

# Developing Competencies for Software Requirements Analysis Through Project Based Learning

René Noël<sup>1</sup>, Roberto Muñoz<sup>1,2</sup>, Carlos Becerra<sup>1</sup>, Rodolfo Villarroel<sup>2</sup>

<sup>1</sup> Universidad de Valparaíso, Chile

<sup>2</sup> Pontificia Universidad Católica de Valparaíso, Chile

{rene.noel, roberto.munoz, carlos.becerra}@uv.cl, rodolfo.villarroel@ucv.cl

**Abstract**— The course “Software Analysis Methodologies” of Computer Science career at the University of Valparaíso has among its objectives to contribute to the development of competencies related to software requirements analysis. Traditionally, through the course the students applied methods and techniques for software requirements analysis in a project with real stakeholders. Nonetheless, this activities were performed outside the classroom, not allowing the teacher to assess the performance of critical competencies, not only related to the application of software engineering knowledge, but also related to the articulation of skills such as obtaining requirements through interviews, information organization and modeling, peer communication, analysis and validation using technical language, conflict management with stakeholders, and requirements validation with stakeholders using nontechnical language. This paper presents the approach of project-based learning with work activities with real stakeholders in classes. As result it has managed to continuously assess the development of skills, a pass rate of 96%, and the achievement of learning outcomes validated by a real counterpart (stakeholder), and also by the teacher and teammates. Also, this initiative shows the articulation of two Institutional Improvement Projects funded by the Ministry of Education, for STEM teaching methodologies, and for the application of Information Technology in the classroom.

**Keywords**—Requirements Analysis; Project Based Learning; Software Engineering Education.

## I. INTRODUCCIÓN

El Análisis de requerimientos de software es una disciplina dentro de la Ingeniería de Software que permite partir de una necesidad de información automatizada y llegar a una descripción técnica del software que deberá construirse [1]. Los requerimientos pueden obtenerse de una variedad de fuentes, siendo una de las más frecuentemente usadas, y a la vez complejas, las entrevistas con usuarios, expertos del dominio del problema y clientes (quienes gestionan los recursos para desarrollar el software). El Ingeniero de requerimientos debe manejar conocimientos y capacidades tales como conocer y aplicar técnicas de modelamiento de requerimientos, pero además articularlos con capacidades de comunicación, habilidades sociales como la empatía y la

negociación, y capacidad de trabajo en equipo, todo dentro de un marco de recursos acotados e implicancias reales, como todo proyecto de Software. Preparar a los estudiantes para enfrentar este escenario es el objetivo de la asignatura Metodología de Análisis de la carrera de Ingeniería Civil Informática de la Universidad de Valparaíso. Es un desafío complejo, pues tradicionalmente los problemas y restricciones abordados no permiten al estudiante desplegar las capacidades descritas, ni al profesor evidenciarlas o entregar retroalimentación del desempeño de los estudiantes.

El presente trabajo expone el desarrollo de la asignatura con un nuevo enfoque de metodologías activas de enseñanza, centrada en el desarrollo de un Aprendizaje basado en equipos, cuyas actividades se llevan a cabo dentro de la sala de clases. En las secciones siguientes se presenta un marco referencial de conceptos para entender la necesidad de incorporar docencia activa a la asignatura, el problema educativo y consideraciones iniciales para implementar aprendizaje en proyectos, luego la ejecución del curso bajo esta metodología, para finalizar con las conclusiones y lecciones aprendidas.

## II. MARCO TEÓRICO

### A. Competencias y niveles cognitivos

El aprendizaje basado en competencias busca la formación de profesionales íntegros soportado por un modelo de aprendizaje constructivista, que cumple con las necesidades que la sociedad exige y se encuentra dentro del marco del ser, saber y hacer. Según [2], una competencia es un saber actuar complejo que se apoya sobre la movilización y la utilización eficaz de una variedad de recursos tanto internos como externos. La preparación de estas actuaciones complejas requiere que los estudiantes no sólo desarrolle habilidades para conocer comprender y recordar contenidos, sino también su aplicación, capacidad de analizar y resolver problemas, crear soluciones y evaluarlas. Dentro de las competencias se pueden distinguir algunas genéricas, como las competencias sociales (también conocidas como habilidades blandas o *soft skills*), y otras específicas de la disciplina, como las asociadas al pensamiento científico o el diseño de soluciones ingenieriles. Estas habilidades son caracterizadas como distintos niveles cognitivos según [3], que indica una escala progresiva de estos

niveles y verbos asociados a su manifestación en actividades de quien desarrolla el aprendizaje. Para que el estudiante desarrolle estas habilidades y evidencie los indicadores de logro, es necesario que su rol en el aula pase de ser de un pasivo receptor de conocimientos, a ser protagonista de su aprendizaje. Las estrategias de aprendizaje activo [4] permiten poner al estudiante en el centro del aprendizaje. Una de estas técnicas es el Aprendizaje basado en proyectos, o *Project Based Learning* (PBL) [5].

