



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2016)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DEL SOFTWARE	ING. EN COMPUT.	28/12	2016	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ABDELAHAD, CORINA NATALIA	Prof. Responsable	P.Adj Exc	40 Hs
BERNARDIS, EDGARDO	Auxiliar de Práctico	A.1ra Simp	10 Hs
PEREZ, NORMA BEATRIZ	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	2 Hs	2 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
14/03/2016	24/06/2016	15	90

IV - Fundamentación

Se introduce al alumno en el desarrollo de software como una actividad ingenieril y al trabajo en equipo. Se pretende que el alumno aprenda los fundamentos básicos de cada una de las etapas que comprende el proceso de desarrollo de software así como las distintas técnicas y metodologías aplicables, conocimientos de los principios de diseño y de las principales técnicas de validación y verificación del software incluyendo conceptos de gestión de proyectos.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el alumno sea capaz de asimilar los conceptos de procesos de desarrollo de software, desde su especificación, análisis, diseño hasta su verificación y validación, incorporando conceptos de gestión de proyectos. El objetivo es que el estudiante tome conocimiento de conceptos básicos de ingeniería de software desde un punto de vista gerencial, transversal al desarrollo en sí, así como también de la automatización de procesos de desarrollo de software. Para cubrir dichos objetivos se integrarán conceptos, modelos y métodos en un proyecto integrador.

VI - Contenidos

CONTENIDOS MINIMOS:

El proceso de software: Ciclos de vida, herramientas. Ingeniería de requerimientos, introducción a los métodos formales. Análisis, diseño, implementación, verificación, validación y mantenimiento de software. Gestión de proyectos: Planificación, métricas, estimaciones, análisis y gestión del riesgo. Conceptos de calidad de software.

PROGRAMA:

Unidad I

Ingeniería de Software. Principios. Modelos. Software. Conceptos. Evolución del software. Características del software. Ciclos de vida. Modelos de procesos de producción de software. Proceso, metodologías y herramientas. Evolución de las metodologías de desarrollo de sistemas de software. Ingeniería de requerimientos. Ingeniería de la Información.

Unidad II

Métodos Formales. Conceptos básicos. Deficiencias de los enfoques menos formales. Las matemáticas en el desarrollo de software. Notación matemática para la especificación formal. Lenguajes de especificación formal.

Unidad III

Gestión de Proyectos de Software. Conceptos. Producto. Proceso. Proyecto. Recursos humanos. Coordinación y comunicación. Ámbito del Software. Descomposición del proceso. Medidas, métricas e indicadores. Estimaciones. Modelos empíricos de estimación. Herramientas automáticas de estimación.

Unidad IV

Modelos en UML. Introducción. Modelos. Importancia de los modelos. Modelos estáticos. Clases: atributos, operaciones y responsabilidades. Relaciones: dependencia, generalización y asociación. Modelos estáticos y dinámicos. Diagrama de Interacción: Secuencia y Colaboración. Modelo de Casos de Usos.

Unidad V

Proceso de Desarrollo. Análisis, diseño, implementación, pruebas y mantenimiento de software. Modelo de Casos de Usos. Ingeniería de Requerimientos. Análisis y Diseño. Implementación. Metodologías Ágiles. Conceptos. Principios. Manifiesto. Estudio comparativo de metodologías ágiles: XP, Cristal, Scrum. Ventajas y desventajas de las metodologías ágiles. Estudio de utilización concreta de las metodologías ágiles en la industria del software: Extreme Programming.

Unidad VI

Análisis y Gestión del Riesgo. Estrategias de riesgo proactivas vs. reactivas. Riesgo del software. Identificación y evaluación global del riesgo del proyecto. Proyección del riesgo y evaluación del impacto. Seguridad. Planeamiento del Proyecto. Principios básicos. Distribución del esfuerzo. Definición y selección de tareas de ingeniería de software. Planificación temporal. Análisis de valor agregado. Plan de proyecto.

Unidad VII

Calidad del Software. Gestión de Configuración del Software. Conceptos. Control. Garantía. Costos. Aseguramiento de la calidad del software. Verificación, Validación. Pruebas de errores del software. Medidas de fiabilidad y disponibilidad. Líneas base, elementos y objetos en la configuración del software. Versiones.

VII - Plan de Trabajos Prácticos

Practico 1: La Ingeniería de Software y los modelos de ciclo de vida del software.

Práctico 2: Modelos en UML.

Práctico 3: Pruebas Estructural.

Práctico 4: Pruebas Funcional.

Práctico 5: Gestión de Proyecto.

Práctico 6: Especificaciones formales del software.

Laboratorio 1: Especificación formal de un caso de estudio haciendo uso de un lenguaje de especificación formal con herramientas automatizadas.

Laboratorio 2: Utilización de Herramientas que soportan la Gestión. Herramientas automáticas de estimación.

Laboratorio 3: Utilización de Herramientas para la generación de Pruebas Estructural y Funcionales.

