

UNIVERSIDAD NACIONAL DE RÍO CUARTO
FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

CARRERA/S: ANALISTA EN COMPUTACIÓN. PROFESORADO Y LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN.

PLAN DE ESTUDIOS: 1999

ASIGNATURA: Análisis y Diseño de Sistemas

CÓDIGO: 3303

DOCENTE RESPONSABLE: Mg. Marcela Daniele

EQUIPO DOCENTE: Lic. Marcelo Uva, Lic. Ariel Arsaute, Lic. Franco Brusatti, Prof. Daniela Solivellas

AÑO ACADÉMICO: 2020

REGIMEN DE LA ASIGNATURA: cuatrimestral

RÉGIMEN DE CORRELATIVIDADES: (para cursado)

<i>Aprobada</i>	<i>Regular</i>
	3325

CARGA HORARIA TOTAL: 180 TEÓRICO: 60 hs PRÁCTICAS y LABORATORIO: 120 hs

CARÁCTER DE LA ASIGNATURA: Obligatorio

A. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La asignatura se desarrolla en el 3er. año de las carreras de: Analista en Computación, Profesorado en Ciencias de la Computación y Licenciatura en Ciencias de la Computación.

B. OBJETIVOS PROPUESTOS

Con el cursado de esta asignatura se espera que el estudiante pueda:

- Comprender conceptos básicos de Ingeniería de Software, diferentes métodos de desarrollo de software y su evolución.
- Adquirir el conocimiento sobre las etapas del ciclo de vida de desarrollo de un software para obtener un producto de calidad. Instanciar en diferentes métodos de desarrollo de software.
- Conocer e internalizar el uso de lenguajes gráficos de especificación para el modelado de sistemas.
- Introducirse en conceptos de diseño, en la construcción de modelos genéricos para solucionar problemas con similares características, e instanciar problemas reales sobre dichos modelos.
- Utilizar los patrones de diseño adecuados para modelar soluciones de problemas recurrentes.
- Desarrollar habilidades y actitudes que favorezcan el trabajo colaborativo y el aprendizaje continuo.
- Comprender la importancia de la prueba del software, conocer y aplicar diferentes técnicas y herramientas.
- Ser capaz de elegir las herramientas de software más adecuadas, de acuerdo al tipo de sistema, con la suficiente formación para adaptarse a los constantes cambios, a las nuevas tecnologías y a los diversos ámbitos de aplicación de los sistemas de software

C. CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR

Conceptos básicos de Ingeniería de software. Modelos de desarrollo de Software. Metodologías de desarrollo de software: tradicionales y ágiles. Modelado orientado a objetos. Desarrollo dirigido por modelos. Modelado Gráfico: UML (Unified Modeling Language) Diagramas de Clases, de Objetos, de Interacción, de Casos de Uso, de Actividades, de Estados, de Componentes y de Despliegue. Ingeniería de requerimientos del Software. Etapas del ciclo de desarrollo de un software. Una metodología de desarrollo de software orientada a objetos: Proceso Unificado: Características, Conceptos, etapas del ciclo de vida. Captura de Requerimientos. Análisis, Diseño e implementación del Software. Una metodología de desarrollo Ágil: SCRUM. Diseño modular. Diseño detallado. Diseño OO. Patrones de Diseño. Prueba: Casos de prueba. Métodos de Prueba del Software. Prueba Funcional. Prueba Estructural.

D. FUNDAMENTACIÓN DE LOS CONTENIDOS

Los contenidos de la asignatura abordan los principales conceptos de ingeniería de software necesarios para la construcción de productos de software de calidad, entendiendo a este proceso como un trabajo ingenieril, basado en metodologías, teorías, técnicas y herramientas, que resultan necesarias para completar el ciclo de vida de desarrollo del software. En otras palabras, se concibe a la fabricación de software como el desarrollo de un producto de ingeniería que requiere planeamiento, análisis, diseño, implementación, prueba, implantación, mantenimiento y actualización.

E. ACTIVIDADES A DESARROLLAR

La asignatura se organiza en clases teóricas, prácticas y proyecto-taller. Los estudiantes asisten semanalmente entre 3 y 6 horas a las clases teóricas, entre 3 y 4 horas a las clases prácticas, y además, realizan un proyecto de software de forma grupal, y asisten al proyecto-taller semanal de entre 2 y 3 horas.

CLASES TEÓRICAS: Presencial en aula, aproximadamente 60 horas totales

CLASES PRÁCTICAS y TALLER: Presencial y semi presencial en laboratorio, 120 horas totales.

F. NÓMINA DE TRABAJOS PRÁCTICOS

Metodologías de desarrollo de software.

Análisis y Especificación de Requerimientos.

Descripción de funcionalidades con metodologías tradicionales PU (Casos de Uso) y metodologías ágiles SCRUM (User Stories)

Prueba Funcional. Prueba Estructural.

