



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 2024)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DE SOFTWARE II	LIC.CS.COMP.	32/12	2024	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
RIESCO, DANIEL EDGARDO	Prof. Responsable	P.Asoc Exc	40 Hs
ABDELAHAD, CORINA NATALIA	Prof. Colaborador	P.Adj Exc	40 Hs
BERNARDIS, HERNAN	Responsable de Práctico	JTP Simp	10 Hs
THOMPSON, HORACIO JESUS	Auxiliar de Práctico	A.1ra 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	3 Hs	2 Hs	4 Hs	9 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
05/08/2024	15/11/2024	15	135

IV - Fundamentación

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

V - Objetivos / Resultados de Aprendizaje

Introducir al estudiante 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, comprendiendo y aplicando, además, conceptos de Arquitecturas Orientadas a Servicios, Sistemas de Tiempo Real y de Ingeniería Basada en Componentes. Introducir al estudiante conceptos de Reingeniería de Sistemas de Software.

VI - Contenidos

Contenidos mínimos: Estudio de Factibilidad de un Proyecto de Software. Clasificación de los Sistemas de Información. Ingeniería de Requerimientos. Gestión de los Requerimientos. Análisis de Requerimientos. Validación de Requerimientos. Desarrollo de Software enmarcado en el Paradigma de la Orientación a Objetos. Arquitectura y Diseño. Patrones. Frameworks de Aplicaciones. Ingeniería de la Información. Ingeniería de Software Basada en Componentes. Modelo de componentes. Composición de componentes. Ingeniería de Software de Sistemas de Tiempo Real. Nociones de Sistemas Colaborativos. Arquitecturas Basadas en Servicios. Reingeniería de los Sistemas de Software.

Unidad 1: Requerimientos

Introducción. Requerimientos de Software. Tipos. Procesos de la Ingeniería de Requerimientos. Estudio de Factibilidad. Obtención y Análisis. Especificación. Modelado. Validación de Requerimientos. Gestión de los Requerimientos. Clasificación de los Sistemas de Información.

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: Framework.

Introducción. Dirigido por Casos de Usos. Centrado en la Arquitectura. Iterativo e Incremental. Modelo de Casos de Usos. Captura de requisitos. Contexto del Sistema. Modelo del Dominio. Distintas Instancias del Proceso. Análisis de la arquitectura. Relación con el Diseño. Componentes.

Unidad 4: Patrones de Diseño

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.

Unidad 5: Proceso Unificado: Análisis y Diseño.

Introducción. Propósito. Diferencias. Artefactos. Modelo del Análisis. Arquitectura. Flujo de Trabajo. Rol del diseño. Artefactos. Modelo del Diseño. Subsistemas. Interfaz. Modelo de Desarrollo. Aplicación de Patrones en el Diseño.

Unidad 6: Ingeniería de la Información y Basada en Componentes.

Ingeniería de la Información. Clasificación. Modelado del área de Negocio. Modelo de Negocio. Notación de Modelado de Procesos. BPMN. Tecnología. Ingeniería de software basada en componentes. Reuso. Modelo de componentes. Desarrollo Basado en Componentes. Composición de Componentes. Arquitectura de Componentes.

Unidad 7: XML.

Introducción. Estructura. Elementos. Atributos. Subelementos. Documento bien formado. Espacio de Nombres. Esquema. Declaración de Tipos de Datos. XPath. XQuery.

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

Introducción. SOA y BPM (Gestión Orientada hacia los Procesos). Nociones de Sistemas Colaborativos. Importancia de la adopción de una estrategia SOA en la organización. La evaluación contextual de la tecnología y el reconocimiento de un cambio de mentalidad como claves para el éxito de la adopción de SOA. Beneficios. Introducción al Protocolo de Acceso a Objetos Simple. Modelo de Procesamiento SOAP. Arquitectura Web y SOAP. WSDL (Web Service Definition Language). Servicios Web. Arquitectura SOA.

