

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Fundamentos de la Arquitectura de Software
Carrera:	Ingeniería en Sistemas Computacionales
Clave de la asignatura:	ARC-1301
(Créditos) SATCA ₁	2-2-4

2.- PRESENTACIÓN

Caracterización de la asignatura.

En la última década cambió la visión que los desarrolladores tenían acerca de la construcción de software. Esta nueva visión se llamó "arquitectura". Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables. Este nuevo aspecto hace posible el estudio de sistemas en diferentes niveles de abstracción y la funcionalidad de los mismos, manifestándose de forma clara en las distintas etapas del proceso de desarrollo de software. Por estas razones es que la carrera incorpora esta materia a la malla curricular.

El propósito de esta asignatura es convertirse en una herramienta académica - metodológica para desarrollar destrezas y técnicas de diseño y modelado de software, afianzando conocimientos básicos en el campo de la programación; además de categorizarse como un aspecto relevante dentro de la creación de sistemas orientados a objetos actuales.

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales los conocimientos para manejar los roles y responsabilidades como arquitecto de software, así como manejar los fundamentos de estilos y patrones de diseño y arquitectura en el desarrollo de software.

Intención didáctica.

Inicia con la definición de conceptos fundamentales en el campo de la Arquitectura de Software como estilos, patrones; abordando temas como estilos Arquitectónicos, Lenguajes de Descripción Arquitectónica, Modelos de Procesos y Diseños, y finaliza con los diagramas UML. Pone énfasis en los distintos Lenguajes de Descripción Arquitectónica (ADL's), llegando a elaborar varios de

los diagramas UML, utilizando 2 tipos de herramientas distintas.

En la segunda unidad se estudian los conceptos de estilos y patrones de diseño, la importancia de la documentación y los diferentes tipos de estilos arquitectónicos.

En la unidad tres se estudia el Modelo de 4+1 Vistas, se analiza cada una de las vistas del modelo, los escenarios, las correspondencias entre las vistas, modelado y documentación.

En la unidad cuatro se estudian los Lenguajes de descripción arquitectónica ADL, sus características, sus componentes y sus restricciones. Se realiza una matriz comparativa.

En la quinta unidad, se estudian los roles y responsabilidades del arquitecto de software.

3.- COMPETENCIAS A DESARROLLAR

Competencias específicas:	Competencias genéricas
<p>Conocer los conceptos fundamentales en el campo de la Arquitectura de Software como estilos, patrones; abordando temas como estilos Arquitectónicos, Lenguajes de Descripción Arquitectónica, Modelos de Procesos y Diseños, y finaliza con los diagramas UML. Pone énfasis en los distintos Lenguajes de Descripción Arquitectónica (ADL's), llegando a elaborar varios de los diagramas UML, utilizando 2 tipos de herramientas distintas.</p>	<p><u>Competencias instrumentales:</u></p> <ul style="list-style-type: none">● Capacidad de análisis y síntesis.● Capacidad de organizar y planificar.● Comunicación oral y escrita.● Habilidad para buscar y analizar información proveniente de fuentes diversas.● Solución de problemas.
<p>Conocer los conceptos de estilos y patrones de diseño, la importancia de la documentación y los diferentes tipos de estilos arquitectónicos.</p>	<p><u>Competencias interpersonales:</u></p> <ul style="list-style-type: none">● Trabajo en equipo.● Capacidad crítica y autocrítica.● Capacidad de venta de ideas. <p><u>Competencias sistémicas:</u></p> <ul style="list-style-type: none">● Capacidad de aplicar los conocimientos en la práctica.● Habilidades de investigación.● Capacidad de aprender.● Capacidad de generar nuevas

<p>Estudiar el Modelo de 4+1 Vistas, se analizan cada una de las vistas del modelo, los escenarios, las correspondencias entre las vistas, modelado y documentación.</p> <p>Conocer los Lenguajes de descripción arquitectónica ADL., sus características, componentes y restricciones. Se realiza una matriz comparativa y se evalúa.</p> <p>Conocer los roles y responsabilidades del arquitecto de software.</p>	<p>ideas (creatividad).</p> <ul style="list-style-type: none"> ● Habilidad para trabajar en forma autónoma. ● Búsqueda del logro.
---	---

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
<p>Hermosillo, Sonora a 26 de febrero del 2013</p>	<p>L.I. Martha Alicia Romero Dueñas</p> <p>L.I. Víctor Manuel Jaime Burgos</p> <p>M.C. Jorge David Gutiérrez Cota</p> <p>Ing. Ana Luisa Millán Castro</p> <p>M.D.R Maria del Carmen Islas Sepulveda</p>	

5.- OBJETIVO(S) GENERAL(ES) DEL CURSO (competencias específicas a desarrollar en el curso)

Conocer los conceptos y generalidades de la arquitectura de software, así como los roles de un arquitecto de software.