### B. Aprendizaje Basado en Proyectos

El aprendizaje basado en proyectos consiste en involucrar a los estudiantes en la resolución de un problema con implicancias reales, que requiere movilizar saberes de distintos ámbitos y competencias genéricas y específicas. Requiere un aprendizaje primordialmente desarrollado en equipos de trabajo, idealmente con integración interdisciplinar. La motivación para llevar a cabo este tipo de aprendizajes es la idea de que solo un proyecto con la complejidad y escala de resolver un problema real permite que los estudiantes ejerciten competencias específicas y genéricas en contextos y niveles similares a los que enfrentarán en el mundo laboral. PBL requiere definir consideraciones iniciales y una serie de pasos que serán descritos en la sección III y IV.

### C. Proceso de análisis de requerimientos de software

El Proceso de análisis de requerimientos de software o Ingeniería de requerimientos [1] es una disciplina propia de la Ingeniería de Software, una de las áreas de la Ingeniería Informática. El objetivo de esta disciplina es poder obtener los requerimientos que definirán un software, que luego será diseñado, construido (programado), probado y posteriormente utilizado por los usuarios finales. En el proceso de análisis de requerimientos se reconocen cuatro actividades, descritas en la figura 1.



Fig. 1. Actividades del proceso de análisis de requerimientos.

En el proceso, el profesional informático toma el rol de Ingeniero de Requerimientos, responsable de ejecutar las actividades anteriores. Una de los medios más tradicionales para realizar el levantamiento de requerimientos es la entrevista con contrapartes, donde se puede distinguir a los Usuarios (quienes utilizarán finalmente el sistema), los Expertos de negocio (quienes conocen el contexto organizacional y los procesos que el software va a automatizar o soportar), y los Clientes (aquellos que toman decisiones estratégicas y administran recursos del proyecto). En este escenario, el rol del

Ingeniero de Requerimientos debe ser capaz de enfrentar los siguientes desafíos:

- En la actividad de Levantamiento de Requerimientos, debe ser capaz de entrevistar a las contrapartes, reconociendo su lenguaje propio de su organización y negocio. Además debe poder relacionar los elementos presentados por cada contraparte, y establecer relaciones entre ellos. Las habilidades comunicacionales y sociales son fundamentales en esta actividad.
- En la actividad de Análisis, el ingeniero debe ser capaz de identificar contradicciones entre las contrapartes, o inconsistencias entre lo expresado por estas y la definición original del proyecto. Debe ser capaz de distinguir si la contraparte está escribiendo el problema o está sesgando su visión a una solución preconcebida (que sería incorrecto), e identificar carencia de detalle en la información que pueda acarrear problemas a la hora de construir el software. Las habilidades de pensamiento lógico y capacidad de análisis sin críticas en esta actividad.
- En la actividad de Especificación, el ingeniero debe ser capaz de organizar la información recopilada mediante modelos de distinto nivel de abstracción y representando distintas perspectivas del sistema. Es crítica la capacidad de pensamiento abstracto para realizar generalizaciones y refinamientos concretos de construcciones abstractas. Adicionalmente se requieren competencias ingenieriles asociadas al conocimiento y aplicación de lenguajes de modelamiento de software, en particular, el Lenguaje Unificado de Modelamiento, UML [6].
- En la actividad de Validación, el ingeniero debe asegurarse de la correctitud y completitud de sus modelos (el análisis está correctamente realizado), y que los requerimientos especificados resuelven el problema de fondo de la contraparte (el análisis aborda el problema correcto). En esta instancia es importante nuevamente el pensamiento lógico, pero también las habilidades de comunicación y sociales, incorporándose además la capacidad de negociación, en caso de surgir conflictos de alcance (la contraparte estima que los requerimientos son insuficientes para resolver su problema), que podrían traducirse en trazar la adición de nuevas funcionalidades dejando de lado otras menos importantes.

Una de las características importantes de estas actividades es que no se dan secuencialmente, sino en un proceso dinámico de interacción entre el ingeniero y la contraparte, esto requiere además que el ingeniero esté consciente en cada minuto de qué tipo de conversación (o discusión está teniendo), es decir, desarrollo meta cognitivo de su propio actuar.

### D. Asignatura Metodología de Análisis

La asignatura de Metodología de Análisis se dicta en el tercer año de la carrera de Ingeniería Civil Informática de la Escuela de Ingeniería civil Informática. Si bien el programa de

estudios hasta el momento no compromete competencias, la versión innovada de esta asignatura, que entrará en vigencia el año 2017, contempla las competencias descritas en la subsección anterior. La asignatura requiere 3 horas de trabajo en aula la semana y una cantidad equivalente fuera de ella.