Unidad 9: Reingeniería de Software

Concepto de Reingeniería de Software. Reingeniería de Procesos de Negocio. Reingeniería de Software. Ingeniería Inversa. Reestructuración. Ingeniería Directa. Aspectos económicos.

Unidad 10: Sistemas de Tiempo Real

Introducción a los sistemas de Tiempo Real. Tipos de Requisitos Temporales. Aplicaciones. Características de los Sistemas de Tiempo Real. Diseño. Métodos y Herramientas. UML. Redes de Petri. Propiedades. Análisis. Árbol de Cobertura. Matriz de Incidencia. Redes de Petri Temporizadas. Lenguajes de Programación.

Unidad 11: Seguridad para Ingenieros de Software

Definiciones. Dimensiones de Seguridad. Capas. Diferencias en seguridad de aplicaciones/infraestructura. Gestión. Seguridad y confiabilidad. Tipos de amenazas. Garantías de seguridad. Seguridad y organizaciones. Políticas de seguridad. Evaluación y gestión de riesgos. Evaluación preliminar, de diseño, operacional. Requisitos de seguridad. Casos de uso de seguridad. Diseño de sistemas seguros. Compromisos de Diseño. Evaluación de Riesgos de Diseño. Diseño Arquitectónico. Pautas de

Diseño para Ingeniería de Seguridad. Pautas para una Programación de Sistemas Seguros. Autenticación. Basada en el objeto. Basada en el conocimiento. Cracking. Biometría. Endurecimiento del código. Mitigación completa, fuerte y débil. Pruebas y aseguramiento de la seguridad.

VII - Plan de Trabajos Prácticos

Para cada unidad se deja disponible el material correspondiente a los contenidos de la unidad, las diapositivas de clase, el apunte teórico y su correspondiente trabajo práctico en el repositorio digital.

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

Laboratorio 2: Uso de Herramientas CASE en Ingeniería Reversa de Modelos de Datos / Objetos

Laboratorio 3: Herramienta CASE para la construcción de Modelos de Negocio.

Laboratorio 4: XML. XQuery.

Laboratorio 5: Herramientas CASE para la Ingeniería del Software Basada en Componentes.

Laboratorio 6: Construcción de Software bajo SOA

Laboratorio 7: Uso de Herramientas basadas en Redes de Petri para el modelado de Sistemas de Tiempo Real Distribuidos.

Laboratorio 8: Patrones de Diseño.

Práctico 1: Modelos Estáticos en UML. Ingeniería Directa e Ingeniería Inversa con Java.

Práctico 2: Modelos de Dominio.

Práctico 3: Modelos Dinámicos en UML. Ingeniería Directa e Ingeniería Inversa con Java.

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 teóricas y prácticas.
- 2.- Tener los prácticos, solicitados por la cátedra, aprobados, como método aplicado por la cátedra para la evaluación continua del estudiante.
- 3.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).
- 4.- Aprobar una evaluación parcial, o cualquiera de sus dos recuperaciones, con 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 la evaluación parcial, o cualquiera de sus dos recuperaciones, con una nota mayor o igual a 7 (siete).
- 3.- Aprobar un coloquio de carácter integrador oral o escrito con una nota mayor o igual a 7 (siete).

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

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

IX - Bibliografía Básica

- [1] [1] Roger S Pressman. Software Engineering: A Practitioner's Approach, 7/e. R. S. Pressman & Associates Inc. 2010.
- [2] [2] Booch, Rumbaugh, Jacobson. El Proceso de Desarrollo de Software Unificado. Addison-Wesley. 1999.
- [3] [3] Booch, Rumbaugh, Jacobson. The Unified Modeling Language User Guide, 2nd Edition. Addison-Wesley. 2005.
- [4] [4] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1995.
- [5] [5] Freeman, Eric, and Elisabeth Robson. Head First Design Patterns. O'Reilly Media, 2020.
- [6] [6] Ian Sommerville. Ingeniería de Software, 8 edición. Addison Wesley. 2005.

