

**Taller de Tecnologías
de Producción de Software****Opción E****Taller de Desarrollo de Software Guiado
por los Requerimientos****Año 2021****Carrera/ Plan:**

Analista Programador Universitario Plan 2015/Plan 2007

Año: 3°**Régimen de Cursada: Semestral****Carácter (Obligatoria/Optativa): Obligatoria****Correlativas:**

Introducción a las 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:6****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 proceso de desarrollo de software estandarizado.*
- Documentación del desarrollo de software: requerimientos, diseño, código y casos de prueba.*
- Trazabilidad de los artefactos del desarrollo de software, tanto en su evolución desde los requerimientos al código, como en su versionado.*
- Calidad en el proceso de desarrollo: principios básicos de 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

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

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

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

Robert K. Wysocki, Effective Project Management: Traditional, Agile, Extreme, Wiley (2011).

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

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)

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

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

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.

EVALUACIÓN

La evaluación se realizará a través de 3 instrumentos: (i) un proyecto, (ii) un informe y (iii) un examen oral.

El Proyecto consiste en el desarrollo de un software en una situación realista. Es decir, se contará con un docente que oficiará de cliente, a la vez que otro docente oficiará de responsable de los aspectos metodológicos. De esta forma, un docente negociará los requerimientos y otro negociará el proceso de desarrollo y el informe que deben realizar del desarrollo. El proyecto estará continuamente monitoreado de forma de encaminar y sobrellevar cualquier eventualidad. Por otro lado, el software a construir tendrá 3 niveles de requerimientos (obligatorios, deseables y adicionales), donde los requerimientos obligatorios son condición necesaria para aprobar.

El informe consiste en una documentación que incluye la especificación de requerimientos, como así también diseño y casos de prueba. Los docentes proveen un template del documento como así también

una guía para describir cada artefacto solicitado. Nuevamente, el informe también posee secciones obligatorias que son necesarias para aprobar el curso, mientras que otras son adicionales.

El examen oral es un coloquio entre los alumnos y los docentes, donde los alumnos deben mostrar conocimiento del proyecto realizado y del informe elaborado, integrando los conceptos del curso, a la vez que deben mostrar un análisis crítico. Dado que el proyecto (y el informe) son seguidos desde su comienzo, hay una sola instancia de coloquio.

La calificación del curso se calcula de la siguiente manera: 30% calificación del proyecto + 30% calificación del informa + 40% coloquio.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	1/sep	Introduccion. Diccionarios, glosarios y ontologías.
2	8/sep	Glosario LEL
3	15/sep	Especificacion de Requerimientos.
4	22/sep	Use Cases
5	29/sep	Mockups
6	6/oct	Requerimientos no funcionales
7	13/oct	Casos de prueba
8	20/oct	Suposiciones, restricciones y dependencias
9	27/oct	Procesos de desarrollo

Evaluaciones previstas	Fecha
Coloquio	24/Nov

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

Firma del/los profesor/es



Leandro Antonelli