

1. Datos Generales de la asignatura

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

2. Presentación

Caracterización de la asignatura

En la actualidad, el software esta presente en gran cantidad de objetivos que nos rodean; desde los teléfonos y otros dispositivos que llevamos con nosotros de forma casi permanente, hasta los sistemas que controlan las operaciones de organizaciones de toda índole o los que operan las sondas robóticas que exploran otros planetas. Uno de los factores clave de éxito de los sistemas es su bien diseño; de manera particular; el diseño de lo que se conoce como arquitectura de software.

Esta asignatura aporta al perfil del ingeniero, la implementación de un conjunto de arquitecturas necesarias para razonar sobre el sistema y la mejor arquitectura a ocupar.

La principal aportación que esta asignatura brinda al perfil profesional son:

1. Integrar las soluciones de tecnologías de información a los procesos organizacionales para fortalecer objetivos estratégicos.
2. Aplicar normas, marcos de referencia y estándares de calidad y seguridad vigentes en el ámbito del desarrollo y gestión de tecnologías y sistemas de información. Esta asignatura se relaciona con las siguientes materias.

Fundamentos de Ingeniería de Software

Taller de Ingeniería de Software

¹ Sistema de Asignación y Transferencia de Créditos Académicos

- Gestión de proyectos de Software
- Taller de Bases de datos
- Tópicos de programación

Intención didáctica

Por lo dicho anteriormente, el docente debe guiar el proceso de desarrollo emparejándolo con las unidades de aprendizaje sin perder de vista que cada equipo puede requerir un soporte distinto, es decir, que el docente maneje de forma dinámica el temario para que sea capaz de adaptarlo al contexto de cada equipo de trabajo.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
<p><i>Instituto Tecnológico de Morelia.</i></p> <p><i>Morelia Michoacán Agosto 2018</i></p>	<p><i>M.C. Abel Pintor Estrada</i></p> <p><i>M.C Miriam Zulma Sánchez Hernández</i></p> <p><i>MGTI María Yaneth Vega Flores</i></p>	<p><i>Definición de los programas de estudio para la especialidad.</i></p>

4. Competencia(s) a desarrollar

<p align="center">Competencia general de la asignatura</p>
<p>Brindar a los estudiantes las herramientas para facilitar el uso de metodologías y propuestas arquitectónicas en el proceso de desarrollo de software, con el propósito de aumentar la calidad de las aplicaciones que se construyen.</p>
<p align="center">Competencias específicas</p> <ol style="list-style-type: none"> 1. Explicar las características de una arquitectura en capas y los principios de diseño que rigen la construcción de sistemas software con este tipo de arquitectura 2. Diseñar sistemas software utilizando una arquitectura en capas y orientada a objetos y usando servicios software, partiendo de la especificación de sus requisitos (funcionales y no funcionales) utilizando metodologías clásicas y ágiles. 3. Aplicar correctamente los principios de diseño al realizar el diseño de sistemas software.
<p align="center">Competencias genéricas</p> <p>Competencias instrumentales:</p> <ul style="list-style-type: none"> • Utilizar de forma apropiada teorías, procedimientos y herramientas en el desarrollo profesional de la ingeniería informática en todos sus ámbitos (especificación, diseño, implementación, despliegue -implantación- y evaluación de productos) de manera que se demuestre la comprensión de los compromisos adoptados en las decisiones de diseño. <p>Competencias interpersonales:</p> <ul style="list-style-type: none"> • Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y

realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

Competencias sistémicas:

- Desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario, que se comporten de forma fiable y eficiente, que tengan un desarrollo y mantenimiento asequible y que cumplan normas de calidad, aplicando las teorías, los principios, los métodos y las prácticas de Ingeniería del Software.

5. Competencias previas

- Fundamentos de Ingeniería de Software
- Ingeniería de Software
- Gestión de proyectos de Software
- Programación II
- Taller de Base de Datos

6. Temario

No.	Temas	Subtemas
1	Introducción a la Arquitectura de Software	1.1. Definición de Arquitecturas de Software 1.2. Arquitecturas, Atributos de calidad y Objetivos de Negocios 1.3. Ciclos de Desarrollo de la Arquitectura 1.4. Beneficios de la arquitectura 1.5. El rol del arquitecto
2	Requerimientos: Identificación de drivers arquitectónicos	2.1 Requerimientos 2.2 Requerimientos con distintos tipos y niveles de abstracción 2.3 Drivers Arquitectónicos 2.4 Fuentes de información para la extracción de drivers arquitectónicos 2.5 Métodos para la identificación de drivers arquitectónicos
3	Diseño: Toma de decisiones para crear estructuras	3.1 Diseño y niveles de diseño 3.2 procesos generales de diseño de la arquitectura 3.3 Principios de Diseño 3.4 Conceptos de diseño 3.5 Diseño de las interfaces 3.6 Métodos de diseño de arquitectura



4	<p>Documentación: Comunicar la arquitectura</p>	<p>4.1 Documentación en el contexto de arquitectura de software 4.2 Razones para documentar la arquitectura 4.3 Vistas 4.4 Notaciones 4.5 Método y Marcos conceptuales de documentación de arquitectura</p>
5	<p>Evaluación: Asegurar la calidad en la arquitectura</p>	<p>5.1 Conceptos de evaluación 5.2 Evaluación de la arquitectura 5.3 Principios de la Evaluación 5.4 Características de los métodos de evaluación de arquitectura 5.5 Métodos de evaluación de Arquitecturas</p>