Ingeniería Directa. Ingeniería Inversa.

Patrones de Diseño. Implementación

Modelado gráfico de sistemas: UML. Diagramas de Clases.

Modelado gráfico de sistemas: UML. Diagramas de Objetos.

Modelado gráfico de sistemas: UML. Diagramas de Actividades.

Modelado gráfico de sistemas: UML. Diagramas de Estados.

Modelado gráfico de sistemas: UML. Diagramas de Secuencia

G. HORARIOS DE CLASES:

TEORICOS miércoles 13 a 16 hs y viernes de 9 a 12 hs (alternando con Taller)

TALLER: viernes 9 a 12 hs.

PRACTICOS Comisión 1: Martes-Jueves 10 a 12 hs. **Comisión 2:** Martes-Jueves 16 a 18 hs.

HORARIO DE CLASES DE CONSULTAS (estos horarios se pueden modificar de acuerdo a la coordinación con los estudiantes, en ese caso se publicarán los nuevos horarios)

Responsable: Lunes 12 hs.

Auxiliares: Martes 13 hs, Jueves 9 hs. Viernes 12 hs

H. MODALIDAD DE EVALUACIÓN:

EVALUACION PARCIAL: dos exámenes parciales escritos sobre el práctico de la materia, con sus respectivos recuperatorios. Desarrollo y presentación de un proyecto-taller de manera grupal.

EVALUACIÓN FINAL: escritos u orales sobre la teoría y práctica de la materia.

CONDICIONES DE REGULARIDAD: Aprobar los dos exámenes parciales y el proyecto final.

CONDICIONES DE PROMOCIÓN: No posee.

I. PROGRAMA ANALÍTICO

Unidad 1: Ingeniería de software. Introducción. Historia. Definición. El Ciclo de Vida del software. Atributos del software. Metodologías tradicionales. Metodologías ágiles. Modelos de desarrollo de Software: Construcción de Prototipos, Procesos Evolutivos, Incremental, Espiral, Ensamblaje de Componentes, Desarrollo Concurrente, Métodos Formales, Orientado a Objetos. Diseño por Contratos. Proceso Unificado. SCRUM. XP. Tipos de Sistemas.

Unidad 2: Ingeniería de requerimientos del Software. Requerimientos funcionales y no funcionales. Requerimientos de usuario y de sistema. Documentación de requerimientos -SRS. Reingeniería del Software: Ingeniería Inversa e Ingeniería Directa. Definición y documentación de Requerimientos en diferente metodologías ágiles y tradicionales. Proceso Unificado: Características, Conceptos, Artefactos, Actividades, Trabajadores, Flujos de Trabajo y Fases. Captura de Requerimientos. Modelo de Negocio. Procesos de negocio. Glosario de términos. Reglas de negocio. Modelo de Casos de Uso. Modelo de Análisis. Clases de análisis. SCRUM: Características. Conceptos. Historias de usuario. Roles.

Unidad 3: Modelado de sistemas. Principios. Técnicas de Descripción de Requerimientos. Modelado estructural o estático y modelado dinámico. MOO. UML: Estereotipos. Valores etiquetados. Restricciones.

Diagramas de clases: Clases: atributos, operaciones y responsabilidades. Relaciones. Interfaces. Paquetes. Diagrama de objetos: Objetos, Instancias, enlaces. Diagrama de Interacción: escenarios. Diagrama de casos de uso. Diagrama de actividades. Diagrama de estados. Eventos y señales. Modelado de nodos y componentes.

Unidad 4: Prueba de software. Fundamentos teóricos. Principios de la prueba. Métodos de Prueba. Prueba funcional. Análisis del valor límite. Clases de equivalencia. Tablas de decisión. Prueba estructural. Cobertura de sentencia. Cobertura de arco. Cobertura de condición. Cobertura de camino. Complejidad ciclomática. Herramientas.

Unidad 5: Diseño de Software. Conceptos de diseño. Abstracción. Refinamiento. Modularidad. Arquitectura del software. Estructura de programa. Ocultamiento de información. Diseño modular. Cohesión y acoplamiento. Diseño detallado. Diseño OO. Proceso unificado: Modelo de diseño: artefactos, actividades y trabajadores. Modelo de implementación. Nodos y componentes. Modelo de prueba. Casos de prueba. Procedimiento de prueba. Plan de prueba.

Unidad 6: Patrones de diseño. Introducción. Conceptos. Descripción. Utilización. Problema. Solución. Consecuencia. Catálogo de patrones de diseño. Categoría de patrones: creacionales, estructurales y de comportamiento.

