

Usando A+S y desarrollo ágil para alcanzar aprendizaje significativo en la enseñanza de Proyectos software

Using A+S and agile development to achieve significant learning in teaching software projects

Víctor Flores

Departamento de Ingeniería de Sistemas y Computación
Universidad Católica del Norte. Av. Angamos 0610
Antofagasta, Chile
vflores@ucn.cl

Claudia Lepe

Centro de Innovación Metodológica y Tecnológica-CIMET
Universidad Católica del Norte. Av. Angamos 0610
Antofagasta, Chile
clepea@ucn.cl

Abstract—The learning based on A+S has been successfully applied in various areas of teaching on several university fields. On the other hand, in the software development context, agile methodologies are currently being the most used option for software development projects because they encourage both early and continuous feedback that helps to focus and adapt the development to the client needs. This document describes a pilot experience that seeks to combine both concepts for teaching in the matter “Taller de Programación”. This pilot experience included the active participation of a community partner (depending of the Antofagasta Council). In order to evaluate the experience results, a plan for continuous and assessment evaluation of the tasks accomplished by the student has been generated. The obtained results indicate improvements in the active participation of the students and show that the method can be used in the future, both to improve students' outcomes and to achieve learning objectives related to development of quality software.

Keywords—A+S; quality software; Active learning methodologies; Agile software methodologies.

I. INTRODUCCIÓN

A partir del año 1990 el Espacio Europeo de Educación Superior establece desafíos para la educación universitaria, generar cambios significativos en su enseñanza, no sólo en Europa, sino también en Latinoamérica [1, 3]. Estos cambios implican asumir un nuevo paradigma centrado en el aprendizaje, es decir, enfocado en los estudiantes. Junto con ello, se hace necesario facilitar la integración de los futuros profesionales al ámbito laboral, para lo cual es fundamental que el proceso de enseñanza-aprendizaje sea a través de la formación por competencias. Es en este contexto que la Universidad Católica del Norte declara en su Proyecto

Educativo Institucional una formación basada en competencias desde un paradigma socio-constructivista¹.

En este contexto, la educación en Computación e Informática se ha visto beneficiada con la aparición de nuevas herramientas pedagógicas, y se pueden encontrar experiencias docentes con excelentes resultados de desarrollo de competencias [15]. Entre los aportes de estas herramientas se puede destacar el cambio de paradigmas del eje vertebrador del conocimiento (como por ejemplo Aprendizaje Basado en Problemas (ABP) [4]) o los cambios en el modelo de interacción alumno-profesor en el proceso de enseñanza-aprendizaje.

Dado lo anterior, es fundamental abordar el proceso de enseñanza-aprendizaje en que el estudiante sea el principal protagonista, por lo que requiere de la implementación de metodologías activas en el aula. Las metodologías activas y sus estrategias conciben el aprendizaje como un proceso constructivo y no receptivo, debido a que el estudiante tiene la oportunidad de experimentar, interactuar, reflexionar, investigar y comunicarse y el docente cumple el rol de facilitador de este proceso. Las metodologías activas de la escuela nueva, enfatizan que la enseñanza debe tener lugar en el contexto de problemas del mundo real o de la práctica profesional [8, 12, 15].

En este artículo se presenta la experiencia docente en una asignatura centrada en la mejora de técnicas de programación orientada a objetos, donde se ha planteado un cambio en el modelo de impartir la clase para incorporar objetivos de

¹ <http://www.ucn.cl/sobre-ucn/somos-ucn/proyecto-educativo-ucn/>

compromiso comunitario al objetivo ya existente de crear un prototipo software de calidad y cercano al ejercicio profesional de desarrollo de software.

De forma concreta, se ha incorporado la metodología A+S [13, 16], con la participación de un socio comunitario que ha actuado como cliente en el desarrollo de un software destinado a ser usado por niños con problema de aprendizaje del lenguaje en la Región de Antofagasta, Chile.

