

1. Datos Generales de la asignatura

Nombre de la asignatura:	Arquitecturas de Software para la Web.
Clave de la asignatura:	IWF-1503
SATCA¹:	3- 2- 5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

La materia aporta al perfil del Ingeniero en Sistemas Computacionales la capacidad para conocer los distintos tipos de estilos de arquitecturas de software, la clasificación y características de los patrones de diseño y como éstos se aplican en el proceso de desarrollo de aplicaciones Web. Así también para tomar decisiones en el modelado de arquitecturas de software empleando herramientas CASE. La ventaja de conocer los distintos tipos de estilos de arquitectura es que el estudiante trabaje con las herramientas más actuales de desarrollo y pueda comparar entre ellas para determinar la más adecuada a un entorno particular.

Esta asignatura aporta al perfil del egresado un conjunto de actividades relacionada con la toma de decisiones, a menudo de naturaleza estructural, comparte con la programación una preocupación relacionada con abstraer la representación de la información y las secuencias del procesamiento, pero el grado de detalle es muy diferente en los extremos, el diseño construye representaciones coherentes y bien planeadas de los programas, que se concentran en las interrelaciones entre las partes al nivel más elevado y las operaciones lógicas en los niveles inferiores.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Orizaba Agosto de 2011	Dr. Ulises Juárez Martínez M.C. Ana María Chávez Trejo	Análisis y adecuación por competencias del módulo de la especialidad "Ingeniería Web" de la carrera de Ingeniería en Sistemas Computacionales.
Instituto Tecnológico de Orizaba Enero de 2015	Dr. Giner Alor Hernández	

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<p>Identificar los conceptos de estilo de arquitectura y arquitectura software, el subproceso de desarrollo de la arquitectura software y su incorporación al proceso de desarrollo del sistema, la técnica de modelado y los atributos de calidad a considerar en el desarrollo de la arquitectura de software.</p> <p>Aplicar los conceptos de los diferentes estilos de arquitectura software fundamentales, así como de los patrones arquitectónicos en casos prácticos de modelado.</p> <p>Aplicar los conceptos del estilo arquitectónico orientado a servicios en casos prácticos de modelado.</p> <p>Aplicar los conceptos del estilo arquitectónico orientado a eventos en casos prácticos de modelado.</p> <p>Aplicar el uso de patrones de diseño en el diseño e implementación de software</p>

5. Competencias previas

<p>Analizar, diseñar y desarrollar soluciones de problemas reales utilizando algoritmos computacionales para implementarlos en un lenguaje de programación orientado a objetos.</p> <p>Diseñar e implementar objetos de programación que permitan resolver situaciones reales y de ingeniería.</p> <p>Modelar, Diseñar y crear bases de datos relacionales con base en los estándares establecidos, así como los nuevos paradigmas de tratamiento de datos.</p> <p>Diseñar y manipular bases de datos utilizando distintos Gestores de Bases de Datos</p>

considerando elementos de integridad y seguridad para el tratamiento de la información en distintas plataformas.

Implementar mecanismos de seguridad y disponibilidad de la base de datos.

Conocer la arquitectura general de aplicaciones web

Programar aplicaciones Web empleando lenguajes del lado del servidor y del lado del cliente

6. Temario

No.	Temas	Subtemas
1	Introducción a la Arquitectura de Software	1.1. Definición y conceptos básicos de Arquitectura de Software 1.2. Notaciones para describir la Arquitectura de Software. 1.3. El proceso de desarrollo de la Arquitectura de Software. 1.4. Arquitectura y atributos de calidad del sistema 1.5. Calidad en la arquitectura de software
2	Diseño de arquitecturas	2.1. Introducción a los estilos y patrones arquitectónicos 2.2. Estilos arquitectónicos 2.2.1 Estilos influenciados por los lenguajes tradicionales. 2.2.2 Estilos en capas 2.2.3 Estilos por flujos de datos. 2.2.4 Estilos de estado compartido 2.2.5 Estilos de interpretación 2.2.6 Estilos de invocación implícita. 2.2.7 Peer to peer. 2.3 Patrones arquitectónicos 2.2.1 SLD – State/Logic/Display 2.2.2 MVC- Model/View/Controller 2.2.3 SCC – Sensor/Compute/ Control. 2.4 Aplicaciones basadas en componentes. 2.5 Líneas de productos software
3	Arquitecturas orientadas a servicios	3.1. Introducción a la Arquitectura SOA. 3.2 Historia y evolución 3.3 Conceptos básicos

