

MEJORAS EN EL PROCESO DE ENSEÑANZA- APRENDIZAJE DE PROGRAMACIÓN UTILIZANDO METODOLOGÍAS DE LA INDUSTRIA DEL SOFTWARE

Aubin, Verónica Blautzik, Leonardo Dejean, Gustavo
Universidad Nacional de la Matanza

Abstract

Diversos estudios han demostrado los beneficios de introducir la programación de a pares en el desarrollo de software. Basado en esto, la propuesta de trasladar esta técnica al aula busca aportar valores didácticos que incrementen las habilidades del estudiante en su desempeño académico y profesional.

*Este trabajo se basa en la experiencia realizada en la cátedra de Programación Avanzada de la Universidad Nacional de la Matanza (UNLaM). Se abordan las ventajas derivadas de la aplicación en el aula de la **programación de a pares**, que es uno de los aspectos significativos de las metodologías XP (Extreme Programming), y se muestran las mejoras obtenidas en el rendimiento académico de los alumnos. Este trabajo hace hincapié en los resultados obtenidos al extender esta práctica al momento de la evaluación. Se presenta la "evaluación de a pares" como una extensión natural, no traumática, del trabajo en el aula.*

Palabras

Metodología de enseñanza, enseñanza-aprendizaje programación, Enseñanza de la Ingeniería, Competencias, ABP, Aprendizaje basado en problemas, Metodologías activas,

1. Introducción

Una de las primeras dificultades que surgen en el aula en un curso de programación avanzada, es la escasa experiencia y predisposición de los alumnos a trabajar en equipos. Es una práctica frecuente entre los estudiantes que, ante consignas que serán resueltas “en equipo”, se sortee al encargado de realizarlo mientras que cada uno de los restantes integrantes del equipo descansa hasta que le toque otro trabajo en el sorteo. Claro es que nada tiene esto que ver con un verdadero “trabajo en equipo”.

Citando a Litwin [1]: “El trabajo en equipo permite que los estudiantes, se organicen, distribuyan tareas, formulen y confronten

hipótesis, deliberen sobre ideas diferentes y cada uno de ellos se transforme en un recurso para los demás. Ellos tendrán que reconocer, asumir y practicar las responsabilidades que cada uno tiene para el desarrollo del trabajo. Los docentes podrán favorecer estas actividades alentando a la autoevaluación, para reconocer las tareas que se llevaron a cabo en la búsqueda por solucionar el problema planteado.”

Las metodologías ágiles de desarrollo de software, y en particular Extreme Programming (XP), constituyen una de las tendencias de mayor impacto en la industria del desarrollo de software en la última década, gracias a su enfoque centrado en la generación temprana de valor y en su acento en el aspecto humano del desarrollo de software [2]. Desde los inicios de estas metodologías surgió el interés de incorporar esta nueva mirada como una forma de enriquecer la formación de los futuros ingenieros de software. Además, es necesario acercar de manera temprana al aula procedimientos que se utilizan con éxito en la industria del software.

XP, formulada por Kent Beck [3], se diferencia del resto de las metodologías por su énfasis en la adaptabilidad. La metodología está diseñada para ofrecer el software que el usuario necesita, en el momento en que lo necesite. El éxito de la metodología se basa en potenciar las relaciones interpersonales, promoviendo el trabajo en equipo, el aprendizaje de los desarrolladores, y un ambiente de trabajo cordial. Toda la producción de código, en

esta metodología, debe realizarse con trabajo en parejas de programadores.

Según Cockburn y Williams en un estudio realizado para identificar los costos y beneficios de la programación en parejas [4], también citado y comentado en [5], las principales ventajas de introducir este estilo de programación son:

- Detección temprana de errores por la inspección continua del código. Por consiguiente, la tasa de errores del producto final es más baja, los diseños son mejores y el tamaño del código es menor.
- Se promueve la continua discusión de ideas de los programadores, los problemas de programación se resuelven más rápido, se posibilita la transferencia de conocimientos de programación entre los miembros del equipo, varias personas entienden las diferentes partes del sistema, los programadores conversan mejorando así el flujo de información y la dinámica del equipo.
- Los programadores disfrutan más su trabajo.

Dichos beneficios se hacen más evidentes a medida que se profundiza en estas prácticas de programación en parejas.