Esta experiencia docente está en estrecha relación con los objetivos del proyecto educativo de la Universidad Católica del Norte (UCN) de Chile. La UCN pretende entre sus objetivos, fomentar en los futuros profesionales el compromiso solidario, que es a su vez un objetivo de la educación superior en Chile [13]. Concretamente en este respecto, la UCN pretende formar profesionales en un contexto de responsabilidad social: búsqueda del bien común, equilibrio social, y la preservación del entorno, lo cual es perfectamente alcanzable con experiencias como la descrita en este documento.

Los resultados de esta experiencia han sido muy satisfactorios, por un lado, se ha reducido el nivel de abandono de la asignatura, ha aumentado la tasa de participación activa y las notas son sensiblemente mejores al curso anterior. Por otro lado, se ha logrado sensibilizar a los alumnos con el trabajo solidario y se ha logrado una mayor participación de los estudiantes en las actividades pautadas en la asignatura, desarrollando de esta forma la competencia de trabajo en equipo.

II. CONTEXTO DE TRABAJO

El trabajo reportado en este documento se sustenta en tres aristas; por un lado y siguiendo el enfoque constructivista de Biggs [6], el cual esencialmente considera que lo que importa es lo que el estudiante hace, y por otro lado centrado en el desarrollo de competencias de Responsabilidad social, se ha seleccionado la metodología A+S para guiar el proceso docente. Por otro lado, la tercera arista la conforman las metodologías ágiles soportadas en el manifiesto ágil [7] que fija la mayor prioridad en un proyecto de desarrollo de software, en la satisfacción del cliente a través de entregas tempranas y continuas que aporten valor.

Las metodologías activas han demostrado ser en la docencia universitaria de la informática, un medio para alcanzar aprendizaje profundo con las competencias transversales y específicas en las asignaturas [1], además de ser métodos que facilitan y permiten aprendizaje mediante la práctica [8, 15].

Son varias las experiencias previas del uso de metodologías activas para dar soporte al desarrollo de asignaturas del área de Computación e Informática, por ejemplo en [9] se describe el uso de ABP y principios de SCRUM²; proceso de la metodología ágil que ha sido usado en diversos proyectos con resultados destacables en características como cumplimiento de plazos o la usabilidad del producto final [10]. Otro trabajo relacionado es el descrito

en [11] que describe el uso de PBL (Problem-Based Learning) en un curso de electrónica en la Universidad de Oviedo, España. En la experiencia usaron el PBL para guiar el proceso de enseñanza-aprendizaje conjuntamente con las técnicas propias del diseño y desarrollo de proyectos de ingeniería electrónica.

La experiencia docente aquí descrita se basa en combinar A+S con principios de métodos ágiles de desarrollo de software, para generar un proyecto de software con un fuerte componente de acción social, a la vez de guiar el aprendizaje activo, participativo y centrado en el estudiante, que permite además potenciar las habilidades del estudiante del siglo XXI [12]. Esto se desarrolla en el contexto de la asignatura “Taller de Programación” de la carrera Ingeniería de Ejecución en Computación e Informática (IECI) de la UCN. La asignatura se imparte en el V semestre de la carrera.

La carrera IECI es una carrera tecnológica que tiene declarado en el Perfil del egresado, un profesional integral y productivo, con principios éticos y con un sentido amplio de responsabilidad social, comprometido con el desarrollo regional y del país³. Dentro del pensum de la carrera, la asignatura Taller de Programación es una asignatura que tiene dos objetivos principales: por un lado, profundizar en técnicas de programación orientada a objetos para lograr aplicaciones software de calidad, y por otro lado, utilizar elementos de apoyo al desarrollo de software como por ejemplo, las IDEs, sistemas de control de versiones o herramientas recientemente (en los últimos dos años académicos) se ha orientado al objetivo de programar aplicaciones para tablets, siguiendo la tendencia de la demanda de aplicaciones en Android.

Esta asignatura está ubicada en el quinto semestre de la carrera y se imparte una vez al año. Concretamente los objetivos de aprendizaje declarados en el curso son: desarrollar destrezas en Programación Orientada a Objetos (POO), desarrollar programas de calidad y utilizar herramientas de ayuda al desarrollo de sistemas software de calidad.

III. DESCRIPCIÓN DE METODOLOGÍAS Y EXPERIENCIA

En este apartado se describe con detalle las metodologías usadas en la estrategia de trabajo, tanto la de aprendizaje activo como la que enmarca el proceso de construcción del prototipo software, además se describe la experiencia docente y sus resultados.

