



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 2020)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DEL SOFTWARE	ING. EN COMPUT.	28/12	2020	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 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	1 Hs	3 Hs	6 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
09/03/2020	19/06/2020	15	90

IV - Fundamentación

Se introduce al alumno en el desarrollo de software como una actividad ingenieril y 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 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.

VI - Contenidos

CONTENIDOS MÍNIMOS:

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. Producto. Proceso, metodologías y herramientas. Evolución de las metodologías de desarrollo de sistemas de software. Ingeniería de la Información.

Unidad II

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

Unidad III

Proceso de Desarrollo. Análisis, diseño, implementación, pruebas y mantenimiento de software. Ingeniería de Requerimientos. Metodologías Ágiles. Conceptos. Principios. Ventajas y desventajas de las metodologías ágiles. Estudio de utilización concreta de las metodologías ágiles en la industria del software: Extreme Programming.

Unidad IV

Calidad del Software. Conceptos. Control. Verificación, Validación. Pruebas de errores del software. Medidas de fiabilidad y disponibilidad.

Unidad V

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 VI

Gestión de proyectos. Medidas, métricas e indicadores. Estimaciones. Análisis y Gestión del Riesgo. Riesgo del software. Identificación y evaluación global del riesgo del proyecto. Distribución del esfuerzo. Definición y selección de tareas de ingeniería de software. Planificación.

VII - Plan de Trabajos Prácticos

Debido a la pandemia del COVID-19, el presente es un programa en fase no presencial para el cual se tuvo que repensar y reprogramar la metodología de enseñanza y aprendizaje.

Para realizar un seguimiento del aprendizaje en los estudiantes en cada trabajo práctico se requiere la entrega de ciertos ejercicios. Para cada ejercicio solicitado se realiza una corrección y se le envía la misma al estudiante.

Laboratorio 1: Modelado Estático con UML. Ingeniería Directa e Ingeniería Inversa con Java.

Laboratorio 2: Modelado Dinámico con UML. Ingeniería Directa e Ingeniería Inversa con Java.

Laboratorio 3: Utilización de Herramientas que soportan la Gestión. Herramientas automáticas de estimación.

Laboratorio 4: Especificaciones formales del software

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

Práctico 2: Pruebas Estructural.

Práctico 3: Pruebas Funcional.

VIII - Regimen de Aprobación

Considerando que este Programa se desarrolla en Fase No Presencial se tuvieron en cuenta las siguientes consideraciones:

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

Existen dos niveles:

a) Regularización: Para regularizar la materia se deberá:

1.- Tener como mínimo un 70% de asistencia, la misma se considera con las entregas de los ejercicios solicitados en cada trabajo práctico.

2.- Aprobar una evaluación, o cualquiera de sus dos recuperaciones, con nota mayor o igual a 6 (seis). Dicha evaluación se realizará en un horario establecido. Se colocará el enunciado en aulas virtuales y tendrán un tiempo para resolverlo; el alumno deberá resolverlo en papel y una vez que termine, deberá sacarle una foto y enviarlo por mail al docente. Posteriormente puede ser requerido un coloquio oral.

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

1.- Tener como mínimo un 80% de asistencia, la misma se considera con las entregas de los ejercicios solicitados en cada trabajo práctico.

- 2.- Aprobar una evaluación, o cualquiera de sus dos recuperaciones, con nota mayor o igual a 7 (siete). Dicha evaluación se realizará en un horario establecido. Se colocará el enunciado en aulas virtuales y tendrán un tiempo para resolverlo; el alumno deberá resolverlo en papel y una vez que termine, deberá sacarle una foto y enviarlo por mail al docente. Posteriormente puede ser requerido un coloquio oral.
- 3.- Aprobar una prueba final integradora con una nota mayor o igual a 7 (siete), la cual podrá ser oral y/o escrita.

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

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] Sommerville, Ian; "Software Engineering", Addison-Wesley, 7ma Edición. 2005.
- [4] Pankaj Jalote; "An Integrated Approach to Software Engineering, 3ra edición". Springer. 2005.
- [5] Booch, Rumbaugh, Jacobson. "The Unified Modeling Language User Guide, 2nd Edition". Addison-Wesley, 2005
- [6] Booch, Rumbaugh, Jacobson. "The Unified Modeling Language Reference Manual, 2nd Edition". Addison-Wesley, 2005.
- [7] Jorgensen, Paul C.; "Software Testing, a Craftsman's Approach", CRC Press, 1995.
- [8] Ghezzi, Carlo y otros "Fundamentals of Software Engineering", Prentice Hall, 1991
- [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] Beck, Kent. Test-driven development by example. Pearson Education, 2003.
- [4] Martin, James, "Information Engineering", Prentice Hall, 1991.
- [5] Hammer M., Champy, J. "Reingeniería", Norma, 1993.
- [6] Fenton, N.E., Pfleeger, S.L., Software Metrics: a Rigorous and Practical Approach, 2nd Ed., PWS Publishing Company, 1997.
- [7] Arthur, Lowell Jay, "Rapid Evolutionary Development", Wiley, 1992.
- [8] Scott Ambler, "Agile Modeling and the Unified Process", <http://www.agilemodeling.com/essays/agileModelingRUP.htm>, 2002.
- [9] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, "Agile Software Development Methods: Review and Analysis", VTT, 2002.
- [10] "Extreme Programming: A gentle introduction", <http://www.extremeprogramming.org/>

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

Ante las medidas nacionales tomadas por la pandemia del COVID-19, la materia se dicta de manera virtual mediante las plataformas moodle y meet. El seguimiento continuo se realiza mediante correo electrónico.

XIV - Otros

--