



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 2012)
 (Programa en trámite de aprobación)
 (Presentado el 08/08/2012 11:59:10)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DE SOFTWARE II	LIC.CS.COMP.	18/11	2012	2° cuatrimestre
INGENIERIA DE SOFTWARE II	LIC.CS.COMP.	006/0	2012	2° cuatrimestre
		5		

II - Equipo Docente

Docente	Función	Cargo	Dedicación
RIESCO, DANIEL EDGARDO	Prof. Responsable	P.Adj Exc	40 Hs
ABDELAHAD, CORINA NATALIA	Auxiliar de Práctico	A.1ra Semi	20 Hs
BERNARDIS, HERNAN	Auxiliar de Práctico	A.2da Simp	10 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	4 Hs	3 Hs	9 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
06/08/2012	16/11/2012	15	135

IV - Fundamentación

Dar las bases teóricas y prácticas que permiten al Ingeniero de Software aplicar distintos métodos de desarrollo utilizando herramientas capaces de automatizar las actividades que se realizan durante el proceso de desarrollo del software.

V - Objetivos / Resultados de Aprendizaje

Introducir al alumno en el desarrollo de sistemas aplicando métodos de desarrollo que permiten producir software de manera fiable, de calidad y que funcione en máquinas reales cubriendo las distintas etapas del proceso de desarrollo. Introducir a los alumnos a las Arquitecturas Orientadas a Servicios, Sistemas de Tiempo Real y Reingeniería de Sistemas de Software.

VI - Contenidos

Unidad 1: Requerimientos

Introducción. Requerimientos de Software. Estudio de Factibilidad. Análisis y elicitación de requerimientos. Modelado de los Requerimientos. Validación de Requerimientos. Gestión de requerimientos. Tipos de Sistemas. Clasificación.

Unidad 2: Modelos Avanzados Orientados a Objetos en UML

Introducción. Modelos. Importancia de los modelos. Modelos estáticos. Modelos dinámicos. Persistencia. Concurrencia. Estado. Comportamiento. Mecanismos comunes. Estereotipos. Valores etiquetados. Restricciones. Máquinas de Estado.

Modelo Arquitectónico. Componentes. Despliegue.

Unidad 3: Proceso Unificado: Requerimientos.

Introducción. Proceso Dirigido por Casos de Usos. Proceso Iterativo e Incremental. Proceso Centrado en la Arquitectura. Gestión de los Requerimientos. Modelo de Casos de Usos. Captura de Requerimientos. Análisis de Requerimientos. Contexto del Sistema. Modelo del Dominio. Validación de Requerimientos.

Unidad 4: Proceso Unificado: Análisis y Diseño.

Introducción. Propósito. Diferencias. Artefactos. Modelo del Análisis. Clases del Análisis. Realización de Casos de Uso del Análisis. Paquetes del análisis. Arquitectura. Flujo de Trabajo. Rol del diseño. Artefactos. Modelo del Diseño. Clases del Diseño. Realización de Casos de Uso del Diseño. Subsistemas. Interfaz. Arquitectura. Modelo de Desarrollo. Flujo de Trabajo. Aplicación de Patrones en el Diseño.

Unidad 5: Patrones de Diseño y Arquitectónicos

Introducción. Conceptos. Descripción. Selección de un patrón de Diseño. Utilización. Problema. Solución. Consecuencia. Catálogo de Patrones de Diseño: patrones creacionales, patrones estructurales y patrones de comportamiento. Arquitectura de Software. Patrones Arquitectónicos.

Unidad 6: Proceso Unificado: Framework

Introducción. Distintas Instanciaciones del Proceso. OpenUP. Modelo de Negocio. Relación con los requisitos. Análisis. Modelo de análisis. Clases de análisis. Realización de casos de uso-análisis. Análisis de la arquitectura. Relación con el Diseño. Pasos a la implementación. Modelo de implementación. Componentes. Modelo de pruebas. Casos de prueba. Procedimiento de prueba. Plan de prueba.

Unidad 7: Ingeniería de la Información y Basada en Componentes.