En la asignatura Taller de Programación se ha fijado como estrategia de trabajo en los tres últimos años, desarrollar prototipo de un software con niveles aceptables de calidad y usando una metodología activa. En los dos cursos anteriores se usó PBL y en el curso académico 2016-2017 se ha usado A+S. El PBL es un método de enseñanza-aprendizaje cuyo punto de partida es un problema diseñado previamente por el profesor, el cual se espera que el estudiante pueda resolver para desarrollar determinadas competencias previamente definidas [2].

² <https://www.scrum.org/>

³ <http://www.disc.ucn.cl/ingenieria-en-computacion-e-informatica/>

Por otro lado, el Aprendizaje más Servicio (A+S) es definida como “una metodología pedagógica experiencial, que se caracteriza por la integración de actividades de servicio a la comunidad en el currículo académico, donde los alumnos utilizan los contenidos y las herramientas académicas en atención a necesidades genuinas de una comunidad” [13].

Respecto al desarrollo ágil, éste se basa en el Manifiesto ágil⁴ como ya se ha comentado en la Introducción. En este contexto, los principios usados fueron la definición/establecimiento de requisitos, establecimiento del equipo de trabajo con el reparto de roles pertinente, el uso de incrementos e iteraciones, para llegar a un prototipo estable; contando además con la participación activa del cliente quien trabajó estrechamente con el grupo de estudiantes del curso.

A. Cómo se trabajó con A+S

La metodología A+S se implementa en esta asignatura en tres etapas [16]: planificación, implementación y evaluación. La primera etapa correspondió a la organización de la asignatura, el primer contacto con el socio comunitario y el trabajo de reflexión al tipo de servicio comunitario; a partir de esto, se firmó un acuerdo con la Oficina de Ayuda a Personas con Discapacidad de Antofagasta, dependiente de la Ilustre Municipalidad de Antofagasta.

La segunda etapa corresponde al desarrollo del proyecto software para dar respuesta a la necesidad del cliente. Esta necesidad fue planteada como un objetivo general que se formalizó de la forma siguiente: facilitar la tarea de terapeuta del aprendizaje, haciendo más amigable los ejercicios de construcción de frases adaptados a niños con trastorno del aprendizaje (TA). Esta etapa correspondió al desarrollo de software y la base de datos, validación de avances en la construcción del prototipo con el cliente, pruebas funcionales y feedback con el cliente. Para dar seriedad al proceso se firmaron actas y se realizaron evaluaciones periódicas que involucran contenidos del temario y aspectos del proyecto.

La tercera etapa corresponde a la evaluación del proyecto A+S y la retroalimentación del socio comunitario. Esta etapa correspondió tanto a la evaluación de aspectos técnicos (propios de las unidades de la asignatura) como de aspectos de la calidad del servicio prestado al socio comunitario. A tal efecto, se realizaron presentaciones donde el socio comunitario calificaba y proporcionaba una evaluación cuantitativa en las presentaciones (ver Tabla 1). Bajo esta metodología, se plantearon los siguientes parámetros:

- El estudiante no sólo debe desarrollar las competencias específicas de su profesión, sino también aquellas competencias genéricas que aseguran un actuar profesional tanto exitoso como socialmente responsable
- El estudiante debe trabajar orientado a la responsabilidad social, la responsabilidad social es una forma distintiva de entender el rol de una

universidad con valores provenientes del Humanismo Cristiano⁵.

La figura 1 ilustra el contexto de trabajo descrito en los párrafos anteriores. Las ventajas o facilidades de la metodología A+S [13, 16] que fueron consideradas para guiar el trabajo en la asignatura son:

- Metodología activa que puede guiar el proceso de aprendizaje en los alumnos
- Metodología que involucra a los alumnos en el desarrollo de un software para una comunidad con necesidades concretas
- Permite desarrollar en los alumnos habilidades extracurriculares, acercando a los estudiantes a una labor social real y concreta.