		<p>3.4. Modelo arquitectónico</p> <p>3.5 Revisión y análisis de casos de estudio.</p> <p>3.6 Modelado de una arquitectura orientada a servicios empleando UML y herramientas CASE</p>
4	Arquitecturas conducidas por eventos	<p>4.1. Introducción a la Arquitectura conducida por eventos, EDA</p> <p>4.2 Historia y evolución</p> <p>4.3 Conceptos básicos</p> <p>4.4 Modelo arquitectónico</p> <p>4.5 Revisión y análisis de casos de estudio</p>
5	Patrones de diseño	<p>5.1 Conceptos básicos</p> <p>5.2 Tipos de patrones de diseño</p> <p>5.3 Patrones orientados a la creación</p> <p>5.4 Patrones orientados a la estructura</p> <p>5.5 Patrones orientados al comportamiento</p>

7. Actividades de aprendizaje de los temas

Introducción a la Arquitectura de Software	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Identificar los conceptos de estilo de arquitectura y arquitectura software, el subproceso de desarrollo de la arquitectura software y su incorporación al proceso de desarrollo del sistema, la técnica de modelado y los atributos de calidad a considerar en el desarrollo de la arquitectura de software.</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organizar y planificar.</p> <p>Conocimientos generales básicos.</p> <p>Conocimientos básicos de la carrera.</p> <p>Capacidad de aplicar los conocimientos en</p>	<p>Investigar las diferencias entre estilo de arquitectura, patrón arquitectónico, arquitectura de referencia y arquitectura de software.</p> <p>Discutir sobre las diferentes vistas que aplica la técnica de modelado de arquitecturas software.</p> <p>Investigar las diferentes propuestas para el proceso de desarrollo de una arquitectura software para aplicaciones Web.</p> <p>Investigar las características de calidad a considerar en el desarrollo de la arquitectura de software para aplicaciones Web.</p>

<p>la práctica.</p> <p>Habilidades de investigación.</p> <p>Capacidad de aprender.</p> <p>Habilidad para trabajar en forma autónoma.</p> <p>Capacidad para diseñar y gestionar proyectos.</p> <p>Solución de problemas.</p> <p>Toma de decisiones.</p>	
<p>Diseño de Arquitecturas</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Explicar y aplicar los conceptos de los diferentes estilos de arquitectura software fundamentales, así como de los patrones arquitectónicos en casos prácticos de modelado.</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organizar y planificar.</p> <p>Conocimientos generales básicos.</p> <p>Conocimientos básicos de la carrera.</p> <p>Capacidad de aplicar los conocimientos en la práctica.</p> <p>Habilidades de investigación.</p> <p>Capacidad de aprender.</p> <p>Habilidad para trabajar en forma autónoma.</p> <p>Capacidad para diseñar y gestionar proyectos.</p> <p>Solución de problemas.</p>	<p>Investigar y discutir las características de los diferentes estilos de arquitectura software.</p> <p>Investigar y discutir las características de los diferentes patrones arquitectónicos.</p> <p>Identificar y realizar las actividades en el desarrollo de casos prácticos de modelado de la arquitectura software para aplicaciones Web aplicando los conceptos de patrones arquitectónicos y estilos de arquitectura software.</p> <p>Elaborar los modelos de la arquitectura de software de un caso práctico</p>

