



INSTITUTO: Tecnología e Ingeniería

CARRERA/S: Tecnicatura Universitaria en Informática / Licenciatura en Informática

MATERIA: Elementos de Ingeniería de Software

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Marisa Daniela Panizzi

CUATRIMESTRE: Tecnicatura: 2do. / Licenciatura: 1ero.

AÑO: Tecnicatura: 2do. / Licenciatura: 3ro.

PROGRAMA N°: Tecnicatura: 16 / Licenciatura: 20

FECHA (de aprobación): 15/02/2022

Instituto/s: Tecnología e Informática

Carrera/s: Tecnicatura Universitaria en Informática.

Nombre de la materia: Elementos de Ingeniería de Software

Responsable de la asignatura y equipo docente: Marisa Daniela Panizzi

Cuatrimestre y año: Tecnicatura: 2do. Del 2do año / Licenciatura: 1ero. Del 3er año

Carga horaria semanal: 6 horas

Programa N°: Tecnicatura: 16 / Licenciatura: 20

Código de la materia en SIU: 760

Elementos de Ingeniería de Software

1. Fundamentación

El desarrollo de sistemas es una actividad compleja, donde la obtención del producto software se realiza mediante 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/las estudiantes conceptualicen estos modelos, 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 sistemática y controlada contribuye a un producto de software con calidad.

Los/las alumnos/as deben internalizar que el uso de buenas prácticas a la gestión de los proyectos de software como la estimación del esfuerzo y la gestión de riesgos contribuyen no solo a lograr que el proyecto sea exitoso, sino que también permite la obtención de un producto software de calidad y que satisfaga al cliente.

2. Propósitos y/u objetivos

Objetivos:

Que el estudiante:

- Comprenda la relación de los sistemas de información y su vinculación con la Ingeniería de software.
- Reflexione que para llevar a cabo un proyecto de desarrollo de software hace falta llevar a cabo varias actividades además de programar, y tenga una noción de cuáles son estas actividades y las técnicas asociadas a cada una.
- Diferencie la existencia de distintos modelos de procesos o ciclos de vida para el desarrollo de software.
- Conozca el concepto de metodología como definición de las actividades que involucra el desarrollo de software, la generación de productos de software y los roles que ocupan las personas que participan.
- Conozca los conceptos principales asociados a metodologías ágiles y estructuradas, las actividades y roles que involucran, y algunas similitudes y diferencias entre ambos enfoques.
- Pueda identificar requerimientos funcionales y no funcionales, y que pueda especificarlos mediante diferentes técnicas, casos de uso e historias de usuario.
- Defina el sistema de software mediante diagramas estructurales, diagrama de clases, que defina interfaces de usuarios mediante prototipos de interfaz.
- Comprenda la relevancia de los distintos tipos de testing existentes y el alcance de cada uno de ellos e identifique cuales son los más relacionados con las actividades de un programador.
- Aplique herramientas CASE para las actividades de la construcción de software, entre ellas herramientas de modelado y herramientas de diseño de interfaz.
- Tenga una pequeña experiencia práctica aplicando las actividades y metodologías que se describen en la materia.

3. Programa sintético:

- Teoría general de sistemas. Sistemas de información. Metodologías ágiles: actividades, productos, formas de articulación, roles. Ejemplos: Scrum. Metodologías estructuradas: actividades, productos, formas de articulación, roles. Ejemplos: UP. Similitudes y diferencias entre metodologías ágiles y estructuradas. El proceso del software. Concepto de ciclo de vida, relación con distintas metodologías. Métricas: qué son, qué miden, para qué y cuándo sirven. Estimación de esfuerzos. Conceptos de requerimiento funcional y no funcional. Distintos tipos de testing: de unidad, funcional, de sistema, de stress, de carga. Noción de cobertura. Tests automáticos, integración continua, interacción de las actividades de coding y refactor. Noción de

TDD. Nociones de riesgo y plan de contingencia. Ingeniería de Software de sistemas de tiempo real.

4. Programa analítico

4.1 Organización del contenido:

Unidad 1: Surgimiento y problemáticas del desarrollo de software

Evolución de la computación y los sistemas de software. Conceptos básicos de la teoría general de sistemas. 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. Prototipeado. Metodologías orientadas al plan y metodologías ágiles. El Proceso unificado y Scrum. El manifiesto ágil.

Unidad 3: Gestión de proyectos.

Definición de proyecto. Variables de proyecto. Criterios de éxito. Estimación y planificación. Técnica de estimación basada en puntos de caso de uso. Técnica de estimación basada en puntos de historias de usuario. Formas de contratación.

Unidad 4: Herramientas y artefactos para el inicio de proyecto

Visión de proyecto. Definición de Alcance. Visual Story Mapping. User Stories. Modelado de Dominio y modelado del negocio.

Unidad 5: 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. Diagramas de casos de uso. Historias de usuario.