Tradicionalmente, las metodologías de enseñanza de la asignatura son clases magistrales de exposición de contenidos, y un proyecto paralelo desarrollado por los estudiantes fuera del aula, en el que deben llevar a cabo el proceso de análisis de requerimientos para un proyecto por equipos, de entre 3 y 5 estudiantes. Cada equipo elige su propio proyecto, con la condición de que sea un proyecto con usuarios reales. El proyecto es presentado al profesor en tres hitos de avance en el semestre, y junto con el informe técnico asociado a cada hito, constituye un 40% de la calificación. Esta se complementa por dos pruebas de desarrollo y controles de lectura.

### III. PROBLEMATIZACIÓN Y CONSIDERACIONES INICIALES

#### A. Problematicación

Si bien en la metodología tradicionalmente usada los estudiantes realizan proyectos en equipo, existen los siguientes obstáculos para desarrollar las competencias antes descritas:

- El tamaño y complejidad de los proyectos de los estudiantes es menor al de los abordados en la práctica profesional, por lo que el ejercicio de las competencias generales y específicas es más superficial, e incluso podría no darse.
- Una de las funciones de la actividad de especificación de requerimientos es la de comunicar la información organizada entre el equipo de trabajo, que en entornos profesionales puede ser de decenas de profesionales. Con equipos de trabajo pequeños y problemas simples, la especificación pierde esta función, por lo que deficiencias en su nivel de detalle o completitud pueden no afectar el entendimiento del problema o de la solución especificada.
- El hecho que los estudiantes realicen el proyecto fuera del aula, no permite evidenciar si los estudiantes están desplegando las competencias genéricas con las contrapartes, y menos aún, da la oportunidad de corregir o entregar feedback respecto a estas actividades.

#### B. Consideraciones para la implementación de Aprendizaje Basado en Proyectos

Para abordar la problemática anterior se utilizará Project Based Learning, considerando un proyecto de escala e implicancias reales, en el que todo el curso será el equipo de desarrollo. En las secciones siguientes se describe el desarrollo de esta práctica educativa.

El proyecto consisten en realizar el Análisis de Requerimientos para la construcción de un sistema de *Clickers* o Tecleras Virtuales, o en otras palabras, una implementación del Sistema de Respuesta en Aula (SRA) basada en dispositivos computacionales (smartphones, tablets, laptops o

computadores de escritorio). A grandes rasgos, el objetivo es construir un sistema que permita el profesor formular preguntas de selección múltiple para posteriormente ser formuladas en clases desde su computador o dispositivo móvil, de manera que los estudiantes puedan responderlas en tiempo real a través de sus propios dispositivos, y acto seguido visualizar las respuestas del curso. Esto permite activar y evaluar conocimientos previos, y ser utilizadas en una diversidad de situaciones de aprendizaje, según lo diseñe cada profesor. Un objetivo secundario es evaluar el desarrollo de aprendizajes del alumno y del profesor, al asociar cada pregunta a las competencias que contribuye a desarrollar, permitiendo consultar datos estadísticos de desempeño a profesores, estudiantes y División Académica (DIVACAD). Esta definición fue conversada y acordada previamente con Sebastián Lepe, integrante del Centro de Desarrollo Docente (CDD) y director ejecutivo del Proyecto de Mejoramiento Institucional UVA 1407, que busca la incorporación de tecnologías al aula.

Tal como indican Herman, Aschbacher y, Winters [7], la definición para las actividades Aprendizaje Basado en Proyectos se deben abordar las siguientes consideraciones iniciales:

**1. ¿Qué habilidades cognitivas importantes deben desarrollar los estudiantes?** Siguiendo la taxonomía de Bloom [3], las habilidades cognitivas requeridas son de Análisis, que incluyen la capacidad de identificar relaciones entre elementos, reconocer supuestos no expresados, reconocer y organizar los elementos de la situación problemática, y poder realizar conclusiones fundamentadas.

**2. ¿Qué habilidades afectivas y sociales deben desarrollar los estudiantes?** Este punto es el más diverso, puesto que abarca desde habilidades sociales básicas como comunicarse en grupo, presentar un punto de vista o formular preguntas, hasta resolver controversias y enfrentar conflictos. En este sentido, no es posible guiar cada una de estas situaciones, por lo que es fundamental identificarlas dentro de la práctica docente.

**3. ¿Qué habilidades metacognitivas deben desarrollar los estudiantes?** Reconocer qué parte del proceso de Análisis de Requerimientos están llevando en cada acción (levantamiento, especificación análisis o validación), y reconocer las habilidades sociales requeridas para resolver situaciones comunes en el ejercicio profesional del rol de analista, como necesidad de negociar, de discutir en pro del éxito de los objetivos del proyecto, a contener expectativas del cliente, entre otras.