6.- COMPETENCIAS PREVIAS

- Identificar y analizar necesidades de información para su representación, tratamiento y automatización para la toma de decisiones.
- Diseñar esquemas de bases de datos para generar soluciones al tratamiento de información
- Desarrollar soluciones de software utilizando programación concurrente, programación de eventos, que soporte interfaz gráfica e incluya dispositivos móviles.
- Desarrollar soluciones de software, considerando los aspectos de negocios, mediante la aplicación de la metodología adecuada a la naturaleza del problema.

7.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción.	1.1. Antecedentes históricos. 1.2. Definiciones y delimitación de la disciplina. 1.3. Conceptos fundamentales. 1.4. Campos de la Arquitectura de Software. 1.5. Modalidades y tendencias. 1.6. Diferencias entre Arquitectura y Diseño. 1.7. Repositorios. 1.8. Problemas pendientes en Arquitectura de Software. 1.9. Relevancia de la Arquitectura de Software.

2	Estilos y patrones.	<p>2.1. Definiciones de estilo.</p> <p>2.2. Clasificaciones de estilos arquitectónicos.</p> <p>2.3. Inventario y descripción de estilos arquitectónicos.</p> <p>2.4. Estilos y patrones de arquitectura y diseño.</p> <p>2.5. El lugar de los estilos en los marcos de referencia y en las vistas arquitectónicas.</p> <p>2.6. Los estilos como valor contable arquitectónicos.</p> <p>2.7 Diseño basado en patrones.</p> <p>2.8 Documentación.</p> <p>2.9 Tipos de arquitectura y sus capas</p> <p> 2.9.1 Cliente/Servidor</p> <p> 2.9.2 Capas</p> <p> 2.9.3 SOA</p>
3	Modelos y escenarios: 4+1 vistas.	<p>3.1 Introducccion.</p> <p>3.2 El Modelo de 4+1 Vistas.</p> <p>3.3 La Arquitectura Lógica.</p> <p>3.4 La Vista de Procesos.</p> <p>3.5 Vista de Desarrollo.</p> <p>3.6 Arquitectura Física.</p> <p>3.7 Escenarios.</p> <p>3.8 Correspondencia entre las Vistas.</p> <p>3.9 Confeccionando el Modelo.</p> <p>3.10 Documentación de la Arquitectura. arquitectónicos.</p>
4	Lenguajes de descripción arquitectónica ADL.	<p>4.1. Introducción a los ADL.</p> <p>4.2. Criterios de definición de un ADL.</p> <p>4.3. Lenguajes: Acme / Armani ADLs: Acme /Armani – ADML – Aesop – ArTek – C2 SADL – CHAM – Darwin – Jacal – LILEANNA – MetaH / AADL – Rapide – UML – UniCon – Wright –xArch / xADL.</p> <p>4.4. Modelos computacionales y paradigmas de modelado.</p> <p>4.5 Componentes y restricciones.</p> <p>4.6 ADLs y Patrones.</p>

5	Rol y responsabilidades del arquitecto de software.	5.1 Características y Competencias del Arquitecto de Software. 5.2 Responsabilidades. 5.2.1 Fases en que participa. 5.2.2 Tipos de Arquitecto.
---	---	---

8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)

El profesor debe:

- Fomentar el uso de las tecnologías de información y comunicación.
- Proyección de videos sobre sistemas de calidad para reflexionar y elaborar resúmenes y conclusiones sobre ellos.
- Realizar visitas a empresas que para observar los roles del Arquitecto de software, los modelos utilizados y alcances de las actividades de arquitectura de software.
- Realizar viajes de prácticas a empresas que apliquen ingeniería del software con el fin de conocer que normas tendrían que usar si deciden crear una empresa desarrolladora de software.
- Solicitar al estudiante que realice investigaciones en diversas fuentes de información sobre arquitectura de software.
- Relacionar los contenidos de esta asignatura con los obtenidos en las demás del plan de estudios y de la especialidad, reforzando la importancia de tener una visión y práctica interdisciplinaria para alcanzar las metas académicas, profesionales y empresariales.
- Elaborar un conjunto de casos prácticos.
- Cuando los temas lo requieran, utilizar medios audiovisuales para una mejor comprensión del estudiante.
- Integrar equipos de trabajo en donde se compartan conocimientos y experiencias académicas y laborales.
- Discutir en grupo la información generada por los equipos de trabajo.
- Propiciar el uso de las diferentes fuentes de información, tanto de índole primaria como secundaria.
- Elaboración de rúbricas.

