

1. Datos Generales de la asignatura

Nombre de la asignatura:	Diseño y desarrollo de sistemas de software complejos
Clave de la asignatura:	INF-1704
SATCA:	3-2-5
Carrera:	Ingeniería en Sistemas Computacionales

2. Presentación

Caracterización de la asignatura

Esta asignatura como parte de la especialidad, contribuye a reforzar el perfil del Ingeniero en sistemas computacionales aportando de forma integral las siguientes habilidades:

- Implementa aplicaciones computacionales para solucionar problemas de diversos contextos, integrando diferentes tecnologías, plataformas o dispositivos.
- Diseña, desarrolla y aplica modelos computacionales para solucionar problemas, mediante la selección y uso de herramientas matemáticas.
- Diseña e implementa interfaces para la automatización de sistemas de hardware y desarrollo de software asociado.
- Coordina y participa en equipos multidisciplinarios para la aplicación de soluciones innovadoras en diferentes contextos.
- Desarrolla y administra software para apoyar la productividad y competitividad de las organizaciones cumpliendo estándares de calidad.

Además permite al estudiante de Ingeniería en sistemas computacionales como futuro profesional ingeniero de software, involucrarse en proyectos integrales que requieran de una visión arquitectónica global del sistema. Esto con el fin de generar destrezas en el diseño y desarrollo de soluciones de software complejas, aplicables en diferentes contextos interconectados, y la formalización de flujos de trabajo para la generación de mejores propuestas de valor en los procesos de negocio.

Intención didáctica

Se organiza el temario de la asignatura en cinco unidades con el propósito de generar un proyecto integral bajo un paradigma de estructuración de sistemas desacoplados en servicios. La materia tiene un fin integrador, pues representa la culminación de la aplicación de los conocimientos adquiridos en las distintas materias relacionadas con el desarrollo de proyectos de software.

En la primera unidad se comprenden algunos conceptos base de la arquitectura de software y su relación con la arquitectura organizacional, como apoyo a la estructuración de sistemas complejos con enfoque en los negocios.

En la segunda unidad se abordan temas relacionados con la gestión de procesos bajo diferentes notaciones, con el propósito de que el docente dirija y represente de forma práctica y gradual, los conceptos a dominar en la asignatura bajo un proyecto integral de software. La representación global del proyecto integral se realizará en BPMN para que los alumnos, una vez integrados en equipos de trabajo, identifiquen y representen bajo la notación adecuada (UML o YAWL) las tareas factibles de automatizar, así como sus interfaces de comunicación para cada subproyecto.

En la tercera unidad se da a conocer el estilo de estructuración de sistemas orientadas a servicios con la finalidad de desacoplar el conjunto de tareas, asociarlas a la gestión de procesos y desacoplarla en servicios y/o microservicios.

En la cuarta unidad se trabaja con la identificación de los diferentes tipos de requerimientos de software para cada servicio y/o microservicio.

En la quinta unidad se documenta la arquitectura del sistema, se completa el desarrollo y se prueba con todas sus interfaces.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico de Hermosillo, Septiembre- Octubre 2016	MSI María de Jesús Velázquez Mendoza Dr. Oscar Mario Rodríguez Elías MC César Enrique Rose Gómez	Reuniones para la propuesta de la nueva Especialidad para la carrera de Ingeniería en Sistemas Computacionales.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> Habilidad para proponer proyectos integrales de software. Desarrollo de proyectos integrales de software.

5. Competencias previas

Habilidad para identificar estrategias de cambio en las propuestas de negocio (SCC-1005).
Desarrollo e implementación de modelos de base de datos (AEF-1031 FBD)
Habilidad para la aplicación de diferentes tipos de operaciones utilizando SQL (AEF-1031 FBD).
Implementación de mecanismos de seguridad y disponibilidad de base de datos (SCB-1001 ABD).

Habilidad para aplicar metodologías para el desarrollo de productos de software (SCD-1011 IS).
 Planificación y gestión de proyectos de software (SCG-1009 GP).
 Aplicación de métricas de calidad en productos de software (SCG-1009 GP).
 Desarrollo de aplicaciones digitales en soluciones de problemas computacionales (SCC-1023 SP).
 Desarrollo de software de sistemas o de aplicación mediante lenguaje de interface (SCC-1023 SP).
 Desarrollo de aplicaciones web (AEB-1055 PW).
 Aplicación de estándares y protocolos de seguridad e interoperabilidad (AEB-1055 PW).