**4. ¿Qué tipo de problemas deben de resolver los estudiantes?** Los principales problemas a resolver son los siguientes:

- ¿Cómo obtener la información con el nivel de detalle necesario para pasar de los deseos del cliente a poder construir un software que los satisfaga?
- ¿Cómo organizar la información obtenida para poder entenderla y dar la entender en su totalidad?
- ¿Cómo verificar que la información completa es correcta, completa y consistente?

- ¿Cómo asegurar que el software que se derivará del análisis satisface los requerimientos del cliente?

**5. ¿Qué conceptos y principios deben los estudiantes estar en capacidad de aplicar?** Los estudiantes deben conocer y comprender algunos de los contenidos revisados en la asignatura prerrequisito “Fundamentos de Ingeniería de Software”:

- El proceso de desarrollo de software (proceso dentro del cual el Análisis de Requerimientos toma lugar)
- Lenguaje de modelamiento UML, en particular Diagramas de Casos de Uso, Especificaciones de Casos de Uso y Modelos de Dominio.
- Herramientas de prototipado.

**6. ¿Los estudiantes tienen acceso fácil a los recursos que necesitan?** Los recursos críticos para la actividad son los siguientes:

- Contraparte: corresponde a quien requiere el desarrollo del software; por lo general es más de un profesional, cada uno con roles distintos: Cliente es quien toma decisiones estratégicas y presupuestarias del proyecto, Experto de Dominio es quien conoce la organización y los procesos de negocio que van a ser soportados por el software a desarrollar, y Usuario, que es quien va a usar el sistema final. Para el proyecto en particular, se requiere que la contraparte conozca el sistema de tecleras tradicional, conozca las necesidades de evaluación de aprendizaje de la Universidad, su estructura y la dinámica de aula. Para este fin, ha sido determinante la colaboración del Centro de Desarrollo Docente, que a través de Sebastián Lepe y en la persona de Antonio González ha permitido contar con una Contraparte trabajando en el aula todas las semanas.
- TICs: En una situación ideal, todos los estudiantes deberían contar con laptops y conexión a internet de alta velocidad para la edición colaborativa de diagramas, documentos y prototipos; sin embargo estos recursos pueden reemplazarse por cartulinas y plumones, quedando su formalización como trabajo fuera del aula, en actividades dirigidas por la ayudante del curso.

**7. ¿Saben los estudiantes cómo utilizar los recursos?** Respecto a las entrevistas con contrapartes, no tienen los conocimientos, por lo que el profesor realizará una clase inicial demostrativa. Respecto a los recursos TIC (que utilizarán fuera del aula), los estudiantes ya los utilizaron en el curso anterior “Fundamentos de Ingeniería de Software”.

**8. ¿Los estudiantes tienen tutores que los ayudan con su trabajo?** El trabajo de análisis se realiza fundamentalmente en clases, y las entregas parciales de documentos son revisadas por la ayudante del curso.

**9. ¿Los estudiantes tienen claros los roles y las responsabilidades de cada una de la personas del grupo?** Este punto puede presentar un riesgo, pues en primera instancia, todos los estudiantes tienen responsabilidad común sobre el

producto final; si bien es posible definir roles, se ha omitido esta complejidad en la primera parte del curso, y se plantea incorporarla luego del primer hito de control.

#### IV. IMPLEMENTACIÓN DE APRENDIZAJE BASADO EN PROYECTOS

##### A. Contexto

La actividad se inició el 13 de agosto de 2015, en la que la edición del curso cuenta con 26 estudiantes. La jornada de clases es de 3 horas semanales, que se llevan a cabo entre las 10:15 y a las 13:30, con un intermedio de 15 minutos. La clase se divide en dos bloques: en los primeros 60 minutos se revisan contenidos asociados al Análisis de Requerimientos, tales como técnicas y marcos conceptuales para el análisis. Los 30 minutos restantes del primer bloque son para una aplicación inicial de las técnicas, realizando la especificación de requerimientos levantados en clases anteriores. En los 90 minutos finales se cuenta con Antonio González como contraparte, y se lleva a cabo la entrevista para el levantamiento, análisis y validación de requerimientos.

Los estudiantes trabajan en equipos de entre 3 y 5 integrantes, elegidos por su propio criterio. La diferencia de tamaño de los equipos se compensa con la complejidad de los sub temas que debe abordar cada equipo.

El presente reporte abarca las actividades realizadas hasta el 01 de octubre de 2015. En cada uno de los puntos siguientes se describirán las actividades y momentos destacados de aprendizaje, en los que ocurren situaciones imposibles de exponer si no es a través de la vivencia en el proyecto.

