

Especialidad en Análisis, Diseño y Desarrollo de Software

INTE-ADD-2017-01

1. Datos Generales de la asignatura

Nombre de la asignatura:	Análisis Avanzado de Software
Clave de la asignatura:	ADF-1702
SATCA¹:	3 - 2 - 5
Carrera:	Ingeniería en Tecnologías de la Información y Comunicaciones, Ingeniería en Informática e Ingeniería en Sistemas Computacionales.

2. Presentación

Caracterización de la asignatura

La presente asignatura proporciona al perfil del Ingeniero en Tecnologías de la Información y Comunicaciones, Ingeniero en Sistemas Computacionales e Ingeniero en Informática las competencias para crear y aplicar de las bases principales de la ingeniería de requisitos, la estructura y el comportamiento antes, durante y después de la implementación de software avanzado.

El programa de la asignatura comprende temas de modelado de requisitos, especificación de requisitos de software, análisis y depuración de código y metodologías de desarrollo ágil distribuidos en cuatro temas principales.

El primer tema proporciona al ingeniero las herramientas necesarias para el modelado de requisitos mediante el uso de diagramas UML, diagramas de secuencia y arquitectura 4+1.

Durante el segundo tema, el ingeniero desarrolla las competencias necesarias para una correcta especificación de requisitos de software e identifica los diferentes modelos y estándares aplicables en el análisis y desarrollo de software.

En el tercer tema, el ingeniero conoce y aplica técnicas para el análisis y la depuración de código para su optimización en etapas de desarrollo de software posteriores.

¹ Sistema de Asignación y Transferencia de Créditos Académicos



Por último, el cuarto tema define las metodologías de desarrollo ágil que permiten al ingeniero identificar y elegir la metodología de desarrollo acorde a las necesidades de cada proyecto de software.

Intención didáctica

Conocimiento de mecanismos para la realización del análisis completo de sistemas de software a desarrollar, determinar la importancia de considerar el preanálisis, análisis y requerimiento de usuario.

Integración de metodologías de análisis/diseño especificando requisitos orientados a objetivos indicando el escenario correcto.

Las competencias generales son:

Capacidad para conocer y aplicar procesos y metodologías de desarrollo software basadas en el desarrollo.

Capacidad de capturar, definir y especificar los requisitos de usuario mediante técnicas basadas de diseño centrado en el usuario y en desarrollo dirigido por modelos.

Capacidad para detectar y aplicar las mejores prácticas fundamentadas en la aplicación de los patrones en cualquier fase de desarrollo software.

Capacidad para diseñar e implementar interfaces de usuario siguiendo los requisitos de calidad.

Capacidad para capturar, diseñar e implementar las pruebas necesarias que aseguren la calidad de un sistema software.

Capacidad de diseñar sistemas donde se consideren los aspectos de usabilidad y accesibilidad mediante técnicas de diseño centrado en el usuario.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Tijuana, 13 de junio de 2017.	Academia del Departamento de Sistemas y Computación. Luz María Lugo Méndez, Reynaldo Andrés Salas Carmona, Fernando Elihonaí Saucedo Lares, Claudia Negrete Sánchez, Maribel Guerrero Luis, Diego Rafael Muñoz Fernández, Alfredo López Chaparro, Jacobo Adrián Rodríguez Aguiñaga.	Reunión permanente de la Academia del Depto. de Sistemas y Computación.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> • Conoce y aplica técnicas del desarrollo dirigido por modelos para establecer una clara trazabilidad entre los modelos que especifican el sistema y la implementación final obtenida. • Conoce, aplica y desarrolla aplicaciones de software pertinentes utilizando una metodología de desarrollo dirigida por modelos adecuada. • Utiliza el desarrollo dirigido por modelos a dominios específicos mediante lenguajes de dominio específico (DSL) y mediante la utilización de perfiles sobre el lenguaje de modelado estándar (UML). • Asocia el conjunto de artefactos (especificaciones, diagramas y código) a cada actividad del proceso de desarrollo de software que le son aplicables. • Identifica y aplica los patrones de las mejores prácticas para el desarrollo de aplicaciones de software. • Diseña e implementa interfaces de usuario siguiendo los requisitos de calidad como usabilidad, seguridad y rendimiento. • Analiza los resultados obtenidos para conocer los límites y puntos críticos de la aplicación. • Diseña y desarrolla soluciones basadas en el paradigma de Arquitectura Orientada a Servicios atendiendo a los parámetros de adaptación de las TIC a los cambios en los requerimientos de los procesos de negocio de las organizaciones. • Elabora y analiza la documentación de especificación de requerimientos con base en las necesidades reales de los usuarios. • Conoce y aplica las metodologías ágiles para el análisis y el diseño de software de acuerdo al contexto y las necesidades para la solución de problemas.