6. Temario

No.	Temas	Subtemas
1	Introducción a los sistemas complejos.	1.1 Que es un sistema complejo. 1.2 Qué es arquitectura de software vs la arquitectura organizacional. 1.3 Ciclo de desarrollo de la arquitectura de software. 1.4 Beneficios de la arquitectura 1.4.1 Objetivos de negocio 1.4.2 Atributos de calidad 1.5 Conceptos fundamentales de la arquitectura 1.5.1 Estilos y patrones 1.5.2 Tácticas 1.5.3 <i>Frameworks</i> 1.5.4 Interfaces
2	Introducción a la gestión de procesos	2.1 Conceptos fundamentales de la gestión de procesos 2.1.1 Proceso 2.1.2 Actividad 2.1.3 Tarea 2.1.4 Instrucciones 2.2 Notación BPM (<i>Bussines Process Management</i>) 2.2.1 Objetos de flujo 2.2.2 Objetos de conexión 2.2.3 Canales 2.2.4 Artefactos 2.3 Otras notaciones 2.3.1 Diagramas de actividad (UML) 2.3.2 Redes Petri (YAWL) 2.4 Patrones de flujo de trabajo
3	Arquitectura Orientada a Servicios	3.1 Definición del estilo arquitectónico

	(SOA)	SOA 3.2 Abstracciones básicas del enfoque SOA 3.2.1 Servicios 3.2.2 Aplicaciones Front-end 3.2.3 Repositorios de servicios 3.2.4 Bus de servicios 3.3 SOA y el modelo de proceso de negocio 3.4 Definición e identificación de microservicios
4	Requerimientos de sistema	4.1 Requerimientos de la arquitectura 4.1.1 Por tipos 4.1.2 Por niveles de abstracción 4.1.3 Por restricción 4.1.4 Atributos de calidad 4.2 Identificación de drivers arquitectónicos 4.2.1 Drivers funcionales 4.2.2 Drivers de atributos de calidad 4.2.3 Drivers de restricción 4.3 Diseño de interfaces
5	Documentación del sistema	5.1 Puntos de vistas 5.1.1 Vista lógica 5.1.2 Vista de comportamiento 5.1.3 Vista física 5.2 Notaciones 5.2.1 Informales 5.2.2 Semiformales 5.2.3 Formales 5.3 Normas de calidad para sistemas e ingeniería de software 5.4 Normas de calidad para la gestión de procesos

7. Actividades de aprendizaje de los temas

1. Introducción a los sistemas complejos	
Competencias	Actividades de Aprendizaje
<p>Aplica los conceptos básicos a utilizar en el desarrollo de sistemas complejos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar los conocimientos en la práctica. • Capacidad para identificar, plantear y 	<p>Aplicará los conceptos básicos a utilizar en el desarrollo de sistemas complejos, a través de:</p> <ul style="list-style-type: none"> • El desarrollo y presentación de una propuesta de proyecto considerando interfaces de comunicación con el resto de los proyectos del grupo.

<p>resolver problemas.</p> <ul style="list-style-type: none"> • Capacidad de analizar información proveniente de diferentes fuentes. • Capacidad de trabajo en equipo. • Comunicación oral y escrita. • Iniciativa y espíritu emprendedor. • Habilidades interpersonales. 	<ul style="list-style-type: none"> • Investigación de patrones arquitectónicos factibles de implementar en el proyecto.
2. Introducción a la gestión de procesos	
Competencias	Actividades de Aprendizaje
<p>Aplica la notación adecuada para la representación de flujos de trabajo en la gestión de procesos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Trabajo en equipo. • Capacidad para identificar, plantear y resolver problemas. • Comunicación oral y escrita. • Iniciativa y espíritu emprendedor. • Habilidades interpersonales. 	<ul style="list-style-type: none"> • Identificar patrones de flujo a utilizar en el proyecto. • Identificar y aplicar la notación adecuada (BPMN, YAWL o UML) de acuerdo al proyecto. • Utilizar alguna herramienta tecnológica para representar los principales flujos de trabajo que mantienen conectividad entre los proyectos del grupo.
3. Arquitectura orientada a servicios	
Competencias	Actividades de Aprendizaje
<p>Aplica y representa de manera adecuada el enfoque orientado a servicios (SOA)</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Trabajo en equipo. • Capacidad para identificar, plantear y resolver problemas. • Comunicación oral y escrita. • Iniciativa y espíritu emprendedor. • Habilidades interpersonales. 	<ul style="list-style-type: none"> • Desarrollar servicios básicos de ejemplo mediante tecnologías de servicios web • Delimitar el flujo de trabajo en servicios del proyecto. • Identificar microservicios del proyecto. • Documentar servicios del proyecto. • Generar el prototipo del proyecto integrando la estructuración general de la base de datos y las interfaces gráficas asociadas a cada servicio.
4. Requerimientos del sistema	
Competencias	Actividades de Aprendizaje
<p>Identifica, documenta e implementa requerimientos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Trabajo en equipo. • Capacidad creativa. • Comunicación oral y escrita. • Capacidad para formular y gestionar proyectos. • Compromiso ético. • Compromiso con la calidad. 	<ul style="list-style-type: none"> • Documentar siguiendo la identificación de requerimientos desde la perspectiva del stakeholder, el sistema y el software (ISO/IEC/IEEE 15289). • Alinear la propuesta de proyecto con los objetivos de negocio (ISO 9000, ubicar la propuesta de valor en la gestión de procesos). • Desarrolla y prueba los requerimientos asociados a cada servicio del proyecto.