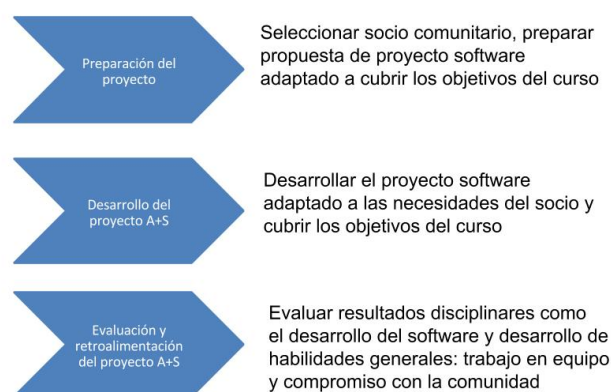


Fig. 1. Etapas y detalle de cada una de las etapas de la metodología A+S.

En la implementación del A+S, el docente debe facilitar a sus estudiantes un proceso de reflexión intencionada y permanente en función de facilitar la conexión del servicio con los resultados de aprendizaje de la asignatura y todos los aspectos relacionados con su intervención en una comunidad determinada. Además de facilitar la integración de aprendizajes, la metodología de A+S permite que el estudiante comprenda el rol social de su disciplina. Desde el A+S, toda disciplina cumple tareas de relevancia social [5], el desarrollo de software se orientó entonces a un servicio social, guiado también por los contenidos de la asignatura.

El temario de la asignatura está dividido en cuatro unidades, en la primera unidad (U-I) se enseña, tanto principios del desarrollo formal del software como bases de Ingeniería del software, ambos elementos tributan al objetivo de conocer y aplicar métodos y técnicas de desarrollo de proyectos software. En la segunda unidad (U-II) se estudia principios de arquitectura de software, técnicas de modelado y documentación, lo que tributa al objetivo de conocer y aplicar dichas técnicas en el diseño de sistemas software. En la tercera unidad (U-III) se estudian técnicas de control de versiones y testing, lo que tributa al objetivo de trabajar con repositorios para la seguridad del proyecto y/o relativo a las pruebas

⁴ <https://www.agilealliance.org/>

⁵ <http://www.ucn.cl/sobre-ucn/somos-ucn/mision-y-vision/>

software. Finalmente la última unidad (U-IV) se estudia el mantenimiento del software, se estudian entonces técnicas relativas a los diversos tipos de mantenimiento del software.

B. Planificación de la docencia y programación del proyecto de prototipo software

La asignatura se ha desarrollado en un conjunto de 15 semanas, según detalla la Tabla 1. El contexto general de trabajo estuvo guiado por las etapas de A+S y el contexto particular del desarrollo del software estuvo guiado por actividades típicas del proceso de desarrollo ágil. Como muestra la Tabla 1, a lo largo de la asignatura se realizaron evaluaciones cortas y presentaciones.

Durante el curso, las evaluaciones cortas han estado relacionadas directamente con lo conceptual, el objetivo en estas evaluaciones ha sido validar los conocimientos técnicos del área de desarrollo de software; mientras que las presentaciones de avances del proyecto están relacionadas tanto con lo técnico como con la calidad del servicio prestado al socio comunitario.

Para evaluar las dos presentaciones de avances del proyecto se utilizó una rúbrica, también se diseñó un instrumento para capturar la coevaluación de los alumnos (con el objetivo de valorar el trabajo en equipo) y se consideró también la evaluación del socio comunitario. Las ponderaciones de estas notas fueron 80% (evaluación del docente), 10% (coevaluación) y 10% (socio comunitario). La semana 16 fue destinada a realizar evaluaciones de ajuste o evaluaciones perdidas.

El instrumento de coevaluación se diseñó de la forma siguiente: cada alumno evaluó a todos los miembros de su grupo de trabajo, la evaluación estuvo dirigida a (1) participación en el trabajo de grupo, (2) cumplimiento de tareas asignadas por rol y (3) calidad y oportunidad del trabajo. Luego las notas de cada estudiante evaluador fueron promediadas. Este instrumento se utilizó cada vez que se realizaron presentaciones ante el cliente y el docente.

TABLE I. DETALLES DEL PLAN DE TRABAJO USANDO A+S Y PRINCIPIOS DE DESARROLLO ÁGIL.

