

PROGRAMA DE INGENIERÍA DE SOFTWARE

Facultad:	Ingeniería	Departamento:	Gestión de Proyectos y Sistemas
Código:	FPTSP07	Asignatura:	Ingeniería de Software
Creditos:	3	Tipo:	<input checked="" type="checkbox"/> Obligatoria <input type="checkbox"/> Electiva
Carreras:	Ingeniería de Sistemas (IS), Matemáticas Industriales (MI)	Trimestres:	VIII (IS)
Prerrequisito	Sistemas de Información (FPTSP04) Bases de Datos I (FPTSP01)	Modalidad:	Presencial
Número de horas semanales			
En aula	Prácticas supervisadas	Laboratorio	Aprendizaje Autónomo
4			4
Coordinador:	Christian Guillén Drija	Fecha de actualización	Oct. 2025

1. **Justificación:** La asignatura "Ingeniería de Software" es fundamental para el perfil profesional de la carrera de Ingeniería de Sistemas, ya que consolida las habilidades del estudiante en el desarrollo integral de sistemas de información. Esta materia cierra el ciclo formativo dedicado a la creación y gestión de aplicaciones, después de cursar asignaturas como Sistemas de Información y Bases de Datos. Su enfoque principal está en promover buenas prácticas de desarrollo, mejorar la productividad y garantizar la calidad del producto software, aspectos imprescindibles para responder a los desafíos del mercado y las necesidades tecnológicas actuales.

2. **Propósito:** Este curso tiene como propósito principal capacitar al estudiante en la adquisición y aplicación de métodos, técnicas y herramientas propias de la ingeniería de software que permitan mejorar la calidad, mantenibilidad y vida útil del producto desarrollado. Se busca que el estudiante comprenda el impacto de las metodologías de desarrollo, el diseño y la arquitectura del software como pilares fundamentales para la calidad del mismo. Además, el curso fomenta un aprendizaje activo y colaborativo entre alumnos y profesores, utilizando metodologías ágiles y basadas en proyectos reales, que preparan al estudiante para enfrentar eficazmente los retos profesionales en el desarrollo de software.

3. Objetivos

- Explicar los principios fundamentales de la ingeniería de software, incluyendo las metodologías de desarrollo (ágiles y tradicionales), los modelos de ciclo de vida y las métricas de calidad, y aplicarlos en el desarrollo de un proyecto de software. (Objetivos 1 y 2 de IS; objetivo 5 de MI)
- Analizar un caso de estudio de un proyecto de software fallido, identificando las causas del fracaso, relacionadas con la gestión del proyecto, la calidad del software o el incumplimiento de los requisitos, y evaluar críticamente las decisiones tomadas durante el desarrollo. (Objetivos 1, 2 y 4 de IS; objetivo 1 de MI)
- Desarrollar un sistema de software funcional, utilizando herramientas y metodologías de desarrollo de software, que cumpla con los requisitos especificados por el usuario y que incorpore buenas prácticas de ingeniería de software. (Objetivos 1, 2 y 3 de IS; objetivo 5 de MI)
- Aplicar diferentes técnicas de prueba de software para la validación y el marco de calidad para evaluar la calidad de un sistema de software, incluyendo pruebas unitarias, de integración y de sistema, y generar un informe con los resultados de las pruebas y recomendaciones para la mejora del software (Objetivos 1, 2 y 4 de IS; objetivo 6 de MI)

4. Resultados de aprendizaje

- a) **RA1 - Gestión y trabajo en equipo.** Capacidad para gestionar y trabajar de manera eficaz individualmente y en equipo, asumiendo roles de liderazgo, creando un entorno colaborativo e inclusivo, estableciendo metas, planificando tareas y cumpliendo objetivos. (N3)
- b) **RA11 - Práctica de la ingeniería.** Capacidad para desarrollar, implementar, gestionar aplicaciones, productos informáticos y sistemas de información, usando metodologías de desarrollo, técnicas y herramientas computacionales apropiadas, asegurando calidad, seguridad y fiabilidad del producto final. (N3)

5. Contenido

Tema	Contenido	Herramientas técnicas y actividades (proyectos, trabajos, laboratorios)	Horas dedicadas
1	Propuesta de Proyecto: Modelos CANVAS. Propuesta de Valor.	Clase invertida: resolución de dudas y casos de estudio. Proyecto en equipo: desarrollo de CANVAS. Gamificación para reforzar conceptos de metodologías.	4

2	Modelado y Programación Orientada a Objetos: Modelado con UML. Metodologías de desarrollo, ágiles o de cuarta generación.	Dinámicas de clase basadas en actividades previas (clase invertida). Aplicación de estándares UML en entornos de desarrollo.	10
3	Desarrollo de Software Orientado a Objetos: Modelos de: requisitos, de análisis, de diseño, de implementación. Patrones. Garantía de Calidad. Modelos de pruebas. Control de cambio y Mantenimiento	Proyecto colaborativo: ciclo de vida orientado a objetos, aplicación de patrones, pruebas de software y control de cambios.	14
4	Ingeniería del Software: Crisis del software. Calidad y cualidades del Software. Métricas y técnicas de estimación. Planificación y control de proyecto.	Role-Playing: simulación de roles Scrum (PO, SM, Team). Debates sobre métricas y gestión de riesgos. Diseño de marcos de calidad.	14
5	Ingeniería de Información: Planificación estratégica de sistemas. Administración de datos. Sistemas estratégicos. Paradigmas tecnológicos.	Estudio de casos: sistemas estratégicos y ventaja competitiva. Presentación final del proyecto ante jurado de especialistas.	6

6. **Métodos de aprendizaje:** Se emplea una metodología de aprendizaje activo y experiencial centrada en el desarrollo de un proyecto real bajo el marco Scrum. El curso combina clases magistrales con talleres prácticos, dinámicas ágiles y simulaciones de roles profesionales. Se fomenta el aprendizaje colaborativo mediante el trabajo en equipos autogestionados y el uso de herramientas de vanguardia como GitHub para el control de versiones. Este enfoque se complementa con aprendizaje autónomo y tutorías personalizadas, permitiendo que el estudiante valide de forma progresiva sus competencias en la construcción de productos de software de alta calidad.

7. **Métodos de evaluación:**

Aprendizaje en contacto con el docente (30%)	Aprendizaje práctico experimental (70%)	Aprendizaje autónomo (0%)
Exposiciones, Participación en clases, Debates, Exámenes escritos u orales, Talleres, Defensa de proyectos, entre otros.	Resolución de problemas prácticos, Prácticas de laboratorio, Salidas de campo o visitas técnicas, Manejo de software especializado, Prototipado técnico, Estudios de caso técnicos, entre otros.	Elaboración de informes, Resolución de problemas y ejercicios, Ensayos de investigación, Creación de mapas conceptuales, Participación en foros, entre otros.

8. Referencias

Obligatoria:

- Laudon, K. C., & Laudon, J. P. (2021). Sistemas de información gerencial (14ª ed.). Pearson Educación.
- Pressman, R. S., & Maxim, B. R. (2021). Ingeniería del software: Un enfoque práctico (8ª ed.). McGraw-Hill Education.
- Sommerville, I. (2019). Ingeniería de software (10ª ed.). Pearson Educación.

Adicional:

- Fowler, M. (2022). UML Distilled: A Brief Guide to the Standard Object Modeling Language (4th ed.). Addison-Wesley.
- Hunt, A., & Thomas, D. (2020). The Pragmatic Programmer: Your Journey to Mastery (20th Anniversary ed.). Addison-Wesley Professional.
- Larman, C. (2022). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Pearson.
- Martin, R. C. (2022). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.