<ul style="list-style-type: none"> • Iniciativa y espíritu emprendedor. • Habilidades interpersonales. 	
5. Documentación del sistema	
Competencias	Actividades de Aprendizaje
<p>Documenta adecuadamente la arquitectura del sistema.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Trabajo en equipo. • Capacidad creativa • Comunicación oral y escrita. • Capacidad para formular y gestionar proyectos. • Compromiso con la calidad. • Iniciativa y espíritu emprendedor. • Habilidades interpersonales. 	<ul style="list-style-type: none"> • Definir la estructuración general del sistema respetando el estilo orientado a servicios. • Delimitar la funcionalidad a aplicar de acuerdo al patrón seleccionado. • Definir el desempeño esperado al implementar dicho patrón. • Establecer y probar las interfaces arquitectónicas establecidas para el proyecto. • Exponer sus resultados en evento de presentación de proyectos institucional.

7. Actividades de aprendizaje de los temas

Las actividades de esta asignatura están establecidas para cumplir en el transcurso del semestre, de forma secuencial, con la generación de un proyecto de software integral en donde se puedan fortalecer las competencias previamente adquiridas.

1. Generar una propuesta de proyecto.
2. Documentar proyecto bajo el enfoque de procesos.
3. Delimitar y documentar proyecto por servicios y micros servicios.
4. Documentar requerimientos por servicios.
5. Documentar la arquitectura del sistema.
6. Implementar y probar los servicios.
7. Presentar el proyecto.

Como complemento al programa, se deberán hacer actividades en donde el alumno pueda observar una implementación en tiempo real de un sistema integral:

- Realizar al menos una visita empresarial en donde los alumnos puedan visualizar flujos de trabajo implementando sistemas integrales (software administrativo, control de producción y embebido)

8. Prácticas

- Generar y/o comprender diagramas de flujo de trabajo bajo diferentes notaciones (BPM, YAWL, UML).
- Representación arquitectónica de diferentes casos de estudio.

9. Proyecto de Asignatura

Proyecto integrado por componentes de software con diferente tipos de funcionalidad (servicios), en diferentes niveles de abstracción, y que sus interfaces resuelvan un proceso global representado a través de flujos de trabajo.

Los componentes deberán ser asignados por equipos de trabajo.

10. Evaluación por competencias

- 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.

11.- Fuentes de Información

1. Cervantes, Velasco y Castro (2015), *Arquitectura de software, conceptos y ciclo de desarrollo*, 1ra. Ed. CENGAGE Learning.
2. Rosing, Scheel, y August-Wilhelm Scheer (2015), *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM*, Volume I, 1st Ed. Elsevier.
3. Hitpass (2014). *BPM: Business Process Management - Fundamentos y Conceptos de Implementación*, 3ra Ed., BBH Ltda- Santiago de Chile.
4. OMG Document Number: formal/2011-01-03, *Business Process Model and Notation, v2.0*, Standard document URL: <http://www.omg.org/spec/BPMN/2.0>
5. Arthur H. M. Ter Hofstede, Wil M. P. van der Aalst, Michael Adams (2009), *Modern Business Process Automation: YAWL and Its Support Environment*, Springer.
6. White, S. A. y Derek, M. (2008). *BPMN modeling and reference guide* (Lighthouse.). Future Strategics Inc.
7. Larman C. (2010), *UML y Patronos*, 2da. Ed., Prentice Hall/Pearson.
8. Newman Sam (2015), *Building microservices, designing fine-grained system*, edit O'Reilly.
9. Erl Thomas (2016), *Service-oriented Architecture: Analysis and Design for Services and Microservices*, 2da. Ed. Prentice Hall.
10. Kuchana Partha (2004), *Software Architecture Design Patterns in Java*, AUERBACH PUBLICATIONS.
11. Ian Gordon (2006), *Essential Software Architecture, Ian Gorton*, 2da Ed. Springer.

12. Clements, Bachmann, Bass, Garden, Ivers, Little, Merson, Nord, Stafford (2011), *Documenting Software Architectures: Views and Beyond*, 2da. Ed. Eddison-Wesley.