Todo lo antes mencionado es llevado a la práctica durante la cursada de la cátedra Programación Avanzada de la UNLaM. Esta metodología se adapta y aplica en cada problema a resolver durante el curso, los problemas son seleccionados del banco de problemas de la O.I.A (olimpiada informática argentina) [6] [7] los cuales se encuentran clasificados en tres niveles, según su dificultad. Se replica la misma metodología al momento de las evaluaciones parciales. Extender esta práctica a las evaluaciones, permite elevar el nivel de exigencia de las mismas,

incrementando la complejidad de los problemas a resolver. Los beneficios obtenidos se analizan en profundidad más adelante.

El artículo está organizado de la siguiente manera: en la sección 2 se presenta el marco teórico y se describen dos cambios culturales necesarios para el éxito de la metodología. En secciones 3, 4 y 5 se exponen los resultados, discusiones y conclusiones respectivamente.

2. Elementos del Trabajo y metodología

2.1. Análisis Teórico

Programación Extrema (XP).

En [8] se define XP como un proceso iterativo e incremental con pruebas unitarias continuas y entregas frecuentes. El cliente o un representante del cliente son integrados al equipo de desarrollo. Recomienda que el desarrollo de las funciones del producto sea realizado por dos personas en el mismo puesto (programación por pares). Antes de incorporar nueva funcionalidad, se deben corregir todos los defectos encontrados. Constantemente, se llevan a cabo pruebas de regresión a fin de detectar los posibles errores.

Diversos estudios han demostrado los beneficios de introducir la programación de a pares [9] [10] [11] [12] en las aulas, no sólo por la conveniencia de enseñar métodos que han sido exitosos en la industria, sino porque aportan valores didácticos que incrementan las habilidades del estudiante en su desempeño académico y profesional.

Las metodologías ágiles en el aula y su aplicación en la formación de conocimiento

Se plantea aquí cuales son las líneas básicas de una nueva propuesta metodológica que intenta alejarse de modelos didácticos tradicionales para proponer una metodología activa basada en la interacción

entre iguales, como eje principal del proceso de enseñanza-aprendizaje, que se apoya en los pilares básicos de la cooperación, convivencia y diálogo.

Considerando lo expuesto en [13], hoy en día se exige a los docentes la implementación, en la práctica diaria, de una metodología activa, que tenga en cuenta la diversidad. Sin embargo, no existe claridad acerca de lo que significa esto y sobre todo cómo llevarlo a cabo en el trabajo diario en el aula. En general predominan las prácticas tradicionales propias de la transmisión de profesor a alumno.

¿Cómo hacer que el trabajo en el aula genere las condiciones adecuadas para que el estudiante sea el centro real de la producción de conocimientos? Las metodologías ágiles de desarrollo de software, y en particular Extreme Programming (XP), parecen dar respuesta a esta pregunta, poniendo el acento en el aspecto humano del desarrollo de software.

Desde un punto de vista pedagógico surge el interés por esta nueva mirada como una forma de enriquecer la formación de los futuros ingenieros de software. Además es necesario acercar de manera temprana al aula procedimientos que se utilizan con éxito en la industria del software.

2.2. Aspectos Fundamentales de la Aplicación de XP en el Aula

El curso de Programación Avanzada, donde se aplican las prácticas metodológicas expuestas, tiene una carga horaria de 10 horas semanales con una modalidad cuatrimestral, sumando 160 hs. Durante la cursada los alumnos deben resolver una guía de ejercicios de 1600 líneas de código (LOC). Resolver entre 4 o 5 trabajos prácticos especiales de entrega obligatoria, que totalizan aproximadamente 1000 LOC, más una aplicación Cliente - Servidor de aproximadamente 1200, 1500 LOC, dependiendo de de las funcionalidades

solicitadas. Además, para completar la aprobación de la materia deben rendir dos evaluaciones parciales.

La preparación de la prueba primero

Un cambio cultural impuesto a los alumnos a esta altura de su carrera y que forma parte de las metodologías XP, es “La preparación de la prueba primero”. Antes de encarar cualquier solución de un problema o consigna, luego de la etapa de análisis y antes de la codificación, deben tener preparado el Lote de Pruebas y/o el programa probador o el método test donde se realizará la prueba unitaria en el caso de un método de una clase.

Todo por supuesto dentro de un ciclo de vida iterativo e incremental.

La preparación temprana de la prueba mejora el análisis, hace que se comprenda mejor el problema pensando en los casos particulares y de fatiga. La prueba se hace inmediatamente después de la codificación de cada incremento o iteración. Al hacer la prueba de ésta manera se descubren errores que se corrigen con el código más “fresco”.

