



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

(Programa del año 2021)

### I - Oferta Académica

Materia	Carrera	Plan	Año	Período
INGENIERIA DE SOFTWARE II	ING. INFORM.	026/1	2- 2021	2° cuatrimestre
		08/15		

### II - Equipo Docente

Docente	Función	Cargo	Dedicación
ABDELAHAD, CORINA NATALIA	Prof. Responsable	P.Adj Exc	40 Hs
RIESCO, DANIEL EDGARDO	Prof. Colaborador	P.Asoc Exc	40 Hs
BERNARDIS, HERNAN	Responsable de Práctico	JTP 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	2 Hs	3 Hs	7 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
23/08/2021	26/11/2021	14	90

### IV - Fundamentación

Dar las bases teóricas y prácticas que permiten al Ingeniero de Software aplicar un método 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.

### VI - Contenidos

#### 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.

#### 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.

### **UUnidad 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. 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.

## **VII - Plan de Trabajos Prácticos**

Debido a la pandemia del COVID-19, el presente es un programa con Modalidad Virtual o Mixta dependiendo las situaciones epidemiológicas 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 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: Herramientas CASE para la Ingeniería del Software Basada en Componentes.

Laboratorio 5: 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 los laboratorios y en las prácticas.

## **VIII - Regimen de Aprobación**

Considerando que este Programa se desarrolla con Modalidad Virtual o Mixta dependiendo las situaciones epidemiológicas, se tuvieron en cuenta las siguientes consideraciones:

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 70% de asistencia, la misma se considera con las entregas de los ejercicios solicitados en cada trabajo práctico.
- 2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).
- 3.- Aprobar el 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.- Tener como mínimo un 80% de asistencia, la misma se considera con las entregas de los ejercicios solicitados en cada trabajo práctico.
- 2.- Presentación y aprobación del proyecto integrador de laboratorio con nota mayor o igual a 7 (siete).
- 3.- Aprobar una evaluación, o cualquiera de sus dos recuperaciones, con nota mayor o igual a 7 (siete).
- 4.- Aprobar una prueba final integradora con una nota mayor o igual a 7 (siete), la cual podrá ser oral y/o escrita.

Las evaluaciones se realizarán en un horario establecido. Se colocará el enunciado en aulas virtuales en caso de que sea virtual 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. En caso de que la evaluación sea presencial será en el horario establecido de clase y no será necesario el coloquio.

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] Software Engineering: A Practitioner's Approach, 7/e, Roger S Pressman, R. S. Pressman & Associates, Inc. 2010.
- [2] El Proceso de Desarrollo de Software Unificado. Booch, Rumbaugh, Jacobson. Addison-Wesley, 1999.
- [3] The Unified Modeling Language User Guide, 2nd Edition. Booch, Rumbaugh, Jacobson. Addison- Wesley, 2005.
- [4] Design Patterns: Elements of Reusable Object-Oriented Software. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison-Wesley. 1995.
- [5] Object-Oriented analysis and design with applications. Booch, Grady. The Benjamin/Cummings Publishing Company Inc. 2007.
- [6] Software Engineering, Ian Sommerville, Addison Wesley; 8 edition, 2006
- [7] OpenUP/Basic, <http://epf.eclipse.org/wikis/openupsp/>, ultimo acceso agosto 2016.

## **X - Bibliografía Complementaria**

- [1] Patterns in Java. Volume 1. A Catalog of Reusable Design Patterns Illustred with UML. Mark Grand. John Wiley & Sons Inc. 1998.
- [2] UML Semantics. <http://www.omg.org>
- [3] UML Notation Guide. <http://www.omg.org>
- [4] UML y Patrones: Introducción al análisis y diseño orientado a objetos. Craig Larman, Prentice Hall, 1999.
- [5] The Unified Modeling Language Reference Manual, 2nd Edition. Booch, Rumbaugh, Jacobson. Addison-Wesley, 2005.
- [6] Component-Based Software Engineering: Putting the Pieces Together, George T. Heineman, William T. Council, Addison-Wesley Professional, 2001.

## **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 las distintas etapas del proceso de desarrollo.

## **XII - Resumen del Programa**

Requerimientos  
Modelos Avanzados Orientados a Objetos en UML  
Proceso Unificado: Framework  
Patrones de Diseño  
Proceso Unificado: Análisis y Diseño.

### **XIII - Imprevistos**

Ante las medidas nacionales tomadas por la pandemia del COVID-19, la materia podrá ser dictada con modalidad mixta, es decir presencial y no presencial, según lo establezca la Facultad. Cuando tengamos clases no presenciales será mediante las plataformas Moodle y meet. El seguimiento continuo se realiza mediante correo electrónico.

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

Según Resolución 1404 el Segundo Cuatrimestre de 2021 posee 14 semanas. A los efectos de que se impartan todos los contenidos y se respete el crédito horario establecido en el Plan de estudios de la carrera para la asignatura, se establece que se de cómo máximo 7hs por semana distribuidas en teorías, prácticos de aula, laboratorios, trabajos tutoriales, consultas, hasta completar las 90hs.

### **XIV - Otros**