



000027

Ingeniería de Software 1

INSTITUTOS/S: INSTITUTO DE TECNOLOGÍA E INGENIERÍA

CARRERA/S: Licenciatura en Informática

RESPONSABLE DE LA ASIGNATURA Y EQUIPO DOCENTE:
Marisa Daniela Panizzi

Lucas Bonadeo

AÑO: 2025

CRÉDITOS: 7

CARGA HORARIA DE INTERACCIÓN PEDAGÓGICA: 64

CARGA HORARIA TOTAL: 175

CÓDIGO DE LA MATERIA EN SIU: xx

000027

1. Fundamentación

El desarrollo de sistemas es una actividad compleja, donde la obtención del producto software se realiza mediante un conjunto de procesos denominados procesos de software, los cuales se componen de actividades y tareas. Los procesos de software son comunes en todos los proyectos de sistemas de información.

Es importante que los alumnos conceptualicen los procesos de software, los artefactos que se construyen en cada una de sus tareas, los roles que se asignan a las tareas, las buenas prácticas que se aplican y las herramientas automatizadas que se utilizan para darles agilidad.

Los proyectos de software se componen por procesos de software o también denominados procesos ingenieriles como la ingeniería de requerimientos, análisis y diseño, el desarrollo, las pruebas y la puesta en marcha del producto software. Además, los proyectos de software cuentan con procesos de gestión y procesos de soporte como la verificación, validación, gestión de configuración.

En función de las características de los proyectos de software se selecciona la metodología de desarrollo más adecuada para su aplicación. En base a esta decisión, la combinación de las actividades y tareas, así como también los roles, los artefactos, prácticas y métodos difieren de una metodología a otra, pero su aplicación sistémica y controlada contribuye a un producto de software con calidad.

2. Propósitos y/u objetivos

- *Comprender la relación de los sistemas de información y su vinculación con la Ingeniería de software.*
- *Reflexionar que para llevar a cabo un proyecto de desarrollo de software hace falta llevar a cabo varias actividades además de programar, y tener una noción de cuáles son estas actividades y las técnicas asociadas a cada una.*

000027



- *Diferenciar la existencia de distintos modelos de procesos o ciclos de vida para el desarrollo de software.*
- *Conocer el concepto de metodología como definición de las actividades que involucran el desarrollo de software, la generación de productos de software y los roles que ocupan las personas que participan.*
- *Conocer los conceptos principales asociados a metodologías ágiles y estructuradas, las actividades y roles que involucran, y algunas similitudes y diferencias entre ambos enfoques.*
- *Identificar requerimientos funcionales y no funcionales, y que puedan especificarse mediante diferentes técnicas, casos de uso e historias de usuario.*
- *Definir el sistema de software mediante diagramas estructurales, diagrama de clases, y definir interfaces de usuarios mediante prototipos de interfaz.*
- *Aplicar herramientas CASE para las actividades de la construcción de software, entre ellas herramientas de modelado y herramientas de diseño de interfaz.*

3. Contenidos mínimos:

Ingeniería de software. Software vs. Sistemas de Información. El proceso del software. modelos de procesos o ciclos de vida para el desarrollo de software. Metodologías ágiles: actividades, productos, formas de articulación, roles. Ejemplo: Scrum. Metodologías estructuradas: actividades, productos, formas de articulación, roles. Ejemplo: UP. Similitudes y diferencias entre metodologías ágiles y estructuradas. Introducción a los requerimientos. Requerimientos funcionales vs. Requerimientos no funcionales. Herramientas de modelado.

4. Carga horaria

Créditos	Interacción pedagógica		Trabajo Autónomo	Total
7	Teoría: 32	Práctica: 32	111	175

4.1. Trabajo autónomo de la/el estudiante

Actividad	Carga horaria
Lectura de contenidos teóricos apuntes o bibliografía.	25

000027



<i>Desarrollo de los ejercicios prácticos propuestos (trabajo práctico integrador y guía de trabajos prácticos).</i>	25
<i>Investigar herramientas de modelado.</i>	11

5. Programa analítico

UNIDAD 1: Surgimiento y problemáticas del desarrollo de software.

Evolución de la computación y los sistemas de software. La vinculación entre los sistemas de información y la Ingeniería de software. La naturaleza del software.

UNIDAD 2: Procesos de software.

Ingeniería de software. Procesos de la Ingeniería de software. Modelos de desarrollo de software o ciclos de vida. Modelo de cascada. Modelo Iterativo-incremental. Modelo evolutivo. Prototipado. Metodologías orientadas al plan y metodologías ágiles. El Proceso unificado y Scrum. El manifiesto ágil. Herramientas CASE.

UNIDAD 3: Introducción a los requerimientos de un proyecto de software.

Importancia de los requerimientos en la construcción del producto software. Requerimientos funcionales y no funcionales. Identificación y especificación. Elicitación de requerimientos. Entrevistas, cuestionarios y observación directa. Requerimientos en los distintos modelos de ciclos de vida de desarrollo de software. Diagramas de casos de uso. Historias de usuario.

UNIDAD 4: Proceso unificado.

Ideas centrales. Fases, roles y artefactos. Modelos del proceso unificado. Diagramas de comportamiento: casos de uso, diagramas de estado, diagrama de comunicación, diagramas de actividad. Diagramas estructurales: clases.