Semana	Etapas A+S	Actividad/contenidos	Peso en Evaluación
1	preparación del proyecto	Activación de conocimientos previos: clases de POO, Lenguaje Java.	
2		U-I, qué es A+S, selección del socio comunitario	
3		U-I, U-II, reuniones con socio comunitario, preparación de la propuesta de proyecto	
4		U-I, U-2, discusión de la propuesta de proyecto con socio comunitario	
5		evaluación corta U-I y U-II, preparación del diseño software	20%

6	desarrollo del proyecto A+S	presentación de propuesta, selección de la plataforma de desarrollo y preparación de las iteraciones e incrementos. Primera presentación	
7		U-II, U-III, programación de la primera iteración del prototipo	
8		U-III, programación de la primera iteración del prototipo	
9		desarrollo y validación del incremento con el cliente	
10		presentación preliminar de resultados del proyecto	30%
11		evaluación corta U-III, programación segunda iteración del proyecto	10%
12	desarrollo del proyecto A+S	U-IV, programación de la segunda iteración del prototipo	
13		U-IV, desarrollo y validación del incremento con el cliente	
14	Evaluación y retroalimentación del proyecto A+S	evaluación corta U-IV, preparación de resultados del proyecto software	10%
15		presentación de resultados del proyecto	30%

El instrumento de evaluación diseñado para el socio comunitario estaba destinado a que el cliente con “vista de usuario” pudiera medir aspectos no-técnicos del desarrollo. Es decir, valorar aspectos como, por ejemplo: la usabilidad potencial del producto software (tanto en su trabajo diario como para otros profesionales en la misma labor), calidad del diseño de la interfaz o funcionalidad del producto software acorde a las necesidades del usuario.



Fig. 2. Ejemplo de ventana de la aplicación software.

De igual forma, este instrumento fue usado por el socio comunitario cada vez que se realizó una presentación del prototipo y la evaluación de este socio comunitario correspondió (en todos los casos) al 10% de la nota.

El resultado de este proceso de construcción software en la asignatura fue una aplicación para Tablets desarrollada en Android Studio⁶ y usando una base de datos SQLITE⁷ para guardar las frases que usa el socio comunitario (Fonoaudiólogo especializada en terapia del lenguaje) para ayudar a niños con dificultades de lenguaje. El desarrollo se realizó utilizando una Tablet Sony xperia® con resolución de 1920x1200 píxeles.

La aplicación tiene en su ventana principal una imagen que describe la situación que el niño debe usar para construir la frase (parte inferior de la ventana), usando las figuras o pictogramas disponibles en la interfaz (parte derecha de la ventana). La aplicación está hecha para cubrir tres niveles de dificultad, el primer nivel está compuesto por oraciones que el niño debe formar usando solo tres pictogramas, en el nivel dos se debe usar cuatro pictogramas y cinco pictogramas en el nivel tres.

La figura 2 corresponde a una ventana de la aplicación desarrollada por los alumnos. La oración se debe formar usando los pictogramas de la parte superior derecha. La figura del perro y la frase (parte inferior derecha) fueron diseñados por los alumnos como un extra del producto y su finalidad es ayudar al niño en la realización del juego. La descripción en texto de la parte inferior de la figura 2 no es parte de la aplicación, corresponde a la descripción de la ventana que los alumnos generaron en la fase de diseño del prototipo software.

IV. EVALUACIÓN, CONCLUSIONES Y LÍNEAS FUTURAS

La evaluación de esta innovación se ha realizado en dos dimensiones: participación de los alumnos y su relación con el abandono en la asignatura y resultados docentes reflejados con las notas. Un tercer elemento de la evaluación es lo relacionado al logro de los objetivos planteados en la asignatura, que por lo descrito se consideran alcanzados al haber generado un prototipo de software excelentemente valorado por el cliente.

En trabajos previos como [14] se denota la importancia pedagógica de aprender a través de la práctica, con el estudiante recibiendo retroalimentación acerca de su actuación. El método de trabajo dio como resultado una mayor implicación de los alumnos en las tareas programadas en la asignatura, quienes participaron en más del 90% del total de actividades programadas.

