

**Taller de Tecnologías
de Producción de Software****Carrera/ Plan:***Analista Programador Universitario Plan 2021/Plan 2015***Opción E
Taller de Desarrollo de Software Guiado
por los Requerimientos****Año:** 3°**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:**

Diseño de bases de datos

Algoritmos y Estructuras de Datos

Introducción a los Sistemas Operativos

Orientación a Objetos 1

Taller de lecto - comprensión y traducción en inglés

Ingeniería de Software 2

Coordinador: Gustavo Rossi**Profesor/es:** Leandro Antonelli**Hs. Semanales teoría:**3**Hs. Semanales práctica:**3**Año 2025****FUNDAMENTACIÓN**

La especificación de requerimientos es una de las etapas que más impacto tienen en el resto de las etapas del ciclo de vida de desarrollo del software. Si no se cuenta con los requerimientos adecuados al inicio del desarrollo no es posible realizar una buena planificación del proyecto y tampoco se puede construir el producto que satisfaga las necesidades del cliente. En este contexto, los proyectos de software tienen alta probabilidad de que se demoren o de construir un producto que no responda a las necesidades.

Si bien las metodologías ágiles como Scrum intentan mitigar falencias en los requerimientos a través de un desarrollo iterativo e incremental, de todas formas es necesario que el product-owner tenga el conocimiento y las habilidades necesarias para especificar requerimientos, y esto no siempre es posible. En un proceso de desarrollo tradicional al estilo RUP, la criticidad de los requerimientos es aún más, puesto que se necesita producir una especificación de requerimientos de cierta calidad.

Especificar requerimientos implica cumplir con varias etapas: relevar, analizar, priorizar, modelar, validar y gestionar. Sin importar el proceso de desarrollo, todas estas etapas se desarrollan en alguna medida y con cierta formalidad

Una de las incumbencias del título de Analista Programador Universitario (<http://www.info.unlp.edu.ar/index.php/analista-programador-universitario>) es "Participar en el relevamiento y análisis de los procesos funcionales de una Organización, con la finalidad de que se diseñen los Sistemas de Información asociados, así como los Sistemas de Software que hagan a su funcionamiento.". Por lo cual, es importante la formación de los profesionales en temas de requerimientos. Puesto que al no estar capacitados para realizar esta tarea (Capturar y describir requerimientos), esta tarea termina siendo realizada por un analista no informático, ocasionando ciertos trastornos en el desarrollo.

OBJETIVOS GENERALES

Introducir a los alumnos en un esquema de Organización de Producción de Software utilizando requerimientos como pilar fundamental para el desarrollo. Principalmente para estimar, diseñar y testear.

Fomentar la práctica del alumno en un esquema de trabajo similar a los que se utilizan en ámbitos reales, a través de un curso tipo taller en que el alumno experimentará reuniones de trabajo, deberá negociar requerimientos y resolver conflictos.

Ofrecer a los alumnos alternativas tecnológicas de actual utilización en el mercado como lo son los lenguajes de programación orientados a objetos y tecnologías de modelados RUP y ágiles.

RESULTADOS DE APRENDIZAJE

Se espera que finalizado el curso el alumno haya asimilado la experiencia de un proyecto de desarrollo realista y esto le permita desarrollar la capacidad crítica para poder evaluar situaciones objetivamente y poder así tomar la mejor decisión por el éxito del proyecto.

COMPETENCIAS

En el curso se trabajaran principalmente dos competencias: análisis y diseño / implementación. A continuación se indica el detalle por cada competencia.

Análisis

- utilizar una serie de técnicas con las que identificar las necesidades de problemas reales, analizar su complejidad y evaluar la viabilidad de las posibles soluciones mediante técnicas informáticas
- describir un determinado problema y su solución a varios niveles de abstracción

Diseño e Implementación

- definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos
- describir las fases implicadas en distintos modelos de ciclo de vida con respecto a la definición, construcción, análisis y puesta en marcha de nuevos sistemas y el mantenimiento de sistemas existentes
- elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales así como aplicaciones emergentes
- describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador
- aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos

CONTENIDOS MINIMOS (de acuerdo al Plan de Estudios)

