

Facultad de Ingeniería

Introducción a la ingeniería en Informática y Sistemas

2 CRÉDITOS



A. Información del profesor

Nombre del profesor

Mgtr. Karen Elizabeth Liska Lima

e-mail

karenliska@gmail.com

Horario:

Lunes - Viernes 19:30-21:00 horas.



B. Información general

Descripción

La Ingeniería de software, es una rama de la ingeniería especializada en la creación de productos de software con la calidad y flexibilidad que el usuario necesita. Abarcando desde su desarrollo hasta su implementación, mantenimiento y mejora continua. En este curso se abordarán y pondrán en práctica los conceptos necesarios para la construcción de software de calidad desde su punto de partida hasta su implementación lo que es vital para un ingeniero en Informática y Sistemas, sin importar el área de especialización.

Modalidad

Presencial.



Facultad de Ingeniería



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	v respeto	

Apreció y respeto por la diversidad e interculturalidad

Creatividad

COMPETENCIAS ESPECÍFICAS (propias del curso)

Competencia 1: Identifica los aspectos que vuelven al software una disciplina de la ingeniería.

Competencia 2: Reconoce la importancia del control de calidad junto a sus diferentes técnicas como parte de esta rama de la ingeniería.

Competencia 3: Distingue las diferentes etapas del ciclo de vida de software y su correcta aplicación.

Competencia 4: Pone en práctica la administración de recursos en un proyecto de software.







Facultad de Ingeniería



METODOLOGÍA

Este curso se desarrollará a través de los siguientes métodos de aprendizaje-enseñanza:

Aprendizaje invertido

«La exposición de saberes se realiza por medio de documentos, videos y otros materiales por parte del estudiante. El tiempo de sesión síncrona¹ se dedica a la discusión, resolución de problemas y actividades prácticas bajo la supervisión del profesor».

Aprendizaje basado en problemas (ABP)



«Metodología centrada en el aprendizaje, en la investigación y reflexión que siguen los alumnos para llegar a una solución ante un problema planteado. Desarrolla aprendizajes activos a través de la resolución de problemas y casos. Puede desarrollarse de manera sincrónica o asíncrona»







Facultad de Ingeniería

PROGRAMACIÓN

COMPETENCIA 1

Identifica los aspectos que vuelven al software una disciplina de la ingeniería.

Saber conceptual (contenido temático)

- 1.1 Introducción del software como modelo de negocio
- 1.2 Introducción del software como complemento de otros giros de negocio
- 1.3 Software como medio para cumplir los objetivos empresariales

Saber procedimental (habilidades y destrezas)

- Analiza los problemas de negocio.
- Aplica soluciones de software ante los problemas de negocio.
- Utiliza nociones del ciclo de vida del software alineadas a cumplir los objetivos del negocio.

Saber actitudinal (conductas observables)

Resuelve problemas tomando en cuenta la Ingeniería de Software.

Indicadores de logro 1 (resultado):

Reconoce el software como una disciplina del campo ingenieril.

COMPETENCIA 2

Reconoce la importancia del control de calidad junto a sus diferentes técnicas como parte de esta rama de la ingeniería.

Saber conceptual (contenido temático)

- 2.1 La importancia del control de calidad según el NIST(National Institute of Standards and Technology)
- 2.2 Planificación de pruebas
- 2.3 Pruebas unitarias
- 2.4 Pruebas de funcionalidad
 - 2.4.1 Pruebas Caja Blanca
 - 2.4.2 Pruebas Caja negra
 - 2.4.3 Pruebas Caja Gris
- 2.5 Pruebas no funcionales
 - 2.5.1 Pruebas de carga
 - 2.5.2 Pruebas de Rendimiento
 - 2.5.3 Pruebas de UX (*User Experience*)
 - 2.5.4 Análisis de seguridad de software
 - 2.5.4.1 Análisis estático de código
 - 2.5.4.2 Análisis dinámico de código
 - 2.5.4.3 Otras Auditorías de Seguridad
- 2.6 Pruebas de Integración
- 2.7 Pruebas de Regresión
- 2.8 Pruebas de Aceptación del cliente
- 2.9 Automatización de pruebas





Facultad de Ingeniería

Saber procedimental (habilidades y destrezas)

- Distingue la importancia de las diferentes pruebas sobre el software.
- Comprende la aplicación de cada tipo de prueba en diferentes escenarios.

Saber actitudinal (conductas observables)

• Aplica diferentes tipos de pruebas sobre el software.

Indicador de logro 2 (resultado):

Distingue y aplica los distintos tipos de pruebas reconociendo sus objetivos.



Facultad de Ingeniería

COMPETENCIA 3

Distingue las diferentes etapas del ciclo de vida de software y su correcta aplicación.

Saber conceptual (contenido temático)

- 3.1 Administración de requerimientos funcionales y no funcionales
- 3.2 Técnicas de estimación de tiempos
- 3.3 Ambientes en un entorno de desarrollo
- 3.4 Control de versiones
- 3.5 Evolución del software
- 3.6 Monitoreo y Control del software
- 3.7 Manejo de Incidentes, Problemas y Bugs

Error! Bookmark not defined.

