



Análisis y Diseño I

4 créditos teóricos



A. Información del profesor

Nombre del profesor

Inga. Hilda Ruth Flores Muñoz

Correo electrónico

hrflores@correo.url.edu.gt

Campus o sede

Central

Horario

Lunes y viernes de 19:30 a 21:00 horas.



B. Información general

Descripción

Para la formación del ingeniero de software de nuestros días, la recomendación curricular de ACM e IEEE enfatiza la importancia de las áreas de conocimiento de Modelación y Análisis de Software (MAA) y Diseño de Software (DES). Este es el primero de dos cursos de especialización en la disciplina de Ingeniería de Software y se asume que el estudiante ya cuenta con conocimientos sólidos de programación orientada a objetos y bases de datos. El primer curso se enfoca en las técnicas básicas de análisis y diseño de software, mientras que el segundo profundiza en las actividades de diseño de componentes, de arquitecturas de software y de la interfaz de usuario

Modalidad

Presencial.



C. Malla curricular

COMPETENCIAS GENÉRICAS



El egresado landivariano se identifica por:

Pensamiento
lógico, reflexivo y
analógico

Pensamiento
crítico

Resolución de
problemas

Habilidades de
investigación

Uso de TIC y
gestión de la
información

Comunicación
efectiva, escrita y
oral

Comprensión
lectora

Compromiso
ético y
ciudadanía

Liderazgo
constructivo

Aprecio y respeto
por la diversidad e
interculturalidad

Creatividad

COMPETENCIAS ESPECÍFICAS (propias del curso)

Competencia 1

Contrasta la Ingeniería de Software con la Ingeniería de Sistemas e interpreta los modelos existentes para el desarrollo de software de calidad.

Competencia 2

Analiza, especifica y valida los requerimientos que representan las necesidades que la solución de software debe atender, de manera que se garantice la objetividad y la factibilidad.

Competencia 3

Diseña los modelos que simbolizan la estructura y funcionamiento de la solución de software planteada en concordancia con los requerimientos funcionales y no funcionales identificados.



METODOLOGÍA

Este curso se desarrollará a través de los siguientes métodos de aprendizaje-enseñanza: Aprendizaje invertido, aprendizaje basado en retos, aprendizaje por indagación, aprendizaje basado en equipos.



PROGRAMACIÓN

COMPETENCIA 1

Contrasta la Ingeniería de Software con la Ingeniería de Sistemas e interpreta los modelos existentes para el desarrollo de software de calidad.

Saber conceptual (contenido temático)

Ingeniería de software, arquitectura de software, ciclo de vida de desarrollo de software. Ciclo de vida clásico, modelos evolutivos (prototipos, cascada, espiral y componentes), modelos ágiles (XP, SCRUM), modelos orientados a arquitectura (Zachman Framework, Model driven) y modelos de referencia (RUP, EUP, AUP, CMM).

Saber procedimental (habilidades y destrezas)

Resolución de problemas con software.
Aplicación del modelo idóneo a un problema que se resuelve con software.

Saber actitudinal (conductas observables)

Lectura comprensiva y preparación antes de la clase.
Participación activa.
Trabajo colaborativo.
Desarrollo de competencias del perfil de egreso.
Integridad, puntualidad y compromiso con la calidad.

Indicador de logro 1 (resultado): Contrasta la Ingeniería de Software con la Ingeniería de Sistemas e interpreta los modelos existentes para el desarrollo de software de calidad.

COMPETENCIA 2

Analiza, especifica y valida los requerimientos que representan las necesidades que la solución de software debe atender, de manera que se garantice la objetividad y la factibilidad.

Saber conceptual (contenido temático)

Negocio, arquitectura del negocio, vistas del negocio, especificación de procesos de negocios.
Application Lifecycle Management (ALM).
Flujos de trabajo y técnicas de recolección de datos.
Actividades roles y artefactos (documentos de visión, especificación de casos de uso,



especificaciones suplementarias).

Actividades de la determinación de requerimientos (espacio del problema, espacio de la solución, alcance, administración del cambio, utilización del modelo del negocio, definición del dominio, organización, procesos).

Modelación de requerimientos utilizando casos de uso (descripción, flujo básico y alternos, escenarios, relaciones uses y extend, procesos para escribir casos de uso).

- Atributos de calidad.
- Diagramas de actividad.
- Historias de usuarios.
- Documento de especificación de requerimientos.