Además, todos los alumnos inscritos llegaron hasta el final de la asignatura, evento único en los últimos cinco cursos académicos. Es decir, se logró anular la deserción en la asignatura ya que la totalidad de la clase se implicó y participó hasta el final de la asignatura.

El docente piensa que el proyecto solidario, las visitas frecuentes a las dependencias del socio comunitario y la

participación activa del socio comunitario fueron determinantes para el resultado antes descrito. Las notas finales mejoraron en un 10% respecto al año anterior, también se atribuye este resultado al tipo de metodología de trabajo usada.

Como conclusiones de la aplicación de la metodología A+S se puede destacar lo siguiente:

- La innovación ha permitido a los estudiantes acceso al ejercicio profesional y a la acción comunitaria, esto último ha resultado altamente atractivo a los alumnos y explica el nivel de compromiso que ha llevado a que se mantengan en la clase durante todo el curso académico.
- La innovación ha permitido poner en práctica las técnicas de buena programación (de calidad) en el marco de un proceso ágil de desarrollo con entregas frecuentes (iteraciones de 2-3 semanas) y feedback frecuente del cliente.
- La innovación ha permitido que los alumnos desarrollen habilidades interpersonales como, por ejemplo, el trato con personas discapacitadas o profesionales del área de salud que tratan deficiencias del lenguaje.
- La unión del A+S y actividades basadas en el desarrollo ágil ha permitido disminuir el abandono en la asignatura, dado al dinamismo del proyecto y la cercanía con un proyecto real enmarcado en la Responsabilidad social.

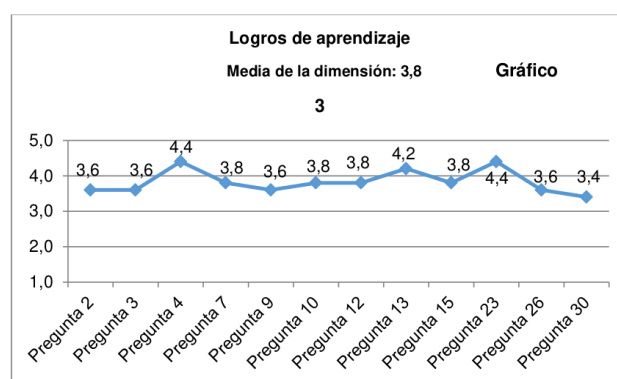


Fig. 3. Gráfico de la dimensión Logro de aprendizaje.

Como conclusiones también se puede destacar lo siguiente, este tipo de innovaciones está generando buenos resultados entre los docentes en la UCN y particularmente en el desarrollo docente del autor principal de este documento, como lo detalla la figura 3. En la figura 3 se presenta la dimensión “logros de aprendizaje” de la evaluación hecha por el CIMET⁸, correspondiente al desempeño docente. Se evidencia en este gráfico que los logros están próximos al valor 4 (media 3,8 con valor máximo de 5 - valor superior al obtenido en los años anteriores que fue de media un 3,4). Esto

⁶ <https://developer.android.com/studio/index.html>

⁷ <https://www.sqlite.org/>

⁸ <http://www.cimet.ucn.cl/sitio/>

es el resultado de la evaluación hecha por los alumnos de los últimos cursos, y muestra que el uso de las metodologías activas está siendo bien apreciado por los alumnos.

A. Líneas futuras

Como líneas futuras se plantea actividades concretas como las siguientes:

- Diseñar nuevas iniciativas basadas en A+S en asignaturas relacionadas con el desarrollo de aplicaciones software, para que mediante actividades didácticas en este marco se pueda acercar a los alumnos a la actividad profesional más solidaria y comprometida con el desarrollo de habilidades blandas como trabajo solidario o trabajo en grupo.
- Se plantea también enfocar las actividades didácticas a la mejora significativa del desempeño de los alumnos con la Responsabilidad social que pretende fomentar la UCN y como punto importante.
- Se pretende seguir aplicando metodologías activas para lograr que otros docentes se involucren e incorporen a la práctica de este tipo de docencia que es más acorde a las capacidades y forma de pensar de los estudiantes del siglo XXI.
- Incorporar la calidad de los aprendizajes de los estudiantes a la carrera IECI.