9.- SUGERENCIAS DE EVALUACIÓN

- Solución de casos prácticos solicitados durante las actividades, así como sus conclusiones de forma escrita.
- Reportes de investigación de campo.
- Reportes de prácticas
- Ejercicios realizados.
- Tareas.
- Exposición.
- Participación en clase.
- Proyecto.
- Exámenes teóricos y/o prácticos.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer los conceptos fundamentales en el campo de la Arquitectura de Software como estilos, patrones; abordando temas como estilos Arquitectónicos, Lenguajes de Descripción Arquitectónica, Modelos de Procesos y Diseños, y finaliza con los diagramas UML.</p>	<ul style="list-style-type: none"> ● Investigar los Antecedentes históricos que han dado lugar a las prácticas de la Arquitectura de Software hoy en día. ● Realizar un mapa mental de los conceptos fundamentales de la Arquitectura de Software ● Realizar un cuadro sinóptico de las diferencias entre Arquitectura y Diseño de Software ● Discutir en el grupo a manera de mesa redonda a cerca de los campos de la arquitectura de software, modalidades y tendencias. ● Realizar un ensayo a cerca de la relevancia de la Arquitectura del Software en el desarrollo Regional y en el desarrollo global del software ● Elaborar varios de los diagramas UML, utilizando 2 tipos de herramientas distintas. ● Justificar el uso de las herramientas elegidas para los diagramas UML

Unidad 2: Estilos y patrones

Competencia específica a desarrollar	Actividades de Aprendizaje

<p>Conocer los conceptos de estilos y patrones de diseño, la importancia de la documentación y los diferentes tipos de estilos arquitectónicos.</p>	<ul style="list-style-type: none"> ● Identificar y explicar las características de un modelo de referencia, estilo o patrón de arquitectura, arquitectura de referencia y arquitectura de software. ● Reconocer algunos de los patrones de arquitectura más representativos, aplicables en el diseño de una arquitectura de software. ● Mapear los componentes de base de una arquitectura de software sobre los componentes de los modelos de referencia, patrones y estilos de arquitectura.
---	---

Unidad 3: Modelos y escenarios: 4+1 vistas

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer y manejar el Modelo de 4+1 Vistas.</p>	<ul style="list-style-type: none"> ● Analizar cada una de las vistas del modelo, los escenarios, las correspondencias entre las vistas, modelado y documentación y plasmar.

Unidad 4: Lenguajes de descripción arquitectónica ADL

Competencia específica a desarrollar	Actividades de Aprendizaje
<p>Conocer los Lenguajes de descripción arquitectónica ADL, sus características, componentes y restricciones. .</p>	<ul style="list-style-type: none"> ● Investigará las características principales de los ADL y los criterios de Definición de los mismos. ● Crear un mapa conceptual de los lenguajes mas relevantes de descripción arquitectónica con sus principales características. <p>(Faltan actividades de aprendizaje al respecto de: Modelos computacionales y paradigmas de modelado. Componentes y restricciones. ADLs y Patrones.)</p> <ul style="list-style-type: none"> ● Realizar una matriz comparativa, evaluarla y discutir en grupo.

Unidad 5: Rol y responsabilidades del arquitecto de software

Competencia específica a desarrollar	Actividades de Aprendizaje

<p>Conocer los roles y responsabilidades del arquitecto de software.</p>	<ul style="list-style-type: none"> ● Investigar las Características y competencias del Arquitecto de Software ● Identificar en un Diagrama, las fases en que participa el Arquitecto de Software ● Identificar en un proyecto de Software desarrollado en alguna asignatura anteriormente, las fases en que participa el Arquitecto del Software y los tipos de arquitectos que pueden trabajar en ese proyecto de manera justificada.
--	---

11.- FUENTES DE INFORMACIÓN

Fuentes impresas (libros)

Bibliografía

- Booch, Grady. *Object-Oriented Analysis and Design*. Second Edition. Benjamin/Cummings, Redwood: 1994.
- Jacobson, Ivar, Grady Booch, and James Rumbaugh. *El Proceso Unificado de Desarrollo de Software*. México: Addison-Wesley, 1999.
- Kruchten, Philippe. "Architectural Blueprints--The 4+1 View Model of Software Architecture". IEEE Software, Institute of Electrical and Electronics Engineers. November 1995, pp. 42-50.
- Larman, Craig. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. México: Prentice Hall, 1999.
- Martin, Robert C. "Design Principles and Design Patterns". [Objectmentor](#)
- Muller, Pierre-Alain. *Modélisation Object avec UML*. Paris: Eyrolles, 1997.
- Wilson, Scott F. *Analyzing Requirements and Defining Solution Architectures*. Redmond: Microsoft Press, 1999.
- Fernández Aramayo, David Ricardo. *Arquitectura de Software*. Universidad Tecmilenio, ITESM
- Zapata Sanchez, Andres felipe. *Arquitectura de Software* www.fi.uba.ar
- Meylin Sigvas Villavicencio www.unpmsn.org

- [1] Rozanski, N., Woods, E., "Software Systems Architecture", Addison Wesley, 2005
- [2] Bass, L. Clements, P., Kazman, R., "Software Architecture in Practice", Addison-Wesley, Second Edition, 2006.
- [3] Paul Clements et al, "Documenting Software Architectures: Views and Beyond", Addison Wesley, 2002.
- [4] Paul Clements et al, "Evaluating Software Architectures", Addison Wesley, 2002.
- [5] Erl, T., "SOA Principles of Service Design", Prentice Hall, 2008
- [6] Richard Taylor, Nenad Medvidovic, Eric Dashofy. "Software Architecture Foundations, Theory and Practice, 2009.

12.- PRÁCTICAS PROPUESTAS