5. Competencias previas

Conoce el ciclo de vida de desarrollo de software para su implementación de acuerdo al contexto y las necesidades actuales.

Desarrolla aplicaciones de software para la solución de problemas mediante el uso de lenguajes de programación y herramientas adecuadas que contribuyan a la creación de soluciones tecnológicas.

Diseña y construye bases de datos bajo los principios de normalización asegurando la integridad de los datos.

6. Temario

No.	Temas	Subtemas
1	Modelado de requisitos.	1.1. Diagramas UML: Clases y objetos. 1.2. Diagramas de secuencia. 1.3. Arquitectura 4+1.
2	Especificación de requisitos de Software	2.1. Proceso de Especificación de Requisitos de Software (ERS). 2.1.1. Entrevistas (historias de usuarios). 2.1.2. Casos de uso y escenarios. 2.1.3. Documentación de Especificación de Requisitos (DER). 2.2. Modelos y estándares. 2.2.1. IEEE 830. 2.2.2. IEEE 1362. 2.2.3. CMMI. 2.2.4. Moprosoft.
3	Análisis y depuración de código	3.1. Flujo de datos. 3.1.1. Análisis de apuntadores. 3.1.2. Análisis basado en restricciones. 3.1.3. Desarrollo guiado por pruebas (TDD). 3.2. Tipo de sistemas. 3.2.1. Errores y excepciones. 3.2.1. Lenguajes tipificados y no tipificados: Fuertes y débiles. 3.3. Depuración de código. 3.3.1. Depuración estadística. 3.3.2. Depuración delta. 3.4. Ejecución simbólica.
4	Metodologías de desarrollo ágil	4.1. Scrum. 4.1.1. Lista de objetivos priorizada (product backlog). 4.2. Programación extrema.



		4.3. Proceso unificado ágil. 4.4. Método de desarrollo de sistemas dinámicos (DSDM).
--	--	---

7. Actividades de aprendizaje de los temas

1. Modelado de Requisitos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Aplicar los conceptos básicos de desarrollo dirigido por modelos, como definir los modelos del sistema, formalizarlos mediante metamodelo y las transformaciones.</p> <p>Utilizar herramientas DSDM para modelar el sistema propuesto. Evaluar diferentes herramientas.</p> <p>Adquirir conocimientos sobre los diferentes tipos de patrones que se van a ir describiendo en la asignatura.</p> <p>Saber si debemos o no aplicar un determinado patrón, en función de las fuerzas del patrón.</p> <p>Genéricas:</p> <p>Habilidad de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.</p> <p>Habilidad de adaptación al ambiente cambiante propio de la disciplina, sabiendo aplicar los conocimientos adquiridos y resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.</p> <p>Capacidad de creatividad.</p> <p>Capacidad para mostrar información de forma clara atendiendo a un amplio conjunto de discapacidades técnicas, físicas y psíquicas.</p>	<p>Investigar en artículos y libros especializados conceptos claves, ejemplos y tendencias de UML.</p> <p>Aplicar la notación UML para representar el modelo de requisitos, casos de uso, documentación de casos de uso y modelo del dominio.</p> <p>Enumerar una lista de ejercicios de análisis de requerimientos.</p> <p>Incorporar el uso de una herramienta CASE para el modelado de requisitos.</p> <p>Seleccionar una problemática real del entorno y dar solución mediante las técnicas aprendidas durante los temas vistos en clase..</p> <p>Realizar foros donde los estudiantes puedan realizar preguntas y sean los compañeros de clase quienes puedan realizar aportaciones de retroalimentación.</p>
2. Especificación de Requisitos de Software	



Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Capacidad de capturar, definir y especificar los requisitos del usuario que permitan establecer de forma trazable el desarrollo, la interfaz de usuarios y pruebas del sistema software</p> <p>Capacidad para detectar y aplicar la aplicación de los patrones de diseño detallado.</p> <p>Capacidad para diseñar e implementar interfaces de usuario siguiendo los requisitos de calidad como usabilidad, seguridad y rendimiento. Conocer e interpretar los estándares, normas y guías de usabilidad y accesibilidad.</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organización y planificación.</p> <p>Capacidad de comunicación oral y escrita de los conocimientos y conclusiones a públicos especializados y no especializados de un modo claro y sin ambigüedades.</p> <p>Capacidad de gestión de la información y de los recursos disponibles.</p> <p>Capacidad de trabajar en equipo con iniciativa y espíritu emprendedor.</p> <p>Habilidades en las relaciones interpersonales.</p> <p>Capacidad de razonamiento y extracción de conclusiones.</p> <p>Compromiso ético y respeto por la propiedad intelectual.</p>	<p>Identificar las características del Proceso de Especificación de Requisitos de Software (ERS).</p> <p>Registrar las especificaciones de requerimientos de los estándares IEEE 830 y 1362.</p> <p>Enumerar un conjunto de ejercicios que den solución a problemas de la vida real donde se registre la recolección de requisitos de usuario, cliente, necesidades del usuario, así como los requisitos que debe cumplir el sistema, realizar entrevistas y de solución a un problema de la vida real.</p> <p>Realizar presentación individual o en equipo de las especificaciones de requisitos de software que el alumno realice.</p> <p>Utilizar una herramienta gráfica para construir los modelos gráficos correspondientes en el proyecto de clase.</p> <p>Realizar visitas de alguna(s) empresas dedicadas al área de desarrollo de software.</p>
3. Análisis y depuración de código	
Competencias	Actividades de aprendizaje



<p>Específica(s):</p> <p>Desarrollar un sistema, en el cual tendrán que ir realizando aplicando los conocimientos adquiridos.</p> <p>Adquirir conocimientos sobre los diferentes tipos de desarrollo actuales y saber cual aplicar a un sistema.</p> <p>Aplicar los conceptos básicos de desarrollo dirigido por modelos, como definir los modelos del sistema, formalizarlos mediante metamodelo y las transformaciones.</p> <p>Utilizar herramientas DSDM para modelar el sistema propuesto. Evaluar diferentes herramientas.</p> <p>Adquirir conocimientos sobre los diferentes tipos de patrones que se van a ir describiendo en la asignatura</p> <p>Genéricas:</p> <p>Capacidad de trabajar en equipo con iniciativa y espíritu emprendedor.</p> <p>Habilidades en las relaciones interpersonales.</p> <p>Capacidad de razonamiento y extracción de conclusiones. Compromiso ético y respeto por la propiedad intelectual.</p> <p>Habilidad de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.</p> <p>Habilidad de adaptación al ambiente cambiante propio de la disciplina, sabiendo aplicar los conocimientos adquiridos y resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinarios) relacionados</p>	<p>Investigar los temas y conceptos claves sobre el Flujo de datos.</p> <p>Realizar una investigación de las diferentes herramientas de depuración, análisis, perfilado y prueba para analizar, depurar y optimizar el código.</p> <p>Registrar las características de la depuración estadística, delta involucradas en la depuración de código.</p> <p>Realiza ejercicios prácticos de análisis y depuración de código.</p> <p>Realiza ejecución simbólica en un caso específico.</p>
--	--



con su área de estudio.	
Nombre de 4. Metodologías de desarrollo ágil	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Realizar un borrador correcto de la interfaz de usuario de una aplicación con un cliente rico.</p> <p>Diseñar un plan de pruebas adecuado para una determinada aplicación.</p> <p>Familiarizarse con el uso de un entorno automático para desarrollar las pruebas.</p> <p>Gestionar un proceso de pruebas para una aplicación.</p> <p>Dada una especificación, decidir qué diagramas son los más adecuados para reflejar la arquitectura del sistema, en función de las necesidades del desarrollo, haciendo un uso efectivo del uso de encapsulación, herencia e interfaces.</p> <p>Genéricas:</p> <p>Capacidad de análisis y síntesis.</p> <p>Capacidad de organización y planificación.</p> <p>Capacidad de comunicación oral y escrita de los conocimientos y conclusiones a públicos especializados y no especializados de un modo claro y sin ambigüedades.</p> <p>Capacidad de gestión de la información y de los recursos disponibles.</p> <p>Capacidad de trabajar en equipo con iniciativa y espíritu emprendedor.</p> <p>Habilidades en las relaciones interpersonales.</p> <p>Capacidad de razonamiento y extracción de conclusiones.</p>	<p>Investigar los componentes y el proceso de la metodología de desarrollo ágil..</p> <p>Investigar los componentes y el proceso del Proceso Unificado Ágil (PUA).</p> <p>Buscar, seleccionar y comparar información sobre los valores de programación extrema (XP), su filosofía de trabajo y sus prácticas.</p> <p>Realizar un debate en clase los aspectos referentes al ciclo de vida de XP.</p> <p>Buscar, seleccionar y comentar información sobre el manifiesto ágil, su aceptación o inconvenientes que tiene.</p> <p>Realizar una exposición de una metodología ágil por equipo y en grupo identificar sus semejanzas y diferencias.</p> <p>Elaborar un cuadro comparativo contrastando todas las metodologías ágiles: Scrum, XP, PUA, DSDM.</p> <p>Plantear preguntas respecto a los diferentes roles de XP, buscar y analizar las respuestas.</p> <p>Plantear un conjunto de historias de usuarios por uno, dos o tres equipo con base en un problema real.</p> <p>Elaborar las pruebas de aceptación</p> <p>Estimar el peso de cada historia de usuario identificada</p> <p>Planear las iteraciones para el proyecto.</p> <p>En clase llevar a cabo una sesión de diseño rápido</p> <p>Desarrollar la programación aplicando las</p>



