020), "Arquitectura Limpia: 3 en 1 - Arquitectura Limpia Guía para principiantes + Consejos y trucos para el software y la programación + Métodos y estrategias avanzadas para el software y la programación", Joiningthedotstv Limited.