##### B. Inicio (13 de agosto de 2015)

En esta clase, junto con presentar el curso en términos generales, se introduce el proyecto a tratar. Primero, se expone el proyecto a tratar, utilizando videos ilustrativos de cómo funcionan las tecleras virtuales, y cuál es su objetivo pedagógico. Se informa que su trabajo será el punto de inicio para que otro curso, “Desarrollo Web” programa el software que ellos están definiendo.

Además se activan conocimientos previos relacionados con proceso de desarrollo de software y diagramas UML, a través de preguntas y debate en clases.

Posteriormente, se informa de la modalidad de trabajo en grupos, y se conforman estos. Luego, se presenta la fecha tentativa de fin del proyecto (8 de octubre, y el proceso de trabajo, consistente en el proceso de Ingeniería de Requerimientos (presentado en el marco teórico), y su contexto en el proceso de desarrollo de software. Se indican además las fechas de hitos de entrega parciales y globales. Se revisa en forma teórica las principales formas de obtención de información (levantamiento de requerimientos). Se indica además que durante las clases se irán revisando distintas técnicas de levantamiento, especificación, análisis y validación de requerimientos, y que la documentación formal y material complementario será entregado según se vaya avanzando con las actividades. Se explica además que todas las semanas se trabajará en la modalidad antes descrita.



Como momento de aprendizaje destacado, es notable la reactivación de conocimientos previos. Para ello, los estudiantes ejercitaron los conceptos que serán usados en la clase siguiente (definición de objetivos de un proyecto, sistema y de negocio) mediante un ejercicio práctico aplicado a un proyecto ficticio desarrollado durante el semestre anterior, en el curso Fundamentos de Ingeniería de Software.

En esta etapa se logró exponer a los estudiantes los objetivos del curso (Preguntas 1 a 5 del diseño), y se entregaron las indicaciones de trabajo, recursos y se reactivaron los aprendizajes a aplicar, siguiendo lo planteado en las preguntas 6 a 9 del diseño.

### *C. Actividades Iniciales (13 y 20 de agosto de 2015)*

La primera reunión de trabajo contó con la presencia de Sebastián Lepe, Antonio González. En ella, el profesor explicó a los estudiantes que él realizará la entrevista inicial para poder conocer el alcance del sistema para luego separar el proyecto en sub temas a asignar a cada grupo. Primero se acuerdan los objetivos del proyecto, y posteriormente, el profesor entrevista a las contrapartes hasta lograr una visión global del sistema a desarrollar. Esta visión queda inconclusa y se resuelve finalizarla en la siguiente reunión. En la reunión del día 20 de agosto se termina la visión de alto nivel, y se asignan elementos separables (Casos de Uso) a cada uno de los seis equipos. Se encarga para la clase siguiente que cada equipo prepare un prototipo de los escenarios del sistema para el caso de uso asignado, y escriba historias de usuario [8].

Como momento de aprendizaje destacado, antes de partir con la entrevista, el profesor plantea la pregunta a la contraparte, ¿cuál es el objetivo del proyecto?, y solicita a los estudiantes complementar con preguntas hasta asegurar un entendimiento común del problema (el nivel de detalle y clasificación de los objetivos fue un contenido visto en el curso anterior y repasado la clase anterior). Sorpresivamente para los estudiantes, las contrapartes plantean objetivos distintos (esto no fue previamente diseñado). Sebastián había señalado el mismo objetivo presentado por el profesor a los estudiantes, mientras que Antonio había dado uno distinto. El profesor, advirtiendo la situación que se estaba dando, observa el comportamiento de los estudiantes durante algunos minutos, observando las siguientes actitudes:

- Mediante lenguaje no verbal, algunos estudiantes expresan incomodidad con el objetivo planteado por Antonio, aunque ninguno dice nada.
- Los estudiantes, para afinar el objetivo del proyecto, preguntan únicamente a Sebastián, sin incluir a Antonio en la conversación.

Ante esta situación, el profesor realiza una “pausa metacognitiva”, y pregunta a uno de los estudiantes en los que advirtió lenguaje no verbal de incomodidad, ¿Qué está pasando?. El estudiante, sorprendido, declara que Antonio habló de algo que para él no era el proyecto que iban a trabajar, así que asumió que estaba equivocado. Al preguntar a quienes formularon preguntas a Sebastián la respuesta fue similar. El profesor señala entonces que se están cometiendo dos errores típicos del análisis: por una parte, asumir que Antonio está

equivocado, siendo que es posible que la contraparte haya decidido hacer más ambicioso el proyecto; en este sentido, esto presenta un conflicto para un Analista de requerimiento, que debe mantener el proyecto en el alcance original. Por otra parte, entonces, se está rehusando a un conflicto, que de no ser abordado será detectado en las fases finales del proceso, produciendo un problema de mayor escala e impacto en el costo del proyecto. El profesor señala entonces que estos conflictos y divergencias deben ser planteados y en lo posible, resueltos. El profesor ejemplifica cómo plantear la situación y cómo acordar un objetivo unificado para el proyecto.