Unidad 6: Proceso unificado.

Ideas centrales. Fases, roles y artefactos. Diagramas de comportamiento: casos de uso, diagramas de estado, diagrama de comunicación. Diagramas estructurales: clases. Especificación con casos de uso. Vista 4+1 de arquitectura.

Unidad 7: Buenas Prácticas en la construcción del software.

Uso de métricas. Tipos de métricas. Utilidad. Riesgos. Plan de contingencia. Mitigación de riesgos. Gestión de configuración. Elementos de la gestión de configuración. Herramientas para la gestión de configuración. Uso de patrones. Herramientas CASE.

Unidad 8: Otras actividades de la construcción del software.

Distintos tipos de testing: de unidad, funcional, de sistema, de stress, de carga. Noción de cobertura. Tests automáticos, integración continua, interacción de las actividades de coding y refactor. Noción de TDD. Puesta en marcha. El movimiento Devops. Entrega continua.

4.2 Bibliografía y recursos obligatorios:

Pressman, Roger (2010). "Ingeniería del Software: un enfoque práctico", 7º edición, Editorial McGraw-Hill.

Sommerville, Ian (2011). Ingeniería del Software, 7º edición, Editorial Pearson Educación.

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

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.

4.3 Bibliografía optativa:

Libros.

Pfleeger S. L. (2002). Ingeniería de Software. Teoría y Práctica, Editorial Prentice Hall, 2002.

Péaire C., Edwards M, Fernandes A., Mancin E. y Carroll K. (2007). The IBM Rational Unified Process for Systems, IBM. Rational Software.

Méndez, Erwin (2018). Estimación de Esfuerzo en proyectos de Desarrollo de software con metodologías ágiles. Tesis de Fin de Master Universitario en Dirección y Gestión de Proyectos. Escuela Superior de Ingenieros Industriales. Universidad Politécnica de Valencia.

Artículos.

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.

Fuggetta A., Di Nitto, E. Software Process. In: Proceedings of the 36th International Conference on Software Engineering - Future of Software Engineering (FOSE'14), pp. 1-12, 2014.

Bass L., Weber I., Zhu L. DevOps: A Software Architect's Perspective, 2015.

Erich F.M.A., Amrit C., Daneva M. "A Qualitative Study of DevOps Usage in Practice". Software: Evolution and Process, 29, pp. 1-20 (2017).

Díaz J., Pérez J., Yague A., Villegas García A., de Antona A. DevOps in Practice – A preliminary Analysis of two Multinational Companies. In: Proceeding of the 20 th Product-Focused Software Process Improvement. (PROFES 2019), pp. 323-330, 2019.

Vázquez P., Marisa Panizzi M., Bertone R. (2018). Estimación del esfuerzo del proceso de implantación de software basada en el método de puntos de caso de uso. Mar del Plata, 29 y 30 de Noviembre de 2018 Universidad Tecnológica Nacional. 6to. Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNalISI 2018). Simposio de Ingeniería de Sistemas y de Software. (En línea) ISSN: 2347-0372.

Felderer M. and Ramler R. Risk orientation in software testing processes of small and médium enterprises: an exploratory and comparative study. Software Qual J (2016) 24:519–548.

Kirman M. and Wahid A. Revised Use Case Point (Re-UCP) Model for Software Effort Estimation. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 3, 2015.

Belalcázar A., Díaz J, Molinari L., Rodríguez C. Principios, Roles y Métricas en alineamiento estratégico de nuevos requerimientos utilizando DevOps. En el libro de actas del XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016).

Links:

Ambler S. (2006). The Agile Unified Process (AUP). Recuperado el 20 de 03 de 2020, de <http://www.ambyssoft.com/unifiedprocess/agileUP.html>

Agile Business Consortium. <https://www.agilebusiness.org/page/generation-agile>

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

Scaled Agile. <https://www.scaledagileframework.com/continuous-deployment>.

Visual Story Mapping. Recuperado el 11 de noviembre de 2020, de <https://miro.com/templates/visual-story-map/>

Herramientas propuestas:

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

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

5. Metodologías de enseñanza:

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

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 para realizar a través del campus.

Se dará a quienes cursen la materia una guía de trabajos prácticos con casos prácticos ejercicios para realizar con el objetivo de asentar los conceptos trabajados en clase. Algunos de estos ejercicios formarán parte de su evaluación.

Durante toda la cursada, se trabajará 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 desarrollados durante la cursada de la asignatura. El trabajo práctico integrador finalizará con una exposición de cada grupo frente al resto de sus compañeros.

Plan de trabajo en el campus:

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía, así como también el programa y cronograma de la asignatura y la guía de trabajos prácticos y el trabajo práctico integrador, el simulacro del primer parcial.

