



SÍLABO 2017-

1

ESCUELA UNIVERSITARIA DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

ASIGNATURA	INGENIERÍA DE SOFTWARE I
ÁREA	INGENIERÍA DE SOFTWARE
CÓDIGO	
NIVEL	SÉPTIMO
CARÁCTER	OBLIGATORIO
REQUISITO(S)	INGENIERÍA DE DATOS LENGUAJES DE PROGRAMACIÓN
CRÉDITOS	TRES (3)
HORAS DE TEORÍA	UNO (1)
HORAS DE PRÁCTICA	CUATRO (4)
PROFESOR(ES)	

I. SUMILLA

En la asignatura se estudian las primeras fases del ciclo de vida de desarrollo de software: identificación de requisitos y análisis, con un amplio nivel de detalle y formalidad. Los temas tratados son: metodologías de desarrollo de software; herramientas para documentar formalmente los requisitos; documentación del análisis, así como la trazabilidad entre los requisitos del software y las necesidades de la organización; técnicas básicas de aseguramiento de calidad aplicadas a las fases de requisitos y análisis; plan de implementación de un proyecto.

II. OBJETIVO GENERAL

Al finalizar la asignatura, el alumno será capaz de comprender la naturaleza e importancia de la especificación de requisitos funcionales y no funcionales, así como las actividades correspondientes al análisis de requisitos, aplicando diversos métodos, herramientas y estrategias que se aprenden a lo largo del desarrollo del curso. Asimismo, utilizando diversas técnicas y herramientas, el alumno será capaz de elaborar modelos que evidencian los resultados de las fases de captura de requisitos y análisis dentro del ciclo de desarrollo de software.

III. OBJETIVOS ESPECÍFICOS

1. Los alumnos entienden la naturaleza e importancia de las fases de especificación de requisitos y análisis en el desarrollo de software, comprendiendo la importancia de los diversos interesados (*stakeholders*) en el desarrollo de *software*.
2. Los alumnos se capacitan en el uso de herramientas para la especificación de requisitos y análisis de sistemas de software.

3. Los alumnos elaboran prototipos efectivos aplicando criterios de facilidad de uso.
4. Los alumnos aprenden una metodología para la especificación de requisitos y el análisis orientado a objetos.
5. Los alumnos aprenden un método ágil para la especificación y el análisis de requisitos.
6. Los alumnos adquieren la capacidad de preparar modelos para especificar los requisitos y análisis de un sistema en el desempeño profesional como ingeniero de sistemas utilizando las más modernas herramientas de modelamiento que ofrece la tecnología actual.

IV. PROGRAMA ANALÍTICO

PRIMERA SEMANA

El Proceso de desarrollo de software. Métodos de desarrollo de software. Estándares de ciclo de vida de software, Requisitos de Software.

La Ingeniería de Software. Proceso de desarrollo de software. Métodos tradicionales. Paradigmas de la orientación a objetos. Los conceptos. Métodos orientados a objetos. Análisis de requisitos de software y análisis de sistemas de software. UML. Planificación de proyectos.

SEGUNDA SEMANA

Métodos Prescriptivos. Método orientado a objetos. Proceso unificado. RUP.

Presentación de herramienta para modelamiento de requisitos. Técnicas para captura de requisitos y análisis de sistemas de software.

TERCERA SEMANA

Análisis de Requisitos Orientado a Objetos: Modelo del Negocio Actual y Nuevo Negocio.

CUARTA SEMANA

Análisis de requisitos Orientado a Objetos: Requisitos del Sistema de Software. Especificación de Requisitos de Software (SRS). Estándares de requisitos de software. Características de un buen SRS.

QUINTA SEMANA

Análisis de requisitos Orientado a Objetos: Herramientas de prototipado.

SEXTA SEMANA

Análisis de requisitos Orientado a Objetos: Arquitectura del Modelo de Análisis. Introducción.

SÉPTIMA SEMANA

Análisis Orientado a Objetos: Arquitectura del Modelo de Análisis. Trazabilidad. Verificación y Validación de Requisitos.

OCTAVA SEMANA

Asesoría y apoyo académico a los alumnos
Evaluación del proceso de aprendizaje

NOVENA SEMANA

Análisis Orientado a Objetos: Análisis de Casos de Uso y Análisis de Paquetes. Diagrama de Arquitectura.

DÉCIMA SEMANA

Análisis Orientado a Objetos: Análisis de Clases. Diagrama de clases y colaboración (comunicación). Modelo conceptual. Plan de verificación. Plan de validación.

DÉCIMO PRIMERA SEMANA

Métodos Ágiles: Los conceptos. Captura de Requisitos Ágil. Técnicas y Artefactos. Planificación y técnicas de estimación: puntos de historia de usuario, horas ideales de historia de usuario, Delphi, planning poker. Taller.