Toma de decisiones.	
Arquitecturas orientadas a servicios	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Explicar y aplicar los conceptos del estilo arquitectónico orientado a servicios en casos prácticos de modelado.</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organizar y planificar.</p> <p>Conocimientos generales básicos.</p> <p>Conocimientos básicos de la carrera.</p> <p>Capacidad de aplicar los conocimientos en la práctica.</p> <p>Habilidades de investigación.</p> <p>Capacidad de aprender.</p> <p>Habilidad para trabajar en forma autónoma.</p> <p>Capacidad para diseñar y gestionar proyectos.</p> <p>Solución de problemas.</p> <p>Toma de decisiones.</p>	<p>Investigar y discutir las características de la arquitectura orientada a servicios.</p> <p>Discutir sobre los elementos que caracterizan a la evolución de los diferentes paradigmas de programación y su influencia en la arquitectura software.</p> <p>Investigar y discutir las características del modelo arquitectónico orientado a servicios.</p> <p>Presentar un análisis de diferentes casos de estudio.</p> <p>Identificar y realizar las actividades en el desarrollo de casos prácticos de modelado de la arquitectura software para aplicaciones Web aplicando los conceptos del estilo de arquitectura orientada a servicios.</p> <p>Realizar la invocación de servicios web a través de herramientas no de usuario final.</p> <p>Elaborar servicios web en un lenguaje de programación.</p> <p>Elaborar el programa cliente del servicio web realizado.</p>
Arquitecturas conducidas por eventos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Explicar y aplicar los conceptos del estilo arquitectónico orientado a eventos en casos prácticos de modelado.</p> <p>Genéricas:</p>	<p>Investigar y discutir las características de la arquitectura orientada a eventos.</p> <p>Investigar y discutir las características del modelo arquitectónico orientado a eventos.</p> <p>Presentar un análisis de diferentes casos de éxito de las arquitecturas conducidas</p>

<p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organizar y planificar.</p> <p>Conocimientos generales básicos.</p> <p>Conocimientos básicos de la carrera.</p> <p>Capacidad de aplicar los conocimientos en la práctica.</p> <p>Habilidades de investigación.</p> <p>Capacidad de aprender.</p> <p>Habilidad para trabajar en forma autónoma.</p> <p>Capacidad para diseñar y gestionar proyectos.</p> <p>Solución de problemas.</p> <p>Toma de decisiones.</p>	<p>por eventos.</p> <p>Identificar y realizar las actividades en el desarrollo de casos prácticos de modelado de la arquitectura software para aplicaciones Web aplicando los conceptos del estilo de arquitectura orientada a eventos.</p>
<p>Patrones de Diseño</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <p>Explicar y aplicar el uso de patrones de diseño en el diseño e implementación de software</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organizar y planificar.</p> <p>Conocimientos generales básicos.</p> <p>Conocimientos básicos de la carrera.</p> <p>Capacidad de aplicar los conocimientos en la práctica.</p> <p>Habilidades de investigación.</p> <p>Capacidad de aprender.</p>	<p>Investigar la clasificación y características básicas de un patrón de diseño.</p> <p>Describir un conjunto representativo de diferentes patrones de diseño, con toda la información de la plantilla.</p> <p>Implementar un caso práctico empleando un patrón de creación o un patrón de estructura o un patrón de comportamiento</p> <p>Presentar un estudio de aplicaciones que emplean patrones de diseño, indicando el problema que atienden.</p>

<p>Habilidad para trabajar en forma autónoma.</p> <p>Capacidad para diseñar y gestionar proyectos.</p> <p>Solución de problemas.</p> <p>Toma de decisiones.</p>	
---	--

8. Práctica(s)

<p>Elaboración de ejercicios de modelado usando diagramas de caso de uso en una Herramienta CASE</p> <ul style="list-style-type: none"> •Elaboración de ejercicios de modelado usando diagramas de clase en una Herramienta CASE •Elaboración de ejercicios de modelado usando diagramas de secuencia en una Herramienta CASE •Elaboración de un cuadro comparativo estableciendo las diferencias entre arquitectura de software, estilo arquitectónico y patrón arquitectónico •Desarrollo de una aplicación Web basada en Java que realice las operaciones CRUD usando el estilo arquitectónico MVC •Desarrollo e implementación de un servicio Web basado en el lenguaje de programación Java •Desarrollo e implementación de un servicio Web basado en el lenguaje de programación PHP •Implementación del patrón de diseño Abstract Factory en el lenguaje de programación Java •Implementación del patrón de diseño Adapter en el lenguaje de Programación Java •Implementación del patrón de diseño Decorator en el lenguaje de Programación Java •Implementación del patrón de diseño Observer en el lenguaje de Programación Java
--

9. Proyecto de asignatura

<p>Desarrollo de un proyecto dosificado durante el semestre, involucrando la elaboración, evaluación y documentación de una arquitectura de software, donde se aplique al menos uno de los métodos de evaluación y al menos dos de las vistas del modelo 4+1 utilizando</p>

el Lenguaje Unificado de Modelado.