6. Actividades de investigación y extensión (si hubiera)

En la materia no se desarrollan actividades de investigación en el marco de un proyecto de investigación. No obstante, algunos puntos de la guía de trabajos prácticos como el trabajo práctico integrador se presentan algunos puntos que requieren que los alumnos investiguen para poder resolverlos. También se realiza lectura de artículos de congresos. Estas actividades tienen el objetivo de acercar a los alumnos a la investigación y de que reconozcan su importancia en el marco de una carrera universitaria.

7. Evaluación y régimen de aprobación

7.1 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.

Se requerirá una asistencia no inferior al 75% en las clases presenciales.

En cuanto a las actividades virtuales se requerirá que el estudiante ingrese al aula virtual como mínimo una vez por semana.

7.2 Aprobación 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.

7.3 Criterios de calificación

El sistema normal de evaluación consistirá en 2 (dos) exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descrita 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. 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.

8. Cronograma

#CLASE	TEMAS A DESARROLLAR	PRESENTACIÓN	OBSERVACIONES	ACTIVIDADES	MODALIDAD
1	Teoría General de Sistemas. Sistemas de información. Tipos de SI.	Introducción a los sistemas	Presentación: Pautas de la materia. Guía de TPs. Trabajo Grupal Integrador. Documento Tipo – Carátulas para los Trabajos Prácticos.	Armar los grupos de trabajo de 4 a 6 integrantes, de ser posible que los grupos se conformen de manera mixta. Realizar la encuesta inicial de la materia. Desarrollar los casos prácticos de la Guía de TPs. Profundizar la lectura del “Enfoque de Caja Negra” en el apunte de Zenón.	Presencial.
2	Ingeniería de Software. Proceso ingenieril y sus elementos.	Introducción a la Ingeniería de Software	Libros a utilizar.	Desarrollar los casos prácticos de la Guía de TPs.	Virtual. Sincrónica.
3	Método de selección del modelo de ciclo de vida.	Selección del MCV	Grilla para el cálculo. Artículo del método (Inglés y Español).	Profundizar la lectura del artículo del método. Trabajo Práctico Integrador:	Virtual Sincrónica.
4	Introducción a la Ingeniería de Requerimientos. Requerimientos de software (RF y RNF). Identificación y especificación.	Introducción a la Ingeniería de Requerimientos	Materiales a compartir: Estadística de Stadish Group. Navegación por el Standard IEEE + traducción.	Profundizar la lectura de los libros de Ingeniería de Software Desarrollar los casos prácticos de la Guía de TPs. Trabajo Práctico Integrador:	Virtual Sincrónica.
5	Técnicas de modelado de requerimientos: casos de uso	Técnicas de modelado: casos de uso	Compartir: plantilla de especificación de casos de uso y otro formato de especificación	Profundizar la lectura del libro de UML y libros de Ingeniería de Software Desarrollar los casos prácticos de la Guía de TPs. Trabajo Práctico Integrador:	Presencial.
6	Técnicas de modelado de requerimientos:	Técnicas de modelado:	Compartir: plantilla de historias de usuario.		Presencial.

	historias de usuario.	historias de usuario.			
7	Repaso de requisitos - Práctica				Virtual Sincrónica.
8	Proyectos de software. Presentación de las actividades de un proyecto de software. Metodologías de desarrollo. Ágiles y Robustas. RUP & SCRUM.	Metodologías de desarrollo.	Compartir: Encuesta Internacional HELENA - Uso de metodologías Links: Scrum / Agile Manifesto.	Consultas de actividades prácticas.	Virtual Sincrónica.
9	Diagrama de Clases	Uso de Mooc sobre diagramas estructurales de UML – Diagrama de Clases	Compartir: link al Mooc	Desarrollar los casos prácticos 23, 24, 25 y 26 de la Guía de Tps. Trabajo Práctico Integrador: Punto 8	Virtual Sincrónica.
10	Simulacro de parcial			Consultas para el parcial	Presencial.
11	Examen - Primer Parcial.		Parcial - Duración: 3 hs.		Presencial.
12	Pruebas.				Virtual Sincrónica.
13	Devops. Integración continua. Despliegue continuo.			Devolución de parciales.	Presencial
14	Gestión de configuración + Gestión de riesgos.	Gestión de configuración		Trabajo Práctico Integrador:	Virtual Sincrónica.
15	Examen Segundo Parcial – Defensa oral		Recepción de la presentación y de la carpeta con el Trabajo Práctico Integrador, fecha a definir por el docente.	<u>Evaluación oral de los Trabajos Prácticos Integradores x grupo</u>	Presencial.

				Modalidad grupal. TODOS LOS INTEGRANTES DEBEN ESTAR PRESENTES.	
16	Recuperatorios			<u>Revisión de correcciones del Trabajo Práctico integrador y de la Guía de Trabajos Prácticos por grupo.</u> Finalización del cuatrimestre. Indicaciones para la evaluación integradora	Presencial.
	Evaluación integradora (en caso de ser necesario).				