DÉCIMO SEGUNDA SEMANA

Métodos Ágiles – Iteraciones: Técnicas y Artefactos. Sprint Backlog, Sprint Review, Daily Meeting, Sprint retrospective y mejora continua del proceso de ingeniería de software (técnicas).

DÉCIMO TERCERA SEMANA

Métodos Ágiles para Requerimientos: Captura de Requisitos Ágil. BDD (Behavior Driven Design). Historias de usuario SMART. Formato Connextra.

DÉCIMO CUARTA SEMANA

Métodos Ágiles para Requerimientos – BDD: Lenguaje Gherkin. Especificación de historias de usuario utilizando lenguaje Gherkin.

DÉCIMO QUINTA SEMANA

Técnicas de estimación basadas en métodos orientados a objetos y técnicas ágiles. Taller integrador análisis de sistemas de software.

DÉCIMO SEXTA SEMANA

Asesoría y apoyo académico a los alumnos
Evaluación del proceso de aprendizaje

V. METODOLOGÍA

La asignatura se desarrollará en un marco teórico - práctico, para lo cual el alumno hará un trabajo de campo, resolverá casuísticas específicas mediante prácticas y controles de lectura en clase e irá reportando el avance del trabajo de campo durante el ciclo; todo ello con la finalidad de dar una mayor visión al estudiante de la construcción de un producto de software en sus primeras fases del ciclo de vida.

VI. SISTEMA DE EVALUACIÓN

El Sistema de Evaluación, contempla la siguiente ponderación de notas (Art. 45° del Reglamento General de Estudios):

- | | | |
|-------------------|---|---------|
| • Examen Parcial | : | peso 3. |
| • Examen Final | : | peso 4. |
| • Tarea Académica | : | peso 3. |

El examen final incluye el contenido total de la materia desarrollada a lo largo del período académico (Art. 46° del Reglamento General de Estudios).
Los contenidos de las prácticas y examen parcial no son cancelatorios.

El alumno que al final del periodo académico tuviera más **del 32% de inasistencias sobre el total de horas de clase programadas**, estará impedido de rendir el examen final. (Art. 41° del Reglamento General de Estudios).

VII. BIBLIOGRAFÍA

OBLIGATORIA

1. Cohn, M. (2010.). *Succeeding with agile: software development using Scrum*. N.J.: Addison-Wesley. Código Ulima: 005.1C684.
2. Fontela, C. (2011). *UML: Modelado de software para profesionales*. Alfaomega. Código Ulima: 005.117 F75.
3. Fox, A., Patterson, D., Joseph, S., & M., C. G. (2015). *Desarrollando software como servicio (SaaS): un enfoque ágil utilizando computación en la nube*. San Francisco, CA: Strawberry Canyon LLC.
4. Sommerville, I. (2011). *Ingeniería de Software*. México, D.F.: Addison Wesley. Código Ulima: 005.1 S67 2011.

COMPLEMENTARIA

5. Álvarez, G. A., Heras, D. R., & Lasa, G. C. (2011). *Manual imprescindible de métodos Ágiles y Scrum*. Madrid: Anaya Multimedia. Código Ulima: 005.1 A45.
6. Beaver, J. (2014). *The Agile team handbook*. Código Ulima: 005.1 B29.
7. Genero, M., Cruz-Lemus, J., & Piattini, M. (2014). *Métodos de investigación en Ingeniería de Software*. Madrid : RA-MA. Código Ulima: 005.1 G37.
8. Jones, C. (2010). *Software engineering best practices : lessons from successful projects in the top companies*. New York: Mc Graw Hill. Código Ulima:005.1 J72.
9. Rubin, K. S. (2012). *Essential Scrum: A practical guide to the most popular agile process*. Upper Saddle River, NJ: Addison-Wesley. Código Ulima: 005.117 B73L.
10. Rumbaugh, J., Booch, G., & Jacobson, I. (2006). *El lenguaje unificado de modelado: guía del usuario*. Madrid: Addison -Wesley. Código Ulima: 005.R91.
11. Viscardi, S. (2013). *The professional ScrumMaster's handbook: A collection of tips, tricks, and war stories to help the professional ScrumMaster break the chains of traditional organization and management*. Birmingham, U.K: Packt Publishing. Código Ulima: 005.1 V75.
12. Woodward, E. (2010). *A practical guide to distributed Scrum*. N.J.: Upper Saddle Rive. Código Ulima: 005.1 W79.

Estándares:

1. IEEE. IEEE-STD-830-1998. (1998). Especificaciones de los Requisitos del Software.
2. ISO/IEC/IEEE 29148-2011. (2011). Systems and software engineering — Life cycle processes — Requirements engineering.