Proyecto Integrador de Laboratorio: Se solicita la construcción de un proyecto real con un cliente a ser seleccionado para llevar adelante un caso de estudio donde se cubra todos los aspectos de la ingeniería de software.

VIII - Regimen de Aprobación

La materia se desarrolla con la modalidad de promoción sin examen final.

Existen dos niveles:

a) Regularización solamente: Para regularizar la materia se deberá:

1.- Tener como mínimo un 60% de asistencia a clases.

2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 6 (seis).

3.- Aprobar dos parciales con una nota mayor o igual a 6 (seis).

Cada parcial posee dos recuperaciones. En caso de no aprobar un parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 6 (seis).

b) Promoción sin examen final: Para regularizar y aprobar la materia se deberá:

1.- Tener como mínimo un 80% de asistencia a clases.

2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).

3.- Aprobar dos parciales o sus respectivas recuperaciones con una nota mayor o igual a 7 (siete). Cada parcial posee dos recuperaciones. En caso de no aprobar un parcial, deberá aprobar alguno de sus recuperatorios con nota mayor o igual a 7 (siete).

4.- Aprobar una prueba final integradora con una nota mayor o igual a 7 (siete).

Aquellos alumnos que sólo regularicen la materia deberán rendir un examen final, en los turnos establecidos.

Alumnos Libres: Por las características propias del proyecto de laboratorio a desarrollarse durante todo el cuatrimestre, no se aceptan alumnos libres.

IX - Bibliografía Básica

[1] J. O' Brien & G. Marakas, "Introduction to Information Systems", 15th edition.

[2] Pressman, Roger; "Ingeniería del Software - Un enfoque práctico", Mc Graw Hill, 6ta Edición. 2006.

[3] Pankaj Jalote; "An Integrated Approach to Software Engineering, 3ra edición". Springer. 2005.

[4] Booch, Rumbaugh, Jacobson. "The Unified Modeling Language User Guide, 2nd Edition". Addison-Wesley, 2005

[5] Booch, Rumbaugh, Jacobson. "The Unified Modeling Language Reference Manual, 2nd Edition". Addison-Wesley, 2005.

[6] Jorgensen, Paul C.; "Software Testing, a Craftsman's Approach", CRC Press, 1995.

[7] Ghezzi, Carlo y otros "Fundamentals of Software Engineering", Prentice Hall, 1991

[8] Daniel Bolaños y otros; "Pruebas de software y JUnit", Prentice-Hall, 2008.

[9] Martín Fowler, The New Methodology, <http://www.martinfowler.com/articles/newMethodology.html>

[10] Kent Beck, "Extreme Programming Explained", 1ra edición, 1999.

[11] Scott Ambler, "Agile Modeling: Effective practices for Extreme Programming and the Unified Process", John Wiley & Sons, 2002.

[12] CTR, "Information Systems Strategic Planning", Computer Technology Research Corp., 4th Edition, 1994

[13] Daniel Jackson; "Software Abstractions", The MIT Press, 2012.

[14] Apuntes de la Cátedra.

X - Bibliografía Complementaria

[1] Booch, Grady. Object-Oriented analysis and design with applications. The Benjamin/Cummings Publishing Company Inc. 1994.

[2] Stephen Schach. "Ingeniería de Software Clásica y Orientada a Objetos, 6ta ed.", Mc Graw Hill, 2005.

[3] Sommerville, Ian; "Software Engineering", Addison-Wesley, 6ta Edición. 2002.

[4] Beck, Kent. Test-driven development by example. Pearson Education, 2003.

[5] Martin, James, "Information Engineering", Prentice Hall, 1991.

[6] Hammer M., Champy, J. "Reingeniería", Norma, 1993.

[7] Fenton, N.E., Pfleeger, S.L., Software Metrics: a Rigorous and Practical Approach, 2nd Ed., PWS Publishing Company, 1997.

[8] Arthur, Lowell Jay, "Rapid Evolutionary Development", Wiley, 1992.

[9] Scott Ambler, "Agile Modeling and the Unified Process", <http://www.agilemodeling.com/essays/agileModelingRUP.htm>, 2002.

[10] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, "Agile Software Development Methods: Review and Analysis", VTT, 2002.

[11] "Extreme Programming: A gentle introduction", <http://www.extremeprogramming.org/>

[12] Página Web del Instituto de Ingeniería de Software (CMU). <http://www.cmu.edu>

XI - Resumen de Objetivos

Asimilar los conceptos de procesos de desarrollo de software, desde su especificación, análisis, diseño hasta su verificación y validación, incorporando conceptos de gestión de proyectos.

XII - Resumen del Programa

Introducción a la Ingeniería de Software. Métodos Formales. Gestión de Proyectos de Software. Modelos en UML. Procesos de Desarrollo. Análisis y Gestión del Riesgo. Calidad del Software.

XIII - Imprevistos

--

XIV - Otros

--