Como parte importante del Lote de Pruebas se considera la documentación del mismo. En ésta, se describen cada uno de los casos de prueba, con su correspondiente objetivo o comportamiento particular a testear, los datos de entrada y la salida esperada.

Estimaciones y métricas - PSP

Otro cambio cultural impuesto, y quizás el de más difícil aceptación, sea el de la toma permanente de métricas. Los pares deben estimar y medir cada incremento del ciclo de vida [14].

“Los grupos nunca son homogéneos. Se diferencian por sexo, por intereses, por experiencias construidas a lo largo de su trayectoria educativa o sus trayectorias familiares” [1]. PSP le permite al estudiante conocer y establecer su ritmo de trabajo para alcanzar la meta fijada.

Programación en Pares

Los estudios demuestran [15] que, contra lo que pudiera pensarse, dos programadores son más eficientes que uno solo para una tarea determinada (el resultado es mayor que la suma de las partes).

Dos desarrolladores trabajan en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Los roles se intercambian.

Uno por momentos es el "conductor" y tiene el control de la generación de código. El otro es el "observador", examina el trabajo del conductor de manera activa y continua. Ve los defectos, pensando en alternativas, evaluando los recursos utilizados, tiene en cuenta las implicaciones estratégicas de la obra en ejecución y hace preguntas. El observador identifica la táctica y las deficiencias estratégicas en el trabajo.

Las metodologías XP enfatizan que la comunicación entre los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. Dichos estándares son determinados en clase.

La selección de los integrantes de cada par queda a cargo de los mismos alumnos. Esta selección se hace de manera natural y espontánea. Una acertada elección del compañero es clave para el buen desempeño del par. E. Faria plantea en [16], que la conformación de los grupos de estudiantes con las mismas capacidades, hacen que su aprendizaje y producción sea más efectiva, que si fuera con capacidades diferentes.

¿Por qué funciona?

Dos cabezas piensan más que una. Mientras uno se concentra en escribir el código, el otro puede percatarse de potenciales problemas al tener una visión más panorámica del asunto. Esto es similar a lo

que ocurre en el caso del conductor y el navegante: mientras uno conduce el otro consulta el mapa y dirige. Balance de experiencia experto-aprendiz. El programador más experimentado transmite su experiencia al novato y lo hace aplicar su experiencia por medio de la práctica supervisada.

Según Zabalza [17] "Saber manejarse en el marco de condiciones y características que presenta el grupo de estudiantes con el que le toque trabajar (jóvenes de los primeros cursos, estudiante adultos, etc.), y ser capaz de estimularles a aprender, pensar y trabajar en grupo. Transmitirles la pasión por el conocimiento, por el rigor científico, por mantenerse siempre al día, etc." Y es en este sentido que se realiza la búsqueda de nuevas técnicas de transmisión-generación de conocimiento.

3. Resultados

En esta sección mostramos los resultados obtenidos durante los años (2010, 2013) de aplicación de la metodología descrita en el curso Programación Avanzada de la UNLaM. Se explica también el método de evaluación de los alumnos así como la impresión de docentes y alumnos respecto de los cambios realizados al curso clásico.

3.1. Efectos de la Programación de a pares en el rendimiento estudiantil

La tabla 1 y el gráfico 1 muestran las mejoras obtenidas en los porcentajes de alumnos aprobados a lo largo de los últimos años.

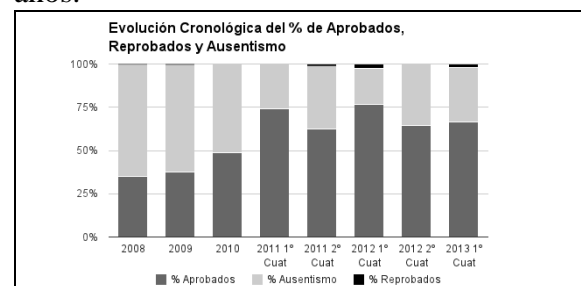


Gráfico 1 Evolución cronológica de la proporción de alumnos aprobados antes y después de aplicar programación de a pares