- Introducir un ambiente de desarrollo de software estandarizado (con herramientas integradas que den una visión homogénea y estandarizada de las aplicaciones, su interface gráfica, el acceso a las bases de datos y la interconexión entre aplicaciones), enfocado a un organismo o "clase" de empresa usuaria.
- Practicar cómo usar el ambiente de desarrollo y las diferencias con el ambiente de producción, ilustrando la metodología organizacional del pasaje de desarrollo a producción.
- Practicar con documentación estandarizada (por ej. Casos de Uso con UML) mostrando como se pasa de una especificación al código ejecutable.
- Ejemplificar la actividad del tester de aplicaciones. Metodología de trabajo y ambiente de prueba (diferencia con los otros ambientes)
- Plantear el proceso estandarizado de desarrollo de software en una tecnología de uso en el mercado. Rol de la documentación en cada etapa.
- Plantear el desarrollo de una solución a un problema real y que ilustre todas las problemáticas antes descriptas
- Describir cuales son las principales características de un proceso de desarrollo de software con calidad (introduciendo los principios básicos de CMM o CMMI)

PROGRAMA ANALÍTICO

Unidad 1. Ingeniería de Software, Gestión de proyectos e Ingeniería de Requerimientos.

Unidad 2. Ciclos de vida. Etapas del ciclo de vida. Productos y procesos.

Unidad 3. Proceso de desarrollo. Gestión de la configuración de productos. Ambientes de desarrollo, prueba y producción.

Unidad 4. Procesos de la ingeniería de Requerimientos: Relevamiento, modelización, priorización y validación.

Unidad 5. Productos de la ingeniería de Requerimientos: Glosarios, User Stories y Use Cases.

Unidad 6. Productos del diseño: diagrama de clases, diagrama de colaboración, máquinas de estados.

Unidad 7. Diseño de casos de prueba. Pruebas de caja blanca. Pruebas de caja negra.

Unidad 8. Transformación de los requerimientos a los distintos modelos.

Unidad 9. Calidad en el proceso y en los productos. CMMi.

BIBLIOGRAFÍA

Jason, Jennifer: *Agile project management: kanban, scrum, kaizen*, ISBN 9781719949187 (2018).

Dick, J., Hull, E., Jackson, K., *Requirements Engineering 4th edition*, Springer (2017).

Ruhe, G., Wohlin, C., *Software Project Management in a Changing World*, Springer (2014)

Robert K. Wysocki: *Effective Project Management: traditional, agile, extreme 7th edition*, John Wiley & Sons (2014)

Sommerville, I., *Software Engineering*, Addison-Wesley, Harlow, England (2010)

Aybüke Aurum, Claes Wohlin, *Engineering and Managing Software Requirements*, Springer-Verlag Berlin Heidelberg (2005)

Cohn, M.: *User Stories Applied*, Addison Wesley, ISBN 0-321-20568-5 (2004)

Cockburn, A.: *Writing Effective Use Cases*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN 0-201-70225-8 (2001)

Loucopoulos, P., Karakostas, V., *System Requirements Engineering*, McGraw-Hill (1995)

IEEE, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE Std 830-1998 (Revision of IEEE Std 830-1993)

Frederick P. Brooks, Jr., *The mythical man-month*, Addison-Wesley Publishing Company Reading, Massachusetts (1975)

Antonelli, L., Leite, J. C. S. P., Oliveros, A. Rossi, G.: "Specification Cases: a lightweight approach based on Natural Language", *Workshop in Requirements Engineering (WER)*, Brasilia, Brasil, Agosto 23 – 27 (2021)

Bimonte, S., Antonelli, L., Rizzi, S., "Requirements-driven data warehouse design based on enhanced pivot tables", *Requirements Engineering Journal*, DOI: 10.1007/s00766-020-00331-3, REEN-D-19-00051R3, submitted: 8 August 2019, Accepted: 28 March 2020, © Springer-Verlag London Ltd., ISSN 0947-3602 part of Springer Nature 2020.