AGRADECIMIENTOS

Los autores agradecen la colaboración del personal de la Oficina para la Integración de personas con Discapacidad OID-Antofagasta por el apoyo incondicional a este proyecto, especialmente agradecer a Paola Jaque la fonoaudióloga de la OID-Antofagasta que ha sido el cliente, usuario y un miembro más del equipo de trabajo para lograr que este proyecto piloto haya tenido un excelente resultado en este curso académico.

REFERENCES

- [1] V. Flores, P. Lara & MC Gaya. "Project Based Engineering School, Una escuela conectada con un mundo profesional sostenible", X Jornadas Universitarias de Innovación Universitaria - JENUI 2013, pp 476-484. Villaviciosa de Odón, España, 2013.
- [2] O. Jerez. "Aprendizaje Activo, Diversidad e Inclusión. enfoque, metodologías y recomendaciones para su implementación. Ediciones Universidad de Chile, 2015.
- [3] Y. Sedelmaier & D. Landes. "Active and Inductive Learning in Software Engineering Education. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, pp. 418-427. IEEE, 5, 2015.
- [4] S. Bell. "Project-based learning for the 21st century: Skills for the future". The Clearing House, 83(2), pp. 39-43. 2010.
- [5] P. Chaverri. El aprendizaje en servicio en ULACIT: conceptos, experiencias y retos. Universidad Latinoamericana de Ciencias y Tecnología. Costa Rica. 2015.
- [6] J. Biggs & C. Tang. "Teaching for Quality Learning at University". Open University Press, 2007.
- [7] M. Fowler & J. Highsmith. The agile manifesto. Software Development, 9, pp. 28-35. 2001.
- [8] C. Tosun, C. Senocak. "The effects of problem-based learning on metacognitive awareness and attitudes toward chemistry of prospective teachers with different academic backgrounds". Australian Journal of Teacher Education, vol. 38, no 3, pp. 4-8. 2013.

- [9] V. Flores & Y. Gomez. "Aplicando metodologías activas en la enseñanza de Ingeniería del Software en Ingeniería Informática". Revista VEAP-RITA, IEEE Eds. Vol. 5(1), pp 1-10. 2017.
- [10] D. G. Lamar, P. F. Miaja, Arias, M., Rodríguez, A., Rodríguez, M., Vázquez, A., ... & Sebastián, J.. Experiences in the application of project-based learning in a switching-mode power supplies course. IEEE Transactions on Education, 55(1), 69-77. 2012.
- [11] M. Singh. U-SCRUM: An agile methodology for promoting usability. In Agile, 2008. AGILE'08. IEEE, pp. 555-560. 2008.
- [12] Lamar, D. G., Miaja, P. F., Arias, M., Rodríguez, A., Rodríguez, M., Vázquez, A., ... & Sebastián, J. Experiences in the application of project-based learning in a switching-mode power supplies course. IEEE Transactions on Education, 55(1), 69-77. 2012.
- [13] Ch. Jouannet, M.H. Salas, M.A. Contreras. Modelo de implementación de Aprendizaje Servicio (A+S) en la UC. Una experiencia que impacta positivamente en la formación profesional integral. Revista Calidad en la Educación, 13. pp. 197-212. [On-line] <http://dx.doi.org/10.4067/S0718-45652013000200007>. 2013.
- [14] P.C. Blumenfeld, E.Soloway, R.W. Marx, J.S. Krajcik, M. Guzdial, A. Palincsar, "Motivating project-based learning: Sustaining the doing supporting the learning". Educational Psychologist, vol. 26, no. 3, pp. 369-398, 1991.
- [15] V. Flores & J Gómez. "Applying Active Methodologies for Teaching Software Engineering in Computer Engineering". IEEE-RITA, vol 12(3), Springer Eds. pp. 147-155, 2017.
- [16] S. B. Cooper, J.H. Cripps, J.I. Reisman. Service-learning in deaf studies: Impact on the development of altruistic behaviors and social justice concern. American Annals of the Deaf, 157 (5), pp. 413-427. 2012.