Respecto al cumplimiento de objetivos del diseño del aprendizaje basado en proyectos, en esta etapa se trabajó en los aprendizajes indicados en las preguntas 2 y 3, es decir, habilidades sociales y aspectos meta-cognitivos. El “saber y sentir qué está pasando” es un desafío en las reuniones iniciales de proyecto, lo que se pudo replicar en las dos primeras reuniones. La mediación del profesor en este caso consistió en dar la oportunidad de que los estudiantes, por ellos mismos, identificaran estos aspectos.

### *D. Implementación del proyecto (27 de agosto al 3 de diciembre)*

En las sesiones sucesivas, los estudiantes van elaborando, presentando y refinando sus productos de trabajo a la contraparte, que va entregando indicaciones para dar mayor detalle cada vez. Dentro de los productos de trabajo considerados hasta el momento, se encuentran:

- Prototipos: creados en una herramienta computacional, representan la parte gráfica del software. Son usados tanto para explicar a la contraparte lo que se ha entendido de cómo debería ser el sistema como para levantar o aclarar requerimientos.
- Historias de usuario: son descripciones que indican qué haría un usuario en cada prototipo y con qué propósito.
- CRC Cards: son tarjetas que describen un concepto del dominio, sus funcionalidades principales y relaciones con otros conceptos. Permiten organizar y uniformar el vocabulario que emerge de las historias de usuario y la conversación con el cliente. Se trabaja con post-it, que cada equipo desarrolla y luego pega en la pizarra; al identificarse conceptos similares o repetidos entre grupos de trabajo, se debate cómo debe quedar en forma final.
- Especificación de caso de uso: Documento que describe con mayor detalle la lógica de funcionamiento de cada escenario del prototipo.
- Diagramas de actividad: describe el flujo de información entre los distintos usuarios del sistema.

Los primeros 90 minutos de clase previos a la reunión sirven para revisar cómo crear estos productos y ensayar con supervisión del profesor; en las reuniones se exhiben los productos creados en clases anteriores y para la siguiente clase, los estudiantes los traen corregidos, y desarrollan un nuevo producto.

Si bien la incorporación progresiva de estas técnicas es definida por el profesor, su necesidad resulta natural para los estudiantes, dado el nivel de complejidad de la información que capturan y que necesitan organizar y profundizar. Por ejemplo, tras una clase de reunión de trabajo con el Cliente en la que se exploran los primeros Casos de Uso, en la siguiente clase de contenidos, cada equipo presenta brevemente sus hallazgos al resto de los equipos. Esto propicia que le digan de forma distinta a los mismos elementos del dominio y a sus acciones, lo que motiva la incorporación de CRC Cards. En luego de esto, los estudiantes descubrieron que varios grupos le daban el mismo nombre a conceptos distintos, y se logró hacer un análisis fino del concepto “Pregunta”. Mientras uno se referían a Pregunta como aquella que había creado el profesor, pero que aún no se había formulado en clase, otros se referían a Pregunta como solo a aquellas que se formulaban en clase, y otros usaban la misma palabra para las respuesta de los alumnos. Así, se logró acordar que se trabajaría con los conceptos Pregunta, Pregunta Formulada y Pregunta Contestada. Esta discusión no se habría dado de no trabajar con un proyecto común a todo el curso y de una complejidad mediana, como para llevar a algún grado de confusión interesante de ser analizado.

Con respecto al logro de objetivos de aprendizaje planteados en el diseño del curso, los conocimientos y habilidades de las preguntas 1 y 4 fueron los más intensamente ejercitados en las sesiones y en las clases preparatorias para entregar métodos y técnicas para su desarrollo. Se repitieron momentos como los mencionados en la etapa anterior, donde también se reforzaron aspectos meta-cognitivos y emocionales.

#### *E. Resultados*

A nivel cualitativo, evidenciado en las actividades formativas y en las entrevistas, fue posible apreciar a los estudiantes desplegar habilidades no solo ingenieriles, sino tecnológicas, que difícilmente pueden exponerse con claridad mediante una clase magistral; de la misma forma, por ser experimentadas directamente por los estudiantes, se estima que estas instancias pueden ser de aprendizaje profundo: varias situaciones que han sido problemáticas han sido llevadas a metáforas o apodos, de manera que cuando se repiten, se hace referencia a estas metáforas, potenciando no sólo el aprendizaje por repetición, sino también la metacognición. Por ejemplo, cuando un analista “inventa requerimientos”, sin que la contraparte los haya solicitado, agregando así costo o complejidad al proyecto, los estudiantes dicen que se está “quemando dinero”; o cuando hay un momento de discrepancia no declarada con la contraparte, los estudiantes son capaces de distinguir que la situación “se puso incómoda”.