Saber procedimental (habilidades y destrezas)

- Estudia cómo hacer una planificación de desarrollo de software.
- Analiza la evolución de un sistema de software en el tiempo.
- Distingue los riesgos que pueden darse en la puesta en producción del software.
- Reconoce las diferentes formas de llevar un control de versiones.
- Identifica los procesos de monitoreo del software.

Saber actitudinal (conductas observables)

- Discierne sobre la importancia de los diferentes pasos del ciclo de vida del desarrollo de software.
- Argumenta las bases para la aplicación de las tecnologías emergentes para la propuesta de soluciones de ingeniería.

Indicador de logro 3 (resultado):

Aplica correctamente el ciclo de vida de software.



Facultad de Ingeniería

COMPETENCIA 4

Pone en práctica la administración de recursos en un proyecto de software.

Saber conceptual (contenido temático)

- 4.1. Triángulo de hierro para administrar el alcance en base al tiempo y presupuesto de un proyecto de software.
- 4.2 Tipos de licenciamiento de software
- 4.3 Opciones para la implementación de un proyecto de software.
- 4.4 Legislación en el software.
- 4.5 Perfiles profesionales en la ingeniería de software.
 - 4.5.1 Estructuración de equipos

Saber procedimental (habilidades y destrezas)

- Distingue las prioridades que un software debe cumplir.
- Comprende la visión y objetivos del software como ayuda al negocio.
- Identifica las diferentes restricciones de licencias y legislación sobre el software.

Saber actitudinal (conductas observables)

- Comprende, las limitantes y riesgos que pueden surgir en un proyecto de software.
- Desarrolla el pensamiento crítico, aplicando asertivamente los conceptos sobre creación de perfiles para la estructuración de equipos.

Indicador de logro 4 (resultado):

Aplica los conceptos para la administración básica de un proyecto de software.





a. Estrategias de evaluación sumativa

Estrategias	Puntaje
Evaluación Parcial	30
Evaluaciones Cortas	15
Trabajo en clase	10
Laboratorio	15
Examen final	30
TOTAL	100

b. Estrategias de evaluación formativa

Técnicas formativas	Procedimiento
One minute paper	Textos cortos sobre los temas vistos para comprobar los saberes
Retroalimentación	Comentarios pertinentes en la entrega de laboratorios y proyectos
Foros en portal	Discusión de temas vistos y conclusiones sobre lo abordado.
Quiz (exámenes cortos)	Pruebas de selección múltiple para identificar los conceptos aprendidos.
Frabajos en pequeños grupos para resolver dudas	Experimentación de diversas tecnologías para la formación de criterios profesionales.
Citas individuales	Resolución de dudas y acompañamiento específico por estudiante.



Facultad de Ingeniería



CALENDARIO DE REFERENCIA POR TEMAS

Fecha	Tema	Actividad de evaluación
Semana 1	Introducción a la ingeniería de software,	Trabajo en clase 1
	por qué es importante para el negocio y	
	considerada una disciplina de la	
	ingeniería Parte 1.	
Semana 2	Introducción a la ingeniería de software,	
	por qué es importante para el negocio y	
	considerada una disciplina de la	
	ingeniería Parte 2	
Semana 3	Ciclo de vida de software parte 1	Laboratorio 1
Semana 4	Ciclo de vida de software parte 2	Trabajo en clase 2
Semana 5	Ciclo de vida de software parte 3	Laboratorio 3
Semana 6	Ciclo de vida de software parte 3	Trabajo en clase 3, Evaluación Parcia
		1
Semana 7	Control de calidad parte 1	Trabajo en clase 4
Semana 8	Control de calidad parte 2	Laboratorio 4
Semana 9	Control de calidad parte 3	Laboratorio 5
Semana 10	Control de calidad parte 4	Evaluación Parcial 2
Semana 11	Control de calidad parte 5	Laboratorio 6
Semana 12	Administración de recursos de un	Trabajo en clase 5
	proyecto de software Parte 1	
Semana 13	Administración de recursos de un	Laboratorio 7
	proyecto de software Parte 2	
Semana 14	Administración de recursos de un	
	proyecto de software Parte 3	
Semana 15	Cierre del curso	Laboratorio 8, Evaluación final

REFERENCIAS BIBLIOGRÁFICAS

- Texto: Pressman, Roger S. (2005). Ingeniería del Software. Un enfoque práctico. Mc-Graw Hill. Sexta edición. México.
- Texto: Sommerville. Software Engineering. Addison-Wesley. 8a. Edición. 2007.
- ☐ Texto: Jean-Luc BAUD, ITIL® 4 Entender el enfoque y adoptar las buenas prácticas
- Texto: ISTQB, SYLLABUS "Certified Tester Foundation Level", Version 2018