Tabla1 Cantidad de alumnos aprobados

Año	Metodología	Modalidad de Cursada	Inscriptos	Aprobados	% Apr.	Reprobados	% Repr.	Ausentes	% Aus.
2008	Tradicional	Anual	253	89	35%	1	0%	163	64%
2009	Tradicional	Anual	217	81	37%	1	0%	135	62%
2010	XP	Anual	193	94	49%	0	0%	99	51%
2011 1º Cuat	XP	Cuatrimestral	96	71	74%	0	0%	25	26%
2011 2º Cuat	XP	Cuatrimestral	88	55	63%	1	1%	32	36%
2012 1º Cuat	XP	Cuatrimestral	77	59	77%	2	3%	16	21%
2012 2º Cuat	XP	Cuatrimestral	62	40	65%	0	0%	22	35%
2013 1º Cuat	XP	Cuatrimestral	93	62	67%	2	2%	29	31%

Los alumnos considerados en este análisis pertenecen a tres cursos, turno mañana, tarde y noche de la cátedra de Programación Avanzada. Curso obligatorio en el tercer año de la carrera de ingeniería en informática de la Universidad Nacional de La Matanza.

3.2. Reacción de los alumnos respecto de la aplicación de la programación de a pares

Al finalizar cada curso los alumnos son encuestados sobre diversos aspectos y uno de ellos es la programación de a pares.

En la encuesta: 1 significa insatisfacción total y 4 satisfacción total. Como se puede apreciar en los gráficos 2, 3 y 4 su valoración es muy positiva.

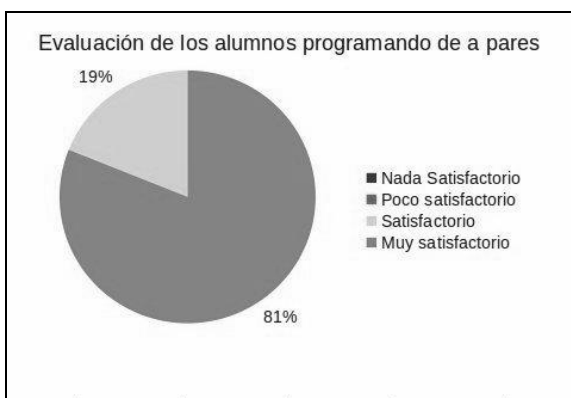


Gráfico 2 Evaluación de los alumnos programando de a pares.

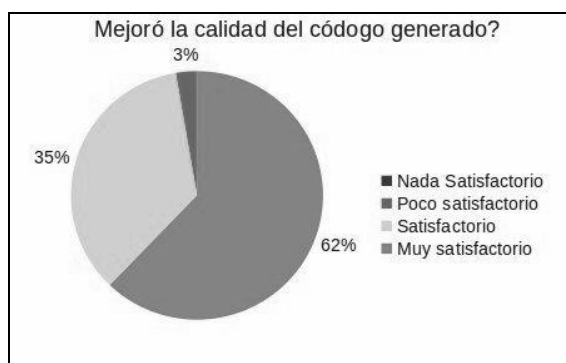


Gráfico 3 Evaluación de los alumnos respecto al código generado programando de a pares.

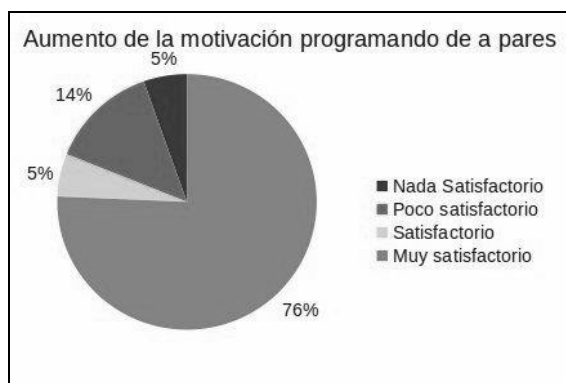


Gráfico 4 Evaluación de los alumnos respecto a la motivación programando de a pares.

3.3. ¿Evaluación de a pares?