Ingeniería de la Información. Arquitectura de Datos. Arquitectura de Aplicaciones. Tecnología. Modelado del área de Negocio. Ingeniería de software basada en componentes. Reuso. Modelo de componentes. Composición de componentes.

Unidad 8: Conceptos de Arquitecturas Basadas en Servicios (SOA).

Conceptos básicos relacionados con SOA y con sistemas orientados a servicios en general. SOA y BPM (Gestión Orientada hacia los Procesos). Notación de Modelado de Procesos. BPMN. Nociones de Sistemas Colaborativos. Potencialidades y dificultades en la adopción de SOA. Importancia de la adopción de una estrategia SOA en la organización. Gestión de SOA. La evaluación contextual de la tecnología y el reconocimiento de un cambio de mentalidad como claves para el éxito de la adopción de SOA. XML. Introducción al Protocolo de Acceso a Objetos Simple. Escenarios Básicos. Mensajes SOAP. Intercambio de Mensajes. Modelo de Procesamiento SOAP. Arquitectura Web y SOAP. Escenarios Avanzados. Introducción a WSDL (Web Service Definition Language). Tipos de Mensajes. Interfaces. Herencia. Operaciones. Binding. Servicios. Documentación del Servicio. Modelo de Componente y Esquema. UDDI. Servicios Web. Arquitectura SOA. Estudio de un caso práctico de adopción de SOA.

Unidad 9: Reingeniería de Software

Concepto de Reingeniería de Software. Necesidad de Reingeniería de Software. Reingeniería entendida como “Ingeniería Reversa + Ingeniería Hacia Adelante”. La necesidad de soporte automatizado en la “Ingeniería Reversa”. Reingeniería de Modelos de Datos / Objetos. Reingeniería de Aplicaciones. Utilización de herramientas CASE en la Reingeniería de Software.

Unidad 10: Ingeniería de Software de Sistemas de Tiempo Real

Introducción a los sistemas de tiempo real. Planeamiento en sistemas monoprocesador. Planeamiento basado en prioridades fijas. Introducción a la gestión de recursos compartidos y a los protocolos de acceso. Planeamiento dinámico. Introducción al planeamiento de recursos en sistemas distribuidos. Nociones de planeamiento del uso de recursos tele informáticos. Introducción a sistemas distribuidos de tiempo real con planificación estática. Comunicación y sincronización basada en variables compartidas. Sincronización y comunicación basadas en mensajes. Acciones atómicas, procesos concurrentes y fiabilidad. Capacidades de tiempo real. Planificación. UML & Sistemas de Tiempo Real. Sistemas distribuidos de tiempo real con prioridades fijas. Sistemas de tiempo real acrílicos. Especificaciones de Java para tiempo real. Introducción a la utilización de las Redes de Petri para modelar sistemas de tiempo real distribuidos.

VII - Plan de Trabajos Prácticos

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

Laboratorio 2: Herramientas CASE de Ingeniería de Requerimientos.

Laboratorio 3: Herramientas CASE de Ingeniería de la Información.

Laboratorio 4: Construcción de Software bajo SOA

Laboratorio 5: Uso de Herramientas CASE en Ingeniería Reversa de Modelos de Datos / Objetos

Laboratorio 6: Uso de Herramientas basadas en Redes de Petri para el modelado de Sistemas de Tiempo Real Distribuidos.

Laboratorio 7: Diseño y Programación de Sistemas de Tiempo Real.

Práctico 1: Patrones de Diseño.

Práctico 2: Modelos del Proceso Unificado.

Laboratorio Integrador: Construcción de un software orientado a objetos usando herramientas que automatizan el proceso de desarrollo generando los distintos artefactos desde los requerimientos hasta su implementación con un caso de estudio real.

Deberán aplicar los distintos conceptos aprendidos y utilizados en teoría, en laboratorios anteriores y en las prácticas.

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 80% de asistencia a clases prácticas.

2.- Tener los prácticos, solicitados por la cátedra, aprobados, como método aplicado por la cátedra para la evaluación continua del alumno.