- [7] [7] Jeff Davis. SOA: Open Source (Spanish Edition). Anaya Multimedia. 2009
- [8] [8] Alan Burns, Andy Wellings. Sistemas de Tiempo Real y Lenguajes de Programacion - 3ra Edición. Pearson Educacion. 2005
- [9] [9] Priscilla Walmsley. XQuery. O'Reilly. 2007.
- [10] [10] Tadao Murata. Petri nets: Properties, Analysis and Applications. Proc. of the IEEE, 77(4), 1989.
- [11] [11] OpenUP/Basic. <http://epf.eclipse.org/wikis/openupsp/>.
- [12] [12] Security for Software Engineers. James Helfrich. CRC Press. 2019.
- [13] [13] Software engineering. Sommerville, Ian. Pearson. 2016.
- [14] [14] Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, David Svoboda, Java™ Coding Guidelines: 75 Recommendations for Reliable and Secure Programs, Addison Wesley, 2014
- [15] [15] Derek Fisher, Application Security Program Handbook: A guide for software engineers and team leaders, Version 2, Manning Publications, 2021
- [16] [16] Apuntes de Cátedra.

X - Bibliografía Complementaria

- [1] [1] Mark Grand. Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustred with UML. John Wiley & Sons Inc. 1998.
- [2] [2] UML Semantics. <http://www.omg.org> - UML Notation Guide. <http://www.omg.org>. Último acceso Agosto de 2017.
- [3] [3] Craig Larman. UML y Patrones: Introducción al análisis y diseño orientado a objetos. Prentice Hall. 1999.
- [4] [4] Sufyan Bin Uzayr. Software design patterns: the ultimate guide. CRC Press, 2023.
- [5] [5] Booch, Grady. Object-Oriented analysis and design with applications. The Benjamin/Cummings Publishing Company Inc. 1994.
- [6] [6] George T. Heineman, William T. Councill. Component-Based Software Engineering: Putting the Pieces Together. Addison-Wesley Professional. 2001.
- [7] [7] Dirk Krafziq, Karl Banke, Dirk Slama. Enterprise SOA. The Coad Series, Prentice Hall. 2004.
- [8] [8] SOAP - Messaging Framework, <http://www.w3.org/TR/soap12-part1/>. Último acceso Agosto de 2013.
- [9] [9] SOAP - Adjuncts, <http://www.w3.org/TR/soap12-part2/>. Último acceso Agosto de 2017.
- [10] [10] Web Services Description Language, <http://www.w3.org/TR/wsdl.html>. Último acceso Agosto de 2017.
- [11] [11] Doug Tidwell, James Snell, Pavel Kulchenko. Building Application Distributed Programming Web Services with SOAP. O'Reilly, First Edition. Diciembre de 2001
- [12] [12] Booch, Rumbaugh, Jacobson. The Unified Modeling Language Reference Manual, 2nd Edition. Addison-Wesley. 2005.
- [13] [13] Dan Pihone, Neil Pitman. UML 2.0 in a Nutshell. O'Reilly. 2005.
- [14] [14] Bernard Kolman. Álgebra Lineal con Aplicaciones en Matlab. Sexta edición. Prentice Hall. 2007.

XI - Resumen de Objetivos

Introducir al estudiante 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 Avanzados Orientados a Objetos en UML
 Proceso Unificado: Framework
 Patrones de Diseño
 Proceso Unificado: Análisis y Diseño.
 Ingeniería de la Información y Basada en Componentes.
 XML. Conceptos de Arquitecturas Basadas en Servicios (SOA).
 Reingeniería de Software
 Sistemas de Tiempo Real
 Seguridad para Ingenieros de Software

XIII - Imprevistos

Contacto con la cátedra: cabdelah@email.unsl.edu.ar

XIV - Otros