Saber procedimental (habilidades y destrezas)

Análisis y especificación de requerimientos de negocios utilizando modelación visual.

Aplicación de diferentes técnicas de recolección de datos para la identificación de requerimientos de una solución.

Especificación y/o modelado de requerimientos de software y hardware basados en los requerimientos de negocio.

Diseño de diagramas de casos de uso, descripción de casos y de uso y diagramas de actividades.

Generación de una primera versión de documento de especificación de requerimientos y modelado funcional de la solución.

Saber actitudinal (conductas observables)

Lectura comprensiva y preparación antes de la clase.

Participación activa.

Trabajo colaborativo.

Desarrollo de competencias del perfil de egreso.

Integridad, puntualidad y compromiso con la calidad.

Indicador de logro 1 (resultado): Analiza, especifica y valida los requerimientos que representan las necesidades que la solución de software debe atender, de manera que se garantice la objetividad y la factibilidad.

COMPETENCIA 3

Diseña los modelos que simbolizan la estructura y funcionamiento de la solución de software planteada en concordancia con los requerimientos funcionales y no funcionales identificados.

Saber conceptual (contenido temático)

Estudio de modelación visual utilizando UML para la vista estructural y funcional de un sistema.

- Diagrama estático de clases y objetos.
- Máquinas de estado.
- Diagramas de comunicación.
- Diagramas de secuencias.



Principio del diseño de la interfaz del usuario

El proceso del diseño de la interfaz del usuario (Diseño de navegación, Diseño de ingresos, Diseño de salidas).

Computación móvil y el diseño de la interfaz del usuario (Diseño de la interfaz de la media social, Diseño de la interfaz del usuario en el ámbito internacional y cultural, Requerimientos no funcionales y el diseño de la capa de interacción Humano-computador).

Validación y verificación del análisis y diseño de las soluciones de software.

Saber procedimental (habilidades y destrezas)

Generación de diagramas que muestran las vistas estructurales y funcionales de una futura solución tecnológica.

Diagramas de navegación de un sistema.

Especificación de entradas y salidas de las interfaces de usuario de un sistema.

Herramientas tecnológicas que apoyan al analista para la creación de mockups o diseños avanzados de las interfaces de usuario.

Restricciones o necesidades expresadas en los requerimientos no funcionales y como estos afectan a la interfaz de usuario.

Saber actitudinal (conductas observables)

Lectura comprensiva y preparación antes de la clase.

Participación activa.

Trabajo colaborativo.

Desarrollo de competencias del perfil de egreso.

Integridad, puntualidad y compromiso con la calidad.

Indicador de logro 1 (resultado): Diseña los modelos que simbolizan la estructura y funcionamiento de la solución de software planteada en concordancia con los requerimientos funcionales y no funcionales identificados.



EVALUACIÓN

a. Estrategias de evaluación sumativa

Estrategias	Puntaje
Pruebas parciales	20
Exámenes cortos y otras actividades en clase	20
Proyecto de aplicación	40
Examen final	20
TOTAL	100

b. Estrategias de evaluación formativa

Técnicas formativas	Procedimiento
Retroalimentación	Comentarios pertinentes en la entrega de los laboratorios y proyectos.
Diálogo socrático	Preguntas y respuestas orales a ejemplos y problemas que se realizarán a lo largo de la secuencia de aprendizaje.
<i>One minute paper</i>	Textos cortos sobre los temas ya vistos comprobando sus saberes.
Trabajos en pequeños grupos para resolver dudas	Dinámicas de grupo en clase como 1-2-3-all.
Citas individuales	Tutorías de retroalimentación solicitadas por el estudiante, por medios electrónicos.

CALENDARIO DE REFERENCIA POR TEMAS

Fecha	Tema	Actividad de evaluación
Semana 1 16 y 20 enero	Ingeniería de software, arquitectura de software, ciclo de vida de desarrollo de software. Ciclo de vida clásico.	Aplicación de estrategias según la metodología definida para el curso.
Semana 2 23 y 27 enero	Modelos evolutivos (prototipos, cascada, espiral y componentes), modelos ágiles (XP, SCRUM), modelos orientados a arquitectura (Zachman Framework, Model driven) y modelos de referencia (RUP, EUP, AUP, CMM).	Aplicación de estrategias según la metodología definida para el curso
Semana 3 30 ene y 3 feb	Modelos evolutivos (prototipos, cascada, espiral y componentes), modelos ágiles (XP, SCRUM), modelos orientados a arquitectura (Zachman Framework, Model driven) y modelos de referencia	Taller de modelos ágiles. Aplicación de estrategias según la metodología definida para el curso.