En cuanto a las evaluaciones sumativas, en la primera primera entrega de productos al cliente (10 de septiembre), el 100% de los equipos presentó los productos solicitados, aunque uno de los equipos no logró comprender cabalmente el propósito de la unidad de trabajo que se le asignó, pero incluso este error, puesto en común de forma respetuosa y constructiva, sirvió para enseñar la importancia de definir primero los objetivos y luego realizar los productos de trabajo. En las siguientes entregas, se mantuvo el porcentaje de cumplimiento y la brecha entre las expectativas del cliente y los productos fue

atenuándose cada vez más. El elemento más significativo fue el porcentaje de aprobación (96%), reprobando sólo un estudiante por abandono de deberes, identificado por sus compañeros de equipo mediante una rúbrica de coevaluación. La situación se conversó tempranamente con el equipo y el estudiante en cuestión, y al no revertirse, las condiciones de calificación determinaron su reprobación.

Con respecto a anteriores ediciones del curso, si bien las calificaciones son bastante parecidas en las ediciones 2013, 2014 y 2015 (en una escala de 1 a 7, con 60% de exigencia los promedios 5.4, 6.1, y 5.8, respectivamente, con desviaciones estándar bajo el 0.8), las evaluaciones de 2013 y 2014 se centran en la presentación de productos y resultados de las distintas fases del proceso de ingeniería de requerimientos, sin posibilidad de evaluar aspectos propios del proceso, que fueron evidenciados en la experiencia descrita anteriormente. Un elemento que permite caracterizar el logro de los resultados de aprendizaje es la rúbrica de evaluación presentada en la tabla 1, que determinó la calificación sumativa final a partir de la evaluación del cliente (40%), profesor (30%) y promedio de evaluaciones del resto de los equipos (30%). La nota individual se calculó aplicando a todas las evaluaciones sumativas del equipo un factor de coevaluación de los compañeros de equipo, que permitía modificar la nota desde un 50% (en caso de evaluación insuficiente) a un 110% (evaluación sobresaliente). Las rúbricas de evaluación, contenidos y otros recursos están disponibles en <https://goo.gl/4HYjPt>

#### **V. CONCLUSIONES Y LECCIONES APRENDIDAS**

El aprendizaje basado en proyectos, como estrategia de aprendizaje, promete el desarrollo de habilidades en contextos similares a los profesionales. El ejercicio de la práctica antes descrita constata la ocurrencia de situaciones de aprendizaje difícilmente reproducibles con el enfoque anterior del curso o con métodos de aprendizaje tradicionales, como las cátedras; incluso, la diversidad de los momentos destacados de aprendizaje no podría haber sido diseñada razonadamente. Los resultados de la práctica han permitido el monitoreo continuo del desarrollo de competencias complejas, y la evaluación de los resultados de aprendizaje en términos concretos y aplicables al mundo profesional real.

Una de las condiciones para aplicar la metodología es disminuir la cantidad de contenidos o su nivel de detalle. En este caso, se tomó como apoyo el Cuerpo de conocimientos de la Ingeniería de Software (SWEBOK) [9] para asegurar que los contenidos revisados cumplan con lo necesario para cubrir este marco de referencia.

Uno de los elementos clave para el éxito y dinamismo de la práctica ha sido contar con contrapartes reales (CDD) durante las sesiones de aula. Esta importante dedicación de tiempo es costosa de replicar en todas las asignaturas, sin embargo, podrían elaborarse estrategias de trabajo para proyectos de otras disciplinas teniendo como contrapartes estudiantes de cursos superiores o incluso intercambio de académicos. En particular, esta asignatura piloto del uso de metodologías de docencia activa, en el marco del Proyecto de Mejoramiento Institucional UVA1405 – STEM, ha permitido la articulación de este proyecto con el Proyecto UVA1407, orientado al uso de

tecnologías de la información en el aula. Se vislumbra así la posibilidad de desarrollar herramientas educativas tecnológicas en cursos de informática, para las necesidades específicas de la Universidad.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado por los Proyectos de Mejoramiento Institucional MECESUP 3, UVA1405 y UVA1407. Roberto Muñoz es beneficiario de la Beca de Doctorado INF-PUCV 2015.

#### REFERENCIAS

[1] R. S. Pressman, *Software engineering: a practitioner's approach*, Eighth edition. New York, NY: McGraw-Hill Education, 2015.

