



PLAN DE ESTUDIOS (PE): Licenciatura en Ingeniería en Tecnologías de la

Información

ÁREA: OPTATIVA DISCIPLINARIA

ASIGNATURA: Arquitectura de Software

CÓDIGO: ITIS 611

CRÉDITOS: 6

FECHA: 20 de septiembre de 2018





1. DATOS GENERALES

Nivel Educativo:	<u>Licenciatura</u>
Nombre del Plan de Estudios:	<u>Licenciatura en Ingeniería en Tecnologías de la Información</u>
Modalidad Académica:	<u>Presencial</u>
Nombre de la Asignatura:	<u>Arquitectura de Software</u>
Ubicación:	<u>Nivel Formativo</u>
Correlación:	
Asignaturas Precedentes:	<u>Ingeniería de Software II</u>
Asignaturas Consecuentes:	<u>Ninguna</u>

2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
Horas teoría y práctica <u>Actividades bajo la conducción del docente como clases teóricas, prácticas de laboratorio, talleres, cursos por internet, seminarios, etc.</u> (16 horas = 1 crédito)	3	2	90	6





3. REVISIONES Y ACTUALIZACIONES

Autores:	<i>María Luz Adolfinia Sánchez Gálvez</i> <i>Mario Anzures García</i> <i>Abraham Sánchez López</i>
Fecha de creación:	<i>20 de septiembre de 2018</i>
Fecha de la última actualización:	
Fecha de aprobación por parte de la academia de área, departamento u otro.	
Sinopsis de la revisión y/o actualización:	<i>Curso de nueva creación.</i>

4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	<i>Tecnologías de la Información, Sistemas Computacionales, Ciencias de la Computación</i>
Nivel académico:	<i>Maestría</i>
Experiencia docente:	<i>Mínima de 2 años</i>
Experiencia profesional:	<i>Mínima de 2 años</i>

5. PROPÓSITO: *Desarrollar la arquitectura de software, que suministre la estructura necesaria para el desarrollo de aplicaciones y sistemas eficaces y eficientes.*

6. COMPETENCIAS PROFESIONALES: *Esta asignatura contribuye a las siguientes competencias:*

8. Desarrolla proyectos de software en grupos multidisciplinarios de trabajo, mediante la aplicación de la capacidad crítica, de análisis y síntesis con la finalidad de generar innovaciones tecnológicas que atiendan las problemáticas del contexto social, tecnológico, ambiental y/o cultural que lo rodean.
2. Diseña e integra elementos que facilitan la interacción-humano computadora, para el desarrollo de sistemas y aplicaciones computacionales desde una perspectiva ética y en apego a las normas y estándares establecidos.
4. Desarrolla plataformas de software de manera interdisciplinaria para hacer más eficientes los procesos de las organizaciones mediante las Tecnologías de la Información.





Justificación

La arquitectura de software sustenta el desarrollo de aplicaciones complejas, permitiendo su adaptación, extensión y evolución a las necesidades cambiantes de los usuarios.

7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Introducción a la Arquitectura de Software	1.1. Definiciones 1.1.1. Vistas. 1.1.2. Estilos. 1.1.3. Patrones de diseño. 1.1.4. Idioms.	Sommerville, I. (2015). Software engineering. 10th Edition. Pearson. Mark, R. (2015). Software Architecture Patterns O'Reilly.
2. Estilos Arquitectónicos	2.1. Estilo centrado en datos y Estilo de llamada y retorno 2.1.1. Sistema orientado a objetos. 2.1.2. Sistemas basados en componentes. 2.1.3. Sistemas basados en capas. 2.1.4. Modelo-Vista-Controlador. 2.2. Estilo de sistemas distribuidos. 2.2.1. Estilo cliente-servidor. 2.2.2. Estilo Objetos distribuidos. 2.2.3. Estilo peer-to-peer. 2.2.3.1. Estilo basado en eventos. 2.2.3.2. Arquitectura orientada a servicios. 2.3. Estilo de arquitecturas de Referencia.	Sommerville, I. (2015). Software engineering. 10th Edition. Pearson. Mark, R. (2015). Software Architecture Patterns O'Reilly. Addy, O. (2015). Learning javascript design patterns. O'Reilly. Heineman, G.T., and Council W.T. (2007) Component-Based Software Engineering: Putting the Pieces Together Erl, T. (2017) Service-Oriented Architecture: Analysis and Design for Services and Microservices 2nd Edition. Prentice Hall.
3. Aplicaciones de la Arquitectura de Software	3.1. Arquitecturas de software para interfaces de usuario. 3.2. Arquitecturas de software para Interfaces de usuario colaborativas. 3.3. Arquitecturas de software en la nube.	Sommerville, I. (2015). Software engineering. 10th Edition. Pearson. Erl, T. (2017) Service-Oriented Architecture: Analysis and Design for Services and Microservices 2nd Edition. Prentice Hall.
4. Desarrollo basado en arquitecturas de softwares	4.1. Elección del estilo arquitectónico. 4.2. Elaboración de la arquitectura de software. 4.3. Definición de los componentes de la arquitectura de software. 4.4. Establecimiento de las relaciones entre los componentes.	Sommerville, I. (2015). Software engineering. 10th Edition. Pearson. Mark, R. (2015). Software Architecture Patterns O'Reilly.





8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS (Enunciada de manera general para aplicarse durante todo el curso)

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> • Lectura y comprensión, • Reflexión, • Comparación, • Resumen. <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> • ABP, • Aprendizaje activo, • Aprendizaje cooperativo, • Aprendizaje colaborativo, • Basado en el descubrimiento. <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> • Aula, • Simuladores. <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> • Visita a empresas. 	<ul style="list-style-type: none"> • Proyector,es, • Plumón y pizarrón, • Software multimedia • Herramientas de modelado • Equipos de cómputo • Pizarrón inteligente • Aplicaciones colaborativas. • Libros y revistas.

9. EJES TRANSVERSALES

Describe cómo se fomenta(n) el eje o los ejes transversales en la asignatura

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	<i>Promover el trabajo en equipo en el desarrollo de Arquitecturas de Software.</i>
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	<i>Promover el uso de herramientas tecnológicas que fomenten el desarrollo de Arquitecturas de Softwares.</i>
Desarrollo de Habilidades del Pensamiento Complejo	<i>Contribuir al proceso de representación de un problema mediante una Arquitectura de Software.</i>
Lengua Extranjera	<i>Lectura de bibliografía especializada en el área de arquitectura de software.</i>
Innovación y Talento Universitario	<i>Representación de la solución de diversos problemas del mundo real basadas en una Arquitectura de Software.</i>





10. CRITERIOS DE EVALUACIÓN *(de los siguientes criterios propuestos elegir o agregar los que considere pertinentes utilizar para evaluar la asignatura y eliminar aquellos que no utilice, el total será el 100%)*

Criterios	Porcentaje
▪ <i>Participación en clase</i>	<i>10 %</i>
▪ <i>Tareas</i>	<i>20%</i>
▪ <i>Exposiciones</i>	<i>20%</i>
▪ <i>Prácticas de laboratorio</i>	<i>20%</i>
▪ <i>Proyecto final</i>	<i>30%</i>
Total	100%
	100%

11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones
La calificación mínima para considerar un curso acreditado será de 6.
No se debe contar con antecedentes comprobados de copia o plagio de prácticas o proyectos durante el curso.
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

Notas:

- a) La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- b) La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.

