



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 y viernes 18:00-19:20 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 planificación de proyectos de software tomando en cuenta los estándares de la industria y abordando temas como la implementación de los mismos lo que es vital para un ingeniero en Informática y Sistemas, sin importar el área de especialización.

Modalidad

Presencial.



C. 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: Identifica las diferentes habilidades necesarias para la contratación de personal para un proyecto de software.

Competencia 2: Reconoce la importancia de la entrega continua en los productos de software.

Competencia 3: Distingue las diferentes licencias y contratos de nivel de servicio para la utilización y reutilización del software.

Competencia 4: Pone en práctica las diferentes estrategias de implementación de un proyecto.



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 síncrona o asíncrona»



PROGRAMACIÓN

COMPETENCIA 1

Identifica las diferentes habilidades necesarias para la contratación de personal para un proyecto de software.

Saber conceptual (contenido temático)

- 1.1 Dirección de proyectos de software y liderazgo
- 1.2 Creación de perfiles de contratación
- 1.3 Diferentes roles administrativos y su función

Saber procedimental (habilidades y destrezas)

- Analiza los problemas en el desarrollo de un producto de software.
- Aplica la definición de perfiles para la creación de equipos de trabajo.
- Utiliza diferentes nociones sobre roles de liderazgo para la creación de equipos de desarrollo.

Saber actitudinal (conductas observables)

- Resuelve problemas tomando en cuenta las diferentes perspectivas que pueden existir de los equipos en la ingeniería de software.

Indicadores de logro 1 (resultado):

Reconoce la importancia de la definición de perfiles de contratación para proyectos de software, así como aplica las diferentes habilidades de dirección de equipos.

COMPETENCIA 2

Reconoce la importancia de la entrega continua en los productos de software.

Saber conceptual (contenido temático)

- 2.1 Entrega continua y su importancia
- 2.2 Estrategias de entrega continua
- 2.3 Integración continua

Saber procedimental (habilidades y destrezas)

- Comprende los conceptos de entrega e integración continua.
- Analiza los diferentes componentes para realizar entregas continuas en entornos productivos.

Saber actitudinal (conductas observables)

- Aplica formas de entrega e integración continua.

Indicador de logro 2 (resultado):

Distingue las estrategias para la entrega continua.



COMPETENCIA 3

Distingue las diferentes licencias y contratos de nivel de servicio para la utilización y reutilización del software.

Saber conceptual (contenido temático)

- 3.1 Diferentes licencias para la utilización del software.
- 3.2 Derechos legales de los programadores
- 3.3 Acuerdos de nivel de servicio y otros contratos.

Saber procedimental (habilidades y destrezas)

- Estudia las diferentes licencias del software para su reutilización y desarrollo.
- Analiza la evolución de un sistema de software en el tiempo.
- Distingue los diferentes contratos de software y derechos que éstos implican.

Saber actitudinal (conductas observables)

- Discierne sobre la importancia de evaluar las distintas licencias y derechos de los programadores a la hora de crear un producto de software.

Indicador de logro 3 (resultado):

Reconoce la importancia de conocer los tipos de licenciamiento a la hora de crear proyectos de software ya sea reutilizando componentes, subcontratando o haciendo entrega de un producto del mismo.



COMPETENCIA 4

Pone en práctica las diferentes estrategias de implementación de un proyecto.

Saber conceptual (contenido temático)

- 4.1. ¿Qué es la implementación de un sistema de software?
- 4.2 Diseño de estrategias de implementación de sistemas de software
 - 4.2.1 Capacitaciones
 - 4.2.2 Migraciones
 - 4.2.3 Proceso de instalación
 - 4.2.4 Procesos de infraestructura
- 4.3 Calendarización de implementación

Saber procedimental (habilidades y destrezas)

- Distingue las bases para la implementación de proyectos de sistemas.
- Comprende las diferentes herramientas que pueden servir en la implementación de los sistemas de software.
- Analiza los pasos para crear un plan de implementación dependiendo del caso.

Saber actitudinal (conductas observables)

- Desarrolla estrategias de implementación.

Indicador de logro 4 (resultado):

Aplica diferentes estrategias de implementación de los sistemas.



EVALUACIÓN

a. Estrategias de evaluación sumativa

Estrategias	Puntaje
Evaluación Parcial	30
Evaluaciones Cortas	10
Trabajo en clase	10
Laboratorio	20
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.
Trabajos 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.



CALENDARIO DE REFERENCIA POR TEMAS

Fecha	Tema	Actividad de evaluación
Semana 1	Repaso sobre administración de proyectos de software y estimación de tiempos.	Evaluación corta 1
Semana 2	Creación de perfiles para un proyecto de software.	Trabajo en clase 1
Semana 3	Roles administrativos y sus funciones.	Laboratorio 1, Evaluación corta 2
Semana 4	Liderazgo y distribución de trabajo de un proyecto de software.	Trabajo en Clase 2
Semana 5	Licenciamiento de software.	Laboratorio 2
Semana 6	Definición de contratos para el desarrollo de software.	Trabajo en clase 3
Semana 7	Derechos de los desarrolladores.	Trabajo en clase 4, Evaluación corta 3
Semana 8	Diferentes contratos en proyectos de software	Laboratorio 3 Evaluación parcial 1
Semana 9	Conceptos de integración y entrega continua	
Semana 10	Herramientas de Integración continua	Laboratorio 4
Semana 11	Asueto Semana Santa	
Semana 12	Herramientas para la entrega continua	Evaluación corta 4
Semana 13	Continuación de entrega e integración continua	Laboratorio 5
Semana 14	Tareas de integración y entregas	Trabajo en clase 5, Evaluación parcial 2
Semana 15	¿Qué es la implementación?	Trabajo en clase 6
Semana 16	Pasos para un plan de implementación	Laboratorio 6
Semana 17	Capacitaciones	Evaluación corta 5, trabajo en clase 7
Semana 18	Migraciones	Laboratorio 7
Semana 19	Diseño de un plan de implementación	Trabajo en clase 8
Semana 20	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