- [2] J. Tardif, "Desarrollo de un programa por competencias: De la intención a su implementación", *Rev Curric Form Profr*, vol. 16, n° 3, pp. 36–45, 2008.
- [3] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview", *Theory Pract.*, vol. 41, n° 4, pp. 212–218, nov. 2002.
- [4] C. Meyers y T. B. Jones, *Promoting active learning: strategies for the college classroom*, 1st ed. San Francisco: Jossey-Bass, 1993.
- [5] B. F. Jones, C. M. Rasmussen, y M. C. Moffitt, *Real-life problem solving: a collaborative approach to interdisciplinary learning*, 1st ed. Washington, DC: American Psychological Association, 1997.
- [6] R. Miles y K. Hamilton, *Learning UML 2.0*, 1st ed. Beijing; Sebastopol, CA: O'Reilly, 2006.
- [7] J. L. Herman, P. R. Aschbacher, y L. Winters, *A practical guide to alternative assessment*. Alexandria, VA: Association for Supervision and Curriculum Development, 1992.
- [8] K. Beck y C. Andres, *Extreme programming explained: embrace change*, 2nd ed. Boston, MA: Addison-Wesley, 2005.
- [9] P. Bourque, "SWEBOOK V3 • IEEE Computer Society". [En línea]. Disponible en: <https://www.computer.org/web/swebok/v3>. [Accedido: 31-jul-2016].

TABLA I. RÚBRICA DE EVALUACIÓN FINAL.

| Criterio  | Nivel de Desempeño: Logrado (5 pts)   | Nivel de Desempeño: Medianamente Logrado (3 pts)   | Nivel de Desempeño: No Logrado (1 pto)  |
|---|---|--|---|
| <b>Presentación de Requerimientos:</b><br><i>Objetivos del Caso de Uso</i>  | Se presenta el objetivo del caso de uso y sus funcionalidades con claridad. Se indica su contribución al resultado final y su contexto en el proceso completo.            | Se presenta el objetivo del caso de uso pero no se aclara su contexto dentro del proceso completo.   | No se indica el propósito, objetivo ni contexto del caso de uso.  |
| <b>Presentación de Requerimientos:</b><br><i>Descripción de los funcionalidad</i>                                       | La totalidad de los escenarios (pantallas prototipadas) presentados permiten cumplir el objetivo del caso de uso, y son consistentes con lo acordado en las reuniones.    | Existen a lo más 2 escenarios muy distintos o que faltan para poder mostrar toda la funcionalidad acordada en clases.  | Existen más de 2 escenarios muy distintos o que faltan para poder mostrar toda la funcionalidad acordada en clases.     |
| <b>Presentación de Requerimientos:</b><br><i>Administración de información (entidades de datos y modelo de dominio)</i> | Toda la información que se requiere que el sistema administre (cree, modifique, elimine o muestre) ha sido presentada por el equipo o ha sido explicitada ante preguntas. | La mayor parte de la la información que se requiere que el sistema administre (cree, modifique, elimine o muestre) ha sido presentada por el equipo o ha sido explicitada ante preguntas. A lo más 3 datos requeridos han sido omitidos. | Más de 3 datos han sido omitidos.   |
| <b>Presentación de Requerimientos:</b><br><i>Consistencia de las representaciones</i>                                   | Las distintas representaciones de la funcionalidad requerida son consistentes entre sí.   | A lo más 2 errores de consistencia han sido identificados.   | Existen más de 2 inconsistencias entre los modelos.   |
| <b>Presentación de Requerimientos:</b><br><i>Aclaración de inconsistencias de análisis (sólo si corresponde)</i>        | El equipo plantea las inconsistencias o problemas de análisis identificados en forma clara, y proponen alternativas para resolverlas.                                     | El equipo plantea las inconsistencias como un problema, traspasando la responsabilidad de resolverlas al Cliente.  | El equipo no plantea inconsistencias que si son observadas por el Cliente o los pares.                                  |
| <b>Presentación de Requerimientos:</b><br><i>Análisis de Cambios</i>  | El equipo es capaz de identificar qué Caso de Uso se debe cambiar y cómo cambiarlo para poder resolver un cambio de requerimiento presentado por el Cliente.              | El equipo es capaz de reconocer el impacto del cambio a nivel local (su propia funcionalidad) peor no puede asociarla a funcionalidades trabajadas por otros equipos.  | El equipo no es capaz de identificar el impacto por cambio de requerimientos.   |
| <b>Exposición oral:</b><br><i>Manejo del tiempo</i>   | Cumple con el tiempo  |  | No cumple con el tiempo   |
| <b>Exposición oral:</b><br><i>Distribución de la presentación</i>   | Todos los integrantes participan  | Algunos participan   | Sólo uno expone   |
| <b>Exposición oral:</b><br><i>Resolución de dudas</i>   | Las respuestas a las preguntas son claras o manifiestan claramente que no se dispone información para responderlas.   | Las respuestas son ambiguas y deben ser guiadas para ofrecer información clara.  | Las respuestas no son claras y existe una actitud <b>defensiva</b> por parte del equipo para recibir las observaciones. |