7. Actividades de aprendizaje de los temas

Unidad 1. Introducción a la Arquitectura de Software	
Competencias	Actividades de aprendizaje
<p>Específica(s): Explica qué es la arquitectura lógica y física de un sistema software y las relaciones entre ellas</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Demuestra los conocimientos de los principios de la arquitectura de software • Analiza las metodologías de la arquitectura de software 	<ul style="list-style-type: none"> • Gestionar información sobre la terminología de la arquitectura de software. • Conocer las principales actividades del rol de un arquitecto de software.
Unidad 2. Requerimientos: Identificación de drivers arquitectónicos	
Competencias	Actividades de aprendizaje
<p>Específica(s): Explica las características de una arquitectura los principios de los requerimientos que rigen la construcción de sistemas software</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Analiza los procedimientos para la revisión de requerimientos en las arquitecturas de software 	<ul style="list-style-type: none"> • Promover en el estudiante el desarrollo de actividades intelectuales de inducción – deducción y análisis – síntesis, las cuales encaminen hacia la investigación, la aplicación de conocimientos y la soluciones de redacción apropiada de los requerimientos de acuerdo a la arquitectura.
Unidad 3. Diseño: Toma de decisiones para crear estructuras	
Competencias	Actividades de aprendizaje
<p>Específica(s): Diseña sistemas software utilizando una arquitectura de software, partiendo de la especificación de sus requisitos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Desarrolla y mantener los 	<ul style="list-style-type: none"> • Buscar herramientas Case, para la generación de estrategias competitivas basadas en el diseño de las arquitecturas. • Diseño de modelado de un proyecto utilizando la arquitectura adecuada.

<p>seguimientos principales en el diseño de la arquitectura del software</p>	
<p>Unidad 4. Documentación: Comunicar la arquitectura</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Conoce las formas de documentación para la arquitectura de software</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Identifica y utiliza correctamente los modelos de documentación de las arquitecturas 	<ul style="list-style-type: none"> Con la información obtenida en las unidades anteriores y considerando el proyecto se analizara las mejores prácticas aplicada y se mostrara las técnicas adecuadas para la documentación.
<p>Unidad 5. Evaluación: Asegurar la calidad en la arquitectura</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Evalúa y critica los diseños de los sistemas software y proponer diseños alternativos que mejoren su calidad.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> Aplica métodos de evaluación de las arquitecturas de software para la integración correcta de los sistemas 	<ul style="list-style-type: none"> La conclusión final del curso debe conducir a la evaluación de la calidad de la arquitectura adecuada. Así como emplear los diseños adecuados de calidad.

8. Práctica(s)

<ol style="list-style-type: none"> Elija un escenario ya sea simulado u organizacional donde se pueda detectar alguna problemática para que realicen la implementación de una arquitectura adecuada del proyecto Analiza y documenta la solución dada en la práctica anterior, utilizando modelos, arquitecturas y adapta el resultado obtenido enfocado en un entorno de calidad. Lleva a cabo la gestión del proyecto de software elegido por los equipos de

trabajo, considerando lo siguiente:

3.1 Documenta adecuadamente cada fase

3.2 Integra y justifica un equipo de desarrollo acorde a la metodología seleccionada para el desarrollo del proyecto de software.

3.3 Presenta durante el semestre avances.

3.4 Expone al final del semestre los resultados.

9. Proyecto de asignatura (Para fortalecer las competencias de la asignatura con otras asignaturas)

El proyecto integrador se realizará aplicando las competencias previas y vinculándolas con las competencias de las materias del semestre en curso; el proyecto integrador también debe tener un método de evaluación para acreditar la asignatura.

El proyecto integrador debe considerar las siguientes fases:

1. Contextualización o diagnóstico
2. Fundamentación
3. Planeación
4. Ejecución
5. Evaluación
6. Socialización

10. Evaluación por competencias

La evaluación debe ser continua y cotidiana por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Actividades que permitan la evaluación de conocimientos: cuestionarios, exámenes escritos, exámenes orales, entre otros.
- Actividades que permitan la evaluación de habilidades: Evaluar ejercicios, prácticas, proyectos de desarrollo tecnológico, proyectos de investigación, proyectos a través de la triple hélice, entre otras.
- Actividades que permitan la evaluación de actitudes: participación en clase, entrega puntual de sus asignaciones, puntualidad y asistencia, orden en el grupo, entre otras.
- Utilizar diferentes instrumentos de evaluación y sus respectivas rúbricas, para poder evaluar ampliamente y continuamente los aspectos conceptuales, procedimentales y actitudinales.

11. Fuentes de información

- Nord RL, Tomayko JE, Software architecture-centric methods and agile development.”. IEEE Software 2006
- Pressman, R.S.; Maxim, B.R, Software engineering: a practitioner's approach -, McGraw Hill Higher Education, 2015. ISBN: 9780078022128
- Sommerville, I, Pearson - Addison Wesley, Software engineering -, 2011. ISBN: 9780137053469
- Rumbaugh, J.; Jacobson, I.; Booch, G, Addison-Wesley, The unified modeling language reference manual -, 2005. ISBN: 0321245628
- Shore, J.; Warden, S, O'Reilly Media, The Art of agile development -, 2008. ISBN: 9780596527679