La evaluación se lleva a cabo en alguno de los laboratorios de la facultad, los alumnos se ubican de a dos por PC. Se les entrega un problema de la OIA de nivel II o III, o un enunciado de nivel similar generado por los docentes de la cátedra. A modo de ejemplo podemos citar: “Usando la Red de Subterráneos” o “Criadero de truchas” OIA - Certámenes Nacionales, Nivel III año 2007 y 2008 respectivamente. El problema lo deben resolver siguiendo la metodología mencionada anteriormente (estimaciones y métricas, preparar la prueba, etc.). Cada PC cuenta con todo el software necesario para

resolver dicho problema (entorno Eclipse y lenguaje Java) y las herramientas adecuadas para cumplir con la metodología. Al finalizar el tiempo de la evaluación, generalmente de 2 a 3 horas, se le entrega a cada par un lote de pruebas confeccionado especialmente por el equipo docente. Con estos casos de prueba, cada par genera las salidas que servirán como un indicador al momento de evaluar el trabajo entregado. Se evalúa también la complejidad computacional del algoritmo que resuelve el problema, las estructuras de datos utilizadas, la eficiencia y la economía de recursos. Otro ítem evaluado es el la calidad del lote de prueba generado por los alumnos durante la evaluación, que contenga casos de fatiga y casos particulares e ingeniosos que muestran que el problema ha sido comprendido en su totalidad y ha sido resuelto cubriendo todos los requerimientos del mismo. El lote de pruebas que generan los alumnos durante la evaluación es un lote reducido, en comparación con el que generan habitualmente. Esto es para ajustar los tiempos del examen.

Se corrige con rúbrica. Dependiendo de cuál sea el problema, y a modo de ejemplo, en la tabla 2 se muestra una rúbrica básica utilizada.

Tabla2 Ejemplo de Rúbrica de corrección

Ítems a evaluar	Valoración	Nota (%)
Calidad del lote de pruebas o del programa probador, según corresponda.	1	80
Documentación del lote de pruebas.	1	80
Planilla de estimaciones y métricas.	1	90
Resultados esperados versus resultados obtenidos.(de acuerdo al lote de pruebas)	4	75
Calidad del código y/o estilo de programación.	2	50
Complejidad computacional. Eficiencia del algoritmo implementado.	1	100
Nota Final		7.50

$$\text{Nota final} = \sum (\text{valoración} * \text{Nota} / 100)$$

Aclaración: que funcione al menos el 60 % del lote de pruebas es condición necesaria para aprobar la evaluación.

Contrariamente a lo que a priori podría inferirse, la evaluación de a pares (para la resolución de problemas) no resulta en una ventaja en favor de los alumnos poco aplicados. Podría pensarse que alumnos con poca preparación, podrían aprobar la materia de manera parasitaria, colgados de otro compañero solidario, pero esto no sucede. Solo funcionan los pares que ya han trabajado juntos, se conocen, se entienden y complementan. En los casos de pares improvisados, el resultado de la evaluación no ha sido exitoso para ninguno de los integrantes del par. Sin embargo los pares consolidados trabajan de manera natural, y si bien el éxito de la evaluación dependerá de muchos factores, la experiencia en sí misma resulta muy enriquecedora. Complementaria a la evaluación práctica, cada alumno es evaluado de manera individual en los conceptos puramente teóricos de la materia. Los resultados son, en un alto porcentaje (más del 50%), equivalentes para cada uno de los integrantes del par. Rinden de manera individual de la misma forma que lo hacen de a pares. Un buen trabajo de conjunto, por lo tanto, casi siempre resulta en buenos rendimientos individuales. Estos resultados son las más relevantes del presente trabajo. Habiendo planteado una metodología para el trabajo en clase y extendiendo ésta práctica a las evaluaciones, se ha conseguido elevar el nivel de la cátedra y mejorar notablemente la proporción de aprobados.

3.4. Evaluación por parte del docente.

Trabajar en el aula de forma cooperativa no anula el trabajo individual ni la evaluación individual. Nadie puede aprender por otro y el aprendizaje es, por tanto, una responsabilidad individual. En lo que respecta a la evaluación de las tareas realizadas en grupo, han de evaluarse de forma global, de manera que todos los

componentes obtengan la misma calificación. La evaluación del trabajo individual contará un 30% de la calificación y, el trabajo de a pares un 70% de la misma. Con esta doble evaluación se pretende que ambos integrantes del par sean evaluados de un forma justa y evitar así, que algún alumno se aproveche del trabajo realizado por el compañero sin que él haya sido partícipe del mismo.

Las relaciones entre los contenidos teóricos y la práctica profesional se amalgaman de manera natural, y son los alumnos quienes, mediante la aplicación constante y progresiva de metodologías ágiles, de procesos iterativos incrementales y del trabajo en equipo, o de a pares, generan nuevas formas de conocimiento y provocan una retroalimentación de la que se nutren también los docentes.