CRONOGRAMA DE CLASES Y PARCIALES

TEÓRICOS	PRÁCTICOS
Presentación. Introducción. Conceptos básicos de IS, 4 Ps. Metodologías. Ciclo de vida en el desarrollo de un software	1. UML. Diagramas de Clases. 8 hs 2. P-T Análisis y Especificación de Requerimientos 3. UML. Diagramas de Objetos. 3 hs 4. Prueba de software. 8 hs 5. Práctica con ejercicios de repaso para el 1er Parcial 6. Especificación y diagrama de casos de uso. 2 hs 7. UML. Diagrama de actividades. 4 hs 8. UML. Diagrama de Estados. 4 hs 9. UML. Diagrama de Secuencia. 3 hs 10. Diseño de software. Patrones de Diseño. 8 hs 11. Práctica con ejercicios de repaso 2º parcial.
Modelado de sistemas. UML. Introducción. Clases. Relaciones. Diagramas de Clases.	
UML. Introducción. Clases. Relaciones. Diagramas de Clases. Diagramas de Objetos. Ingeniería Inversa y Directa.	
Análisis del Problema. SRS c/ más de un método. Ingeniería de requerimientos del Software. Requerimientos funcionales y no funcionales.	
Prueba. Funcional y Estructural. Modelo de Prueba.	
UML. Diagramas de Casos de Uso. UML. Diagramas de Actividades. Diagramas de Estados	
Metodologías de desarrollo: Metodología de desarrollo OO: Proceso Unificado. Metodología Ágil: SCRUM	
Captura de Requerimientos: M Negocio + M Casos de Uso. Plantillas Genéricas p/desc de CU	
Análisis y Diseño de Software. Diseño Funcional. Diseño de Casos de Uso. Diseño Detallado. PU: Análisis. Diseño.	
UML. Diagramas de Interacción: Diagrama de Secuencia.	
Patrones de Diseño. Plantillas genéricas para A y Diseño.	
Implementación. Conceptos. Nodo y Componente. Modelo de Implementación.	
UML: Diagrama de Despliegue. Diagrama de Artefactos.	

Miércoles 22/4	PRIMER PARCIAL
Miércoles 03/6	SEGUNDO PARCIAL
Miércoles 12/6	RECUPERATORIO 1º PARCIAL y 2º PARCIAL
Miércoles 10/6	PRESENTACION FINAL DEL PROYECTO

BIBLIOGRAFÍA

1. Pressman, Roger. Software Engineering: A Practitioner's Approach. 8va edition. McGraw Hill. 2015.
2. Sommerville Ian. Ingeniería del Software, 7th edition, Addison Wesley. Pearson Education, 2005. ISBN: 978-84-7829-074-1
3. Ghezzi Carlo, Jazayeri M., Mandrioli D.. Fundamentals of Software Engineering. Prentice Hall, 1991.
4. Kendall y Kendall. Análisis y Diseño de Sistemas. Pearson Education. 2005.
5. Pankaj Jalote. An Integrated Approach to Software Engineering. Springer. 2005.
6. Booch Grady, Rumbaugh J., Jacobson Ivar. The Unified Modeling Language. Addison Wesley. 1999.
7. Booch Grady. Object-oriented analysis and design with applications. Benjamin Cummins, 1994. Versión español: Análisis y Diseño Orientado a Objetos: Con Aplicaciones. Addison Wesley. 1996.
8. Paul Ammann and Jeff Offutt. Introduction to Software Testing. Cambridge University Press. Second Edition. 2016
9. Paul C. Jorgensen. Software Testing: A Craftsman's Approach, Auerbach Publications; Fourth Edition. 2013
10. Gamma Erich, Helm Richard, Johnson Ralph, Vlissides John. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. 1995.
11. Grand Mark. Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustrated with UML. John Wiley & Sons Inc. 1998.
12. Highsmith Jim. Agile Software Development Ecosystems. Addison-Wesley. 2002. ISBN: 0 201-760-13-6.
13. Jacobson Ivar, Booch Grady, Rumbaugh James. The Unified Software Development Process. Addison Wesley. 1999.
14. Meyer Bertrand. Object Oriented Software Construction. Prentice Hall. 1997.
15. OMG. Object Management Group. Unified Modeling Language Specification. <http://www.omg.org>.
16. Pattern-Oriented Software Architecture. Volume 1: A System of Patterns. [Frank Buschmann](#) , [Regine Meunier](#) , [Hans Rohnert](#) , [Peter Sommerlad](#) , [Michael Stal](#) .
17. Rational Unified Process. <http://www.rational.com/rup/>
18. www.agilemanifesto.org - <http://www.agile-spain.com/>

NOTA: Se utilizan otros materiales de lectura complementarios extraídos de diferentes sitios web, manuales de las herramientas utilizadas, entre otros. En cada una de las actividades se indican los capítulos de lectura del material presentado en la bibliografía y se ponen a disposición todo otro material digital utilizado.

Profesora Responsable: Mg. Marcela Daniele. _____