Antonelli, L., Rossi, G., Leite, J. C. S. P., Oliveros, A.: *Deriving requirements specifications from the application domain language captured by Language Extended Lexicon*. WER 2012

Antonelli, L., Rossi, G., Leite, J. C. S. P., Oliveros, A.: *Buenas prácticas en la especificación del dominio de una aplicación*. WER 2013

METODOLOGÍA DE ENSEÑANZA

El curso es un taller, en donde si bien se presentan conceptos teóricos, el objetivo principal es que los alumnos hagan experiencia real en el desarrollo de software con énfasis en relevamiento y especificación de requerimientos. En el curso, habrá dos tipos de encuentros. Por un lado habrá clases teóricas tradicionales, y por otro lado, habrá encuentros en que se simulen reuniones con clientes. En estas reuniones, los alumnos pondrán en práctica técnicas y herramienta propias de la disciplina de requerimientos, por lo cual, desarrollarán las capacidades de análisis. Luego, los alumnos deberán diseñar e implementar la aplicación, por lo cual, se pondrán en práctica las competencias de diseño e

implementación. El curso finaliza con un coloquio / demo, en la cual, los alumnos presentan el trabajo realizado.

Dado que el objetivo principal es ofrecer una experiencia real en el desarrollo de software y considerando que el desarrollo distribuido y encuentros virtuales es moneda corriente en la actualidad, se coordinará con los alumnos la naturaleza de los encuentros (presenciales o virtuales).

Siguiendo la misma línea de brindar una experiencia realista de desarrollo de software, y dado que en los proyectos suelen sumarse stakeholders durante alguna fase específica del proyecto, en el curso se contará con Profesores invitados, los cuales presentarán algún método, técnica o herramienta, con el fin de enriquecer el contenido del curso, a la vez de requerir la incorporación de tal método, técnica o herramienta (y su consiguiente evaluación) al proceso de desarrollo.

EVALUACIÓN

La evaluación se realizará a través del desarrollo de un software en una situación realista y de actividades prácticas a modo de ejercitación de los conceptos teóricos.

El desarrollo de software será un desarrollo realista dado que algunos docentes del curso oficiarán de clientes para brindar los requerimientos y calificar el software desarrollado determinando si cumple o no con sus necesidades. El desarrollo de software constará de 3 partes: (i) documentación, (ii) aplicación, y (iii) una presentación / coloquio del trabajo realizado. Se evaluará la calidad de las distintas partes e instancias, y no solamente el cumplimiento con los requerimientos.

La documentación incluirá distintos artefactos del ciclo de vida: especificación de requerimientos, mockups, casos de prueba, arquitectura, y diseño entre otros. Los docentes proveerán un template del documento como así también una guía para describir cada artefacto solicitado.

El desarrollo estará continuamente monitoreado de forma de encaminar y sobrellevar cualquier eventualidad o desvío respecto de los planes originales. Por otro lado, el software a construir tendrá diferentes niveles de requerimientos (por ejemplo: obligatorios, deseables y adicionales) que podrán ser negociados (o no) con los equipos en función del tiempo, los miembros del grupo, etc.

La presentación o coloquio se desarrollará entre los alumnos y los docentes, donde los alumnos deben mostrar conocimiento (en cuanto a participación, como así también en cuanto a las tecnologías y herramientas) del proyecto realizado y del informe elaborado, integrando los conceptos del curso, a la vez de que deben mostrar un análisis crítico. Dado que el proyecto (y el informe) son seguidos desde su comienzo, hay una sola instancia de coloquio.

Además del proyecto, durante el curso se realizarán actividades prácticas sobre distintos conceptos que podrán ser para incluir en el proyecto de desarrollo o simplemente para practicar su aplicación.

La calificación del curso se calculará a partir del trabajo realizado en la documentación, en el desarrollo, el desempeño en el coloquio y las actividades prácticas durante el curso. A partir de todo este trabajo, la cátedra elaborará una nota final para cada alumno.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	8/sep	Introducción. Importancia y rol de los requerimientos.
2	15/sep	Conocimiento del dominio. Aspectos dinámicos.
3	22/sep	Escenarios.
4	29/sep	Lenguaje en el dominio. Diccionarios, glosarios y ontologías.
5	6/oct	Glosario LEL.
6	13/oct	Especificación de Requerimientos. Requerimientos funcionales. User Stories. Use Cases
7	20/oct	Mockups. Requerimientos no funcionales. Casos de prueba.
8	27/oct	Documentos de especificación de requerimientos. Suposiciones, restricciones y dependencias
9	3/nov	Procesos de desarrollo.

Evaluaciones previstas	Fecha
Coloquio	1/dic

Contactos de la cátedra

- **Mail (obligatorio):** leandro.antonelli@lifa.info.unlp.edu.ar
- **sitio WEB:**
- **plataforma virtual:**
- **Otros:**

Firma del/los profesor/es

Leandro Antonelli