A modo de resumen se enumeran en la tabla 3 los puntos a favor y en contra de la aplicación de esta metodología tomando en consideración las miradas de los alumnos y los docentes.

Tabla3 puntos a favor y en contra de la aplicación de esta metodología.

	Puntos a favor	Puntos en contra
Alumnos	Reduce la cantidad de defectos en el código.	Requiere un cambio cultural.
	Mejora la calidad de diseño.	Un par no afianzado demora en adquirir competencias.
	Comparte conocimiento en equipo.	Resistencia de algunos alumnos a trabajar de a pares hasta que ven el resultado de sus compañeros.
	Se despierta la camaradería	Diferencias en el diseño pueden provocar conflictos en el par, pero es parte de lo que tienen que aprender a resolver discutiendo y fundamentando su postura.
	Mayor concentración	
	Auto disciplina.	
	Baja el nivel de stress frente a la evaluación.	

Docente	Se aumenta la complejidad de los problemas.	Los profesores deben conocer y haber aplicado estas técnicas en su trabajo para poder transmitir las mejor.
	Menor tiempo en la resolución.	Número impar de de alumnos.
	Buen clima en el aula.	Requiere un cambio cultural.
	No es difícil de transmitir la metodología de trabajo de la programación de pares y se ve el cambio de roles dentro del par.	Exige una esmerada formulación de cada ejercicio o en su defecto, recurrir a los bancos de problemas de la OIA y similares

4. Discusión

Estas prácticas están siendo aplicadas y consideradas en la Universidad Nacional de la Matanza, como Modelo aplicable en las asignaturas de programación correlativas a Programación Avanzada. Al ser una Metodología que toma técnicas y Metodologías propias de la Industria del Software, le da un valor agregado a su utilización en el aula.

Una modificación necesaria para su aplicación exitosa en el aula fue el cambio de escala. En particular, las Metodología XP es usada frecuentemente en proyectos de software de mediana y gran envergadura, aquí se lo adapto a una escala muy pequeña. Se lo adapta a ejercicios que pueden tener desde 100 LOC a 1500 LOC. Una iteración típica de un alumno, puede implicar solo el desarrollo de uno o varios métodos y que le puede implicar desde unos pocos minutos a varias horas.

Se obtienen varios beneficios adicionales al usar las técnicas PSP en el aula, algunas de ellas son: acopio de datos estadísticos, como ser la tasa de fallos que tienen los alumnos cada 100 LOC; la cantidad de LOC que pueden escribir por hora y discernida por cada nivel de dificultad, el tamaño total de cada una de las guías de trabajos prácticos, de los trabajos prácticos especiales y de cada evaluación.

Es importante destacar que la Metodología activa utilizada es orientada a las Competencias del egresado, lo que se está midiendo es lo que el alumno “sabe hacer” y no solamente lo que sabe. En este sentido, tiene un amplio campo de aplicaciones, como ser su utilización para nuevos planes curriculares donde los programas no estén centrados en contenidos sino en Competencias o en carreras con modalidad no presencial.

El presente estudio utiliza datos estadísticos y experiencias que van desde el año 2008 al 2013; en esos años el plan de estudios sufrió un doble cambio, por un lado se paso de la modalidad anual a la modalidad cuatrimestral. Esto implicó un aporte positivo a los resultados aquí mostrados y discernir en que porcentaje exacto influyo quedo fuera del alcance del presente trabajo. Por otra parte, se cambio en una pequeña parte los contenidos mínimos, manteniendo la misma carga horaria para el plan viejo y aumentándola para el plan nuevo. Cómo ya se dijo, por ser una cátedra netamente orientada a las competencias, estos cambios prácticamente no tuvieron influencias en los resultados.

5. Conclusión

Se han logrado avances significativos en cuanto al trabajo cooperativo y a la adaptación a los cambios. Los alumnos son capaces de enfrentarse a situaciones problemáticas nuevas y lograr resolverlas exitosamente. En lo que respecta a las evaluaciones, éstas son tomadas por los alumnos de manera menos traumáticas y con menos presión. Cada evaluación es claramente una experiencia motivadora y enriquecedora. La prueba continua, las estimaciones, registro de métricas y el trabajo de a pares hace que los alumnos conozcan antes que el docente el resultado de su propio trabajo. Se ha mejorado considerablemente el ambiente en el aula, los alumnos investigan y cooperan para lograr los objetivos y consignas planteadas. Todo esto ha contribuido a mejorar la

motivación, a disminuir sensiblemente la deserción de quienes comienzan la cursada y finalmente a aumentar la tasa de aprobados al final del curso. El hecho que la Metodología aquí propuesta, haya sorteado con éxito el cambio de Modalidad de anual a cuatrimestral y el cambio de plan de estudio, demuestra cierta su robustez.