6. Ejes y enunciados Multidimensionales y transversales

Eje	Nivel de logro de Aprendizaje	Acciones de enseñanza
-----	-------------------------------	-----------------------

000027



<i>C1. Identificación, formulación y resolución de problemas de informática</i>	<i>Alto</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C2. Concepción, diseño y desarrollo de proyectos de informática</i>	<i>Medio</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C3. Gestión, planificación, ejecución y control de proyectos de informática</i>	<i>Bajo</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C4. Utilización de técnicas y herramientas de aplicación en la informática</i>	<i>Medio</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas</i>	<i>No aporta</i>	<i>En esta materia no se realizan desarrollos tecnológicos y/o innovaciones tecnológicas.</i>
<i>C6. Fundamentos para el desempeño en equipos de trabajo</i>	<i>Alto</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C7. Fundamentos para la comunicación efectiva</i>	<i>Alto</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C8. Fundamentos para la acción ética y responsable</i>	<i>Bajo</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local</i>	<i>Medio</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C10. Fundamentos para el aprendizaje continuo</i>	<i>Alto</i>	<i>Clase expositiva, clase práctica, resolución de ejercicios y métodos de casos.</i>
<i>C11. Fundamentos para la acción emprendedora</i>	<i>No aporta</i>	<i>Esta materia no cuenta con contenidos vinculados a la acción emprendedora de un profesional informático.</i>

7. Bibliografía y recursos.

7.1. Bibliografía obligatoria

Pressman, Roger (2021). "Ingeniería del Software: un enfoque práctico", 9ª edición, Editorial McGraw-Hill.

000027

Sommerville, Ian (2016). "Ingeniería del Software", 10ª edición, Editorial Pearson Educación.

Jacobson Ivar, Booch Grady, Rumbaugh James (2000). "El proceso Unificado de Desarrollo de Software", Editorial Pearson - Addison Wesley.

Jacobson Ivar, Booch Grady, Rumbaugh James (2007). "El lenguaje unificado de modelado-Manual de referencia". 2ª Edición Editorial Pearson - Addison Wesley.

7.2. Bibliografía optativa

Kuhrmann, Marco, Tell Paolo, Klünder Jil, Hebig Regina, Licorish Sherlock, Stephen MacDonell Stephen (Eds.): Complementing Materials for the HELENA Study (Stage 2). [online] DOI: 10.13140/RG.2.2.11032.65288, published: 2018-11-28.

Palacio J. (2015). Scrum Manager. "Gestión de Proyectos SCRUM Manager" (Scrum Manager I y II). Disponible en: <http://www.scrummanager.net>.

Schwaber K. y Sutherland J. (2013). "La Guía Definitiva de Scrum: Las reglas del juego".

7.3. Recursos

MOOC Diagramas UML Estructurales para la Ingeniería de Software. Realizado por Óscar Pastor, Ignacio Panach de la Universidad Politécnica de Valencia. https://www.youtube.com/playlist?list=PL6kQim6ljTJvLt8gBznAzwrBQf_UDfobY

Para el diseño de interfaces: Figma. Link: <http://www.figma.com/>

Para el modelado: SmartUML. Link: <https://sourceforge.net/projects/smartuml/> o similar.

8. Metodología de enseñanza

8.1 Modalidades u opciones pedagógicas

Las clases serán en modalidades teórica, práctica y teórico-práctico dependiendo del tema a desarrollar.

000027

En las clases teóricas se reforzará con un material de lectura (artículos, capítulos de libros, y links a sitios de internet) y en algunos casos con cuestionarios guía.

Los alumnos trabajan de manera colaborativa conformando grupos de trabajos para el desarrollo de la guía de trabajos prácticos y el trabajo práctico integrador. Durante las clases se realizan puntos de control de las actividades prácticas de cada uno de los grupos. Se dará a los alumnos un guía de trabajos prácticos con casos prácticos con el objetivo de aplicar los conceptos trabajados en clase.

8.2 Formación Práctica

Durante el desarrollo de la cursada, se trabajará con los alumnos en un trabajo práctico integrador grupal, con entregas incrementales y se realizarán puntos de control semanales donde tendrán oportunidad de poner en práctica los temas teóricos de las diferentes unidades del programa de la materia. El trabajo práctico integrador finalizará con una exposición de cada grupo frente al resto de sus compañeros.

Algunos de los casos prácticos de la guía de trabajos prácticos y del trabajo práctico integrador se desarrollan en el laboratorio, para el modelado de los diagramas de casos de uso, diagrama de clases se emplea la herramienta SmartUML o similar y para el prototipado de las interfaces de software se emplea la herramienta Figma.

9. Evaluación y régimen de aprobación

9.1 Modalidad de evaluación

El sistema normal de evaluación consistirá en 2 (dos) exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descripta en el programa. Los mismos se realizarán en las fechas que, a tal efecto, se establezcan en el cronograma. El primer parcial es teórico-práctico e individual y el segundo parcial es un trabajo integrador desarrollado por los alumnos de manera grupal y se debe defender por todos los integrantes del grupo de manera oral. Además, se considera como parte de la evaluación de la cursada el desarrollo de la guía de trabajos prácticos que los alumnos deben presentar de manera grupal al final de la cursada de la asignatura. El alumno deberá poseer una asistencia no inferior al 75% en las clases presenciales.

000027

9.2 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

9.3 Acreditación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.