El objetivo del proyecto 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

La evaluación de la asignatura debe de ser: un proceso integral, continuo y permanente. Tomando en cuenta que toda evaluación debe de cumplir el tener fuentes de información y criterio de evaluación. Como la base de la adquisición de competencias es la resolución de tareas, deberemos de identificar las fuentes de información que nos indiquen como ha resuelto la tarea, a que problemas se ha enfrentado el estudiante y cuáles de estos ha superado.

Por lo anteriormente expuesto se sugiere que al inicio del semestre se establezcan:

- Los criterios de evaluación, que deberán estar bien definidos y consensuados con los estudiantes a fin de que no existan incertidumbres.

- Establecer las fuentes de información que se usarán durante el curso.

La evaluación de la asignatura por unidad se recomienda hacerla con base a la siguiente ponderación:

Examen escrito	30%
Proyecto	40%
Productos de investigación	10%
Reportes de prácticas	10%
Prácticas de laboratorio funcionando	10%

11. Fuentes de información

Como fuentes de información para cada unidad se sugiere:

- Se debe aplicar un examen teórico-práctico para cada unidad. Es responsabilidad del maestro indicar el porcentaje de teoría y práctica que tendrá el examen así como la forma de evaluarlo.
- Reportes de prácticas de laboratorio las que podrán tener entre otras cosas Objetivo, Fundamento teórico, Material y equipo, Indicaciones, Actividades desarrolladas, Resultados obtenidos y Conclusiones
- Proyecto de cada unidad funcionando y con los productos de Ingeniería de Software mínimos indispensables para su definición
- Productos de las investigaciones realizadas que pudieran ser: resúmenes, mapas conceptuales, mapas mentales, conclusiones, entre otros.

LIBROS DE TEXTO:

* Jazayeri, Mehdi., Software architecture for product families : principles and practice / Mehdi Jazayeri, Alexander Ran, Frank van der Linden., Boston : Addison-Wesley,

* Bass, Len., Software architecture in practice / Len Bass, Paul Clements, Rick Kazman., 2nd ed., Boston : México: Addison-Wesley, 2003.

* Kuchana, Partha., Software architecture design patterns in Java / Partha Kuchana., , Boca Raton, FL : Auerbach Publications, 2004

* R.N. Taylor, N. Medvidovic, E.M. Dashofy, Software Architecture: Foundations, Theory, and Practice. John Wiley & Sons, 2009.

* James O. Coplien, Gertrud Bjørnvig, Lean Architecture for Agile Software Development.

John Wiley & Sons, 2010.

* Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., and Stal, M. Pattern-Oriented Software Architecture. A System of Patterns. John Wiley & Sons, 1996.

* Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Systems. Addison-Wesley, 1994

* Bass, L., Clements, P., and Kazman, R. Software Architecture in Practice. Addison-Wesley, Reading Massachusetts, 1998

* Bennett, D. Designing Hard Software. The Essential Tasks. Manning Publication Co., Greenwich, Connecticut, 1997

* Reichtin, E. and Maier M. The Art of Systems Architecting. CRC Press, 1997

LIBROS DE CONSULTA:

* Szyperski, Clemens., Component software : beyond object-oriented programming, 2nd ed., London ; Boston, MA : Addison-Wesley, England, 2002

* Bloch, Joshua., Effective Java : programming language guide / Joshua Bloch ; [foreword by Guy Steele], , Boston : Addison-Wesley

* Braude, Eric J., Software design : from programming to architecture / Eric J. Braude., , Hoboken, NJ : J. Wiley, c2004.

* , Documenting software architectures : views and beyond / Paul Clements ... [et al]., , Boston : Addison-Wesley, c2003.

* Shalloway, Alan., Design patterns explained : a new perspective on object-oriented design / Alan Shalloway, James R. Trott., 2nd ed., Boston, MA : Addison-Wesley, 2005.

* Gorton, Ian., Essential software architecture / Ian Gorton., , Berlin ; New York : Springer, 2006.