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

4.- Aprobar dos parciales o sus respectivas recuperaciones con una nota mayor o igual a 6 (seis).

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

1.- Cumplir con los requisitos a.1, a.2 y a.3.

2.- Aprobar dos parciales o sus respectivas recuperaciones con una nota mayor o igual a 7 (siete).

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

La nota final será la que surja de aplicar la siguiente fórmula:

$$\text{NotaFinal} = (\max(p1, r1) + \max(p2, r2) + pfi + pil) / 4$$

max : función maximo

p1 : primer parcial,

p2 : segundo parcial

r1 : recuperatorio del primer parcial

r2: recuperatorio del segundo parcial

pfi: prueba final integradora

pil : proyecto integrador de laboratorio.

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.

Alumnos que Trabajan: Los alumnos que trabajan tienen una recuperación adicional, según las Ordenanzas CS N° 26/97, 15/00 y 13/03.

IX - Bibliografía Básica

[1] Software Engineering: A Practitioner's Approach, 7/e, Roger S Pressman, R. S. Pressman & Associates, Inc. 2010.

[2] OpenUP/Basic, <http://epf.eclipse.org/wikis/openupsp/>, 2010.

[3] El Proceso de Desarrollo de Software Unificado. Booch, Rumbaugh, Jacobson. Addison-Wesley, 1999.

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

- [5] The Unified Modeling Language Reference Manual, 2nd Edition. Booch, Rumbaugh, Jacobson. Addison-Wesley, 2005.
- [6] Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison-Wesley. 1995.
- [7] Object-Oriented analysis and design with applications. Booch, Grady. The Benjamin/Cummings Publishing Company Inc. 1994.
- [8] Component-Based Software Engineering: Putting the Pieces Together, George T. Heineman, William T. Councill, Addison-Wesley Professional, 2001.
- [9] Software Engineering, Ian Sommerville, Addison Wesley; 8 edition, 2006
- [10] Enterprise SOA, Dirk Krafziq, Karl Banke, Dirk Slama, "The Coad Series", Prentice Hall, 2004 (disponible como free e-book)
- [11] SOAP - Messaging Framework, <http://www.w3.org/TR/soap12-part1/>
- [12] SOAP - Adjuncts, <http://www.w3.org/TR/soap12-part2/>
- [13] Web Services Description Language, <http://www.w3.org/TR/wsdl.html>
- [14] SOA: Open Source (Spanish Edition), Jeff Davis (Author), Anaya Multimedia, 2009
- [15] Sistemas de Tiempo Real y Lenguajes de Programacion - 3b: Edición, Alan Burns (Author), Andy Wellings, Pearson Educacion, 2005
- [16] Building Application Distributed Programming Web Services with SOAP, Doug Tidwell, James Snell, Pavel Kulchenko, O'Reilly, First Edition December 2001

X - Bibliografia Complementaria

- [1] Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustrated with UML. Mark Grand. John Wiley & Sons Inc. 1998.
- [2] UML Semantics. <http://www.omg.org> - UML Notation Guide. <http://www.omg.org>
- [3] UML y Patrones: Introducción al análisis y diseño orientado a objetos. Craig Larman, Prentice Hall, 1999.

XI - Resumen de Objetivos

Introducir al alumno en el desarrollo de sistemas aplicando métodos de desarrollo que permiten producir software de manera fiable, de calidad y que funcione en máquinas reales, cubriendo desde la especificación de requisitos hasta la obtención del producto, construyendo artefactos tanto formales como semi-formales. Introducir conceptos relacionados con Arquitecturas Orientadas a Servicios, Sistemas de Tiempo Real y Reingeniería de Sistemas de Software.

XII - Resumen del Programa

Requerimientos Modelos Orientados a Objetos en UML
 Proceso Unificado: Requerimientos. UML Avanzado. Patrones de Diseño. Proceso Unificado: Análisis y Diseño. Proceso Unificado: Framework Ingeniería de la Información y Basada en Componentes.
 Arquitecturas Orientadas a Servicios.
 Sistemas de Tiempo Real.
 Reingeniería de Sistemas de Software.

XIII - Imprevistos

.

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	