<p>Compromiso ético y respeto por la propiedad intelectual.</p> <p>Habilidad de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.</p> <p>Habilidad de adaptación al ambiente cambiante propio de la disciplina, sabiendo aplicar los conocimientos adquiridos y resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios relacionados con su área de estudio.</p> <p>Capacidad de creatividad.</p>	<p>prácticas de XP.</p> <p>Buscar información sobre el manejo de defectos en XP</p>
--	---

8. Práctica(s)

1. Realizar el estudio y documentación de la etapa de planeación y análisis de requerimientos para un caso de estudio.
2. Examinar y practicar los aspectos detallados de los diferentes tipos de modelos.
3. Participación en dinámicas de (Foros de discusión, paneles, etc)
4. Elaborar un proyecto integrador que dé solución a un problema dentro de la institución e implemente la metodología aprendida durante la asignatura.
5. Trabajar en equipos de forma colaborativa utilizando metodologías de desarrollo ágiles.

9. Proyecto de asignatura

El objetivo del proyecto que plantee el docente que imparta esta asignatura, 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-profesional, 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 debe ser permanente y continua, se debe hacer una evaluación diagnóstica, formativa y sumativa. Se debe aplicar la autoevaluación, coevaluación y heteroevaluación.

- Realizar mapas conceptuales o mentales para que el estudiante visualice los conceptos vistos en clase.
- Realizar ejercicios individual y grupal de casos de la vida real.
- Reportes escritos de las observaciones realizadas durante las actividades propuestas por el docente, así como avances del proyecto integrador.
- Portafolio de evidencias digital de las prácticas realizadas durante clase.
- Realizar investigaciones escritas y plasmar en documentos digitales y/o escritos.
- Descripción de otras experiencias que podrían realizarse adicionalmente.
- Exámenes escritos para comprobar el manejo de aspectos teóricos.
- Realización de un proyecto integrador donde se plasmen los conocimientos adquiridos durante la asignatura y su relación con otras.

11. Fuentes de información

1. Software Requirements Third Edition, Karl Wieggers and Joy Beatty, Microsoft Press; Edición: 3 (15 de agosto de 2013), ISBN: 978-0-7356-7966-5.
2. More About Software Requirements, Karl E. Wieggers, Microsoft Press; Edición: 1 (15 de noviembre de 2010), ISBN: 0735622671.
3. Modelado de software para profesionales, Fontela Carlos, Editorial Alfaomega, 2010, ISBN, 978-987. 1609-22-2, ISBN- ebook 9786077073062.
4. Code complete (second edition), Steve McConnell, Microsoft Press, 2004, ISBN 0-7356-1967-0.
- 5.
6. The art of scrum, Dave McKenna, a press, 2016, ISBN-13 (pbk): 978-1-4842-2276-8 ISBN-13 (electronic): 978-1-4842-2277-5.
- 7.
8. The Scrum Culture, Dominik Maximini, springer, 2015, ISSN 2192-8096 ISSN 2192-810X (electronic) ISBN 978-3-319-11826-0 ISBN 978-3-319-11827-7 (eBook)
- 9.
10. Planning Extreme Programming, Kent Beck, Martin Fowler, Addison Wesley, ISBN: 0201710919
- 11.
12. Agile Modeling with UML, Bernhard Rumpe, Springer, ISBN: 3319588613
13. Agile and iterative development: a manager's guide Autor(es): LARMAN, Craig : Boston : Pearson Education, 2003. ISBN: 0-13-111155-8
- 14.
15. Star Showcase '11: Exploring the Goals of your Systems and Businesses Autor(es): YU, Eric Yu (coord.) et al. Edición: London : City University London, 2011. Notas: Acceso electrónico: http://www.cs.utoronto.ca/km/istar/iStarShowcase_Proceedings.pdf [Consultado: 31



octubre 2014]