Agradecimientos

A las autoridades del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza, por los cambios puestos en marcha durante estos últimos años. A Julio Crispino por su meticulosa toma de métricas en el aula.

Referencias

- [1] E. Litwin. “El oficio de enseñar”. Paidós. Bs. As., Argentina. 2008.
- [2] A. A. Villena Moya. “Un modelo empírico de enseñanza de las metodologías ágiles”. Tesis para optar al grado de Magíster en Ciencias, Mención Computación. Universidad de Chile. Santiago de Chile – Enero 2008
- [3] K. Beck. “Extreme Programming Explained”. Embrace Change. Reading, Massachusetts: Addison-Wesley. 2000.
- [4] A. Cockbun, L. Williams. " The Costs and Benefits of Pair Programming". Humans and Technology Technical Report. 2000.
- [5] P. Letelier, C. Penadés. “Metodologías ágiles para el desarrollo de Software: eXtreme Programming (XP)”, Universidad Politécnica de Valencia 2006. Fecha de consulta: 30 de julio de 2012. URLs: <http://www.willydev.net/descargas/masyxp.pdf>.
- [6] G. Dejean, “Uso de nuevas métricas en la Gestión de carreras de ingeniería y licenciaturas en Informática”, Congreso Mundial de Ingeniería. Buenos Aires – Argentina- octubre, 2010.
- [7] Olimpiada Informática Argentina O.I.A Fecha de consulta: 1 de agosto de 2012. URLs: [www/oia.org.ar](http://www.oia.org.ar).
- [8] K. M. Calo, E. Estevez, P. Fillottrani. “Un Framework para Evaluación de Metodologías Ágiles”. Fecha de consulta: 1 de agosto de 2012. URLs: <http://ebookbrowse.com/un-framework-para-evaluacion-de-metodologias-agiles-pdf-d283940617>

- [9] L. Williams, R. L. Upchurch. "In Support of Student Pair-Programming". ACM SIGCSE Conference for Computer Science Educators. February. 2001.
- [10] C. Bravo, M.A. Redondo, M. Ortega, "Aprendizaje en grupo de la programación mediante técnicas de colaboración distribuida en tiempo real". In Proceedings of V Congreso Interacción Persona Ordenador, Lleida, Spain. 351-357. 2004.
- [11] S.F Freeman, B.K. Jaeger, J.C. Brougham. "Pair programming: More learning and less anxiety in a first programming course". 2004. Fecha de consulta: 1 de agosto de 2012. URLs: http://www.jonaschalk.neu.edu/search_archives/documents/pairprog03.pdf .
- [12] N. Salleh, E. Mendes, J. Grundy "Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review". IEEE Transactions on Software Engineering. Volume 37 Issue 4, Pages 509-525. 2011
- [13] I. Donaire Castillo, J. Gallardo Arrebola, S. Pilar Macías Aguado. "Nuevas Metodologías En El Aula: Aprendizaje Cooperativo". Revista Digital: Práctica Docente. Nº 3 Cep De Granada. ISSN: 1885-6667. DI: GR-2475/05 Julio/Septiembre. 2006.
- [14] G. Dejean, V. Aubin, L. Blautzik. "El Uso De Técnicas PSP Para El Logro De Competencias" Las Jornadas Chilenas de Computación (JCC) 2011. XIII Congreso Chileno de Educación Superior en Computación (CCESC'2011). 2011.
- [15] L. Williams. "Experimenting with industry's pairprogramming model in the computer science classroom". Computer Science Education. 2001.
- [16] E. S. J. de Faria, J. M. Adan-Coello, K. Yamanaka. "Forming groups for collaborative learning in introductory computer programming courses based on students programming styles: An empirical study". In 36th ASEE/IEEE Frontiers in Education Conference, pages 6-11. 2006.
- [17] M. A. Zabalza. "La enseñanza universitaria". Nancea. España. 2001.

Datos de Contacto:

Universidad Nacional de La Matanza. Florencio Varela 1903 (B1754JEC) -- San Justo, Buenos Aires, Argentina . vaubin@unlam.edu.ar