	(RUP, EUP, AUP, CMM).	
Semana 4 6 y 10 feb	Negocio, arquitectura del negocio, vistas del negocio, especificación de procesos de negocios.	Definición de proyecto. Aplicación de estrategias según la metodología definida para el curso.
Semana 5 13 y 17 feb	<i>Application Lifecycle Management (ALM)</i>	Primer parcial. Aplicación de estrategias según la metodología definida para el curso.
Semana 6 20 y 24 feb	Flujos de trabajo y técnicas de recolección de datos. Modelación de requerimientos utilizando casos de uso (descripción, flujo básico y alternos, escenarios, relaciones <i>uses</i> y <i>extend</i> , procesos para escribir casos de uso).	Aplicación de estrategias según la metodología definida para el curso.
Semana 7 27 feb y 3 mar	Actividades roles y artefactos (documentos de visión, especificación de casos de uso, especificaciones suplementarias). Actividades de la determinación de requerimientos (espacio del problema, espacio de la solución, alcance, administración del cambio, utilización del modelo del negocio, definición del dominio, organización, procesos). Atributos de calidad. <ul style="list-style-type: none"> • Diagramas de actividad. • Historias de usuarios. • Documento de especificación de requerimientos. 	Entrega 1 Aplicación de estrategias según la metodología definida para el curso.
Semana 8 6 y 10 mar	Estudio de modelación visual utilizando UML para la vista estructural y funcional de un sistema. <ul style="list-style-type: none"> • Diagrama estático de clases y objetos. 	Entrega 2 Aplicación de estrategias según la metodología definida para el curso.
Semana 9 13 y 17 mar	Estudio de modelación visual utilizando UML para la vista estructural y funcional de un sistema. <ul style="list-style-type: none"> • Máquinas de estado. • Diagramas de comunicación. • Diagramas de secuencias. 	Entrega 3 Aplicación de estrategias según la metodología definida para el curso.
Semana 10 20 y 24 mar	Integración	Segundo parcial. Aplicación de estrategias según la metodología definida para el curso.



Semana 11 27 y 31 mar	Principio del diseño de la interfaz del usuario.	Entrega 4. Aplicación de estrategias según la metodología definida para el curso.
Semana 12 3 y 7 abr	Semana Santa	
Semana 13 10 y 14 abr	El proceso del diseño de la interfaz del usuario (Diseño de navegación, Diseño de ingresos, Diseño de salidas).	Aplicación de estrategias según la metodología definida para el curso.
Semana 14 17 y 21 abr	Computación móvil y el diseño de la interfaz del usuario (Diseño de la interfaz de la media social, Diseño de la interfaz del usuario en el ámbito internacional y cultural, Requerimientos no funcionales y el diseño de la capa de interacción Humano-computador).	Entrega 5. Aplicación de estrategias según la metodología definida para el curso.
Semana 15 24 y 28 abr	Validación y verificación del análisis y diseño de las soluciones de software.	Aplicación de estrategias según la metodología definida para el curso.
Semana 16 1 y 5 may	Examen final	Examen final

REFERENCIAS BIBLIOGRÁFICAS

IEEE (s.f.). Software Engineering Body of Knowledge – SWEBOK. Recuperado de:
<https://www.computer.org/education/bodies-of-knowledge/software-engineering>

IEEE SA (s.f.). Standards. Institute of Electrical and Electronics Engineers, Inc – IEEE. Recuperado de:
<https://standards.ieee.org/>

ISO (s.f.) International Standards. International Organization for Standardization – ISO. Recuperado de: <https://www.iso.org/home.html>

Jacobson, I., Lawson, H., Ng, P.W., McMahon, P.E., Goedicke, M. (2019). The Essentials of Modern Software Engineering. Association for Computing Machinery and Morgan & Claypool Publishers.

Pressman, R.S., Maxim, B.R. (2010) Software Engineering. A practitioner’s Approach. 8th. Edition: McGrawHill Education.

Sommerville, I. (2016). Software Engineering. Pearson Education Limited.



Stephens, R. (2015). Beginning Software Engineering.

Williams, L. (2013). An Introduction to Software Engineering. Edition one.

Winters, T., Manshreck, T., Wright, H. (2020) Software Engineering at Google. Lessons Learned from Programming Over Time. O'Reilly.