

Realización de Torneos de programación como estrategia para la enseñanza y el aprendizaje de programación

Claudia Carina Fracchia, Pablo Kogan, Ana Alonso de Armiño, Ingrid Godoy,
Lidia López

Facultad de Informática, Universidad Nacional del Comahue,
Buenos Aires 1400, Neuquén
{carina.fracchia, pablo.kogan, ana.alonso, ingrid.godoy, lidia.lopez} @fi.uncoma.edu.ar

Resumen. La enseñanza de la programación tradicionalmente se ha efectuado como una actividad principalmente de carácter individual. Esto se ha ido modificando en los últimos años, adoptándose diferentes enfoques que priorizan las prácticas de aprendizaje colaborativo, aprendizaje basado en proyectos y el competitivo. Estos enfoques permiten crear o fortalecer los lazos entre los estudiantes, a la vez que aumenta la motivación y se mejora la comprensión y el aprendizaje a través del “hacer” en forma colaborativa. En la búsqueda de estrategias para la enseñanza de programación surgen los Torneos de programación como recursos prometedores. Se presentan resultados parciales desprendidos de la experiencia desarrollada en la Facultad de Informática, en el marco del proyecto de extensión “*Entrenamiento en la programación de la computadora a partir de una aplicación para competencias*”.

Palabras clave: Enseñanza de programación, Torneos de programación, Aprendizaje colaborativo, TIC

1 Introducción

Este trabajo se enmarca en la investigación realizada dentro de los proyectos de investigación “Simulación y Métodos Computacionales En Ciencias y Educación”, en la línea “Desarrollo y Uso de TIC”, “Aspectos Avanzados de la Programación de Computadoras” y “Agentes inteligentes en ambientes dinámicos”, los tres pertenecientes a la Facultad de Informática de la Universidad Nacional del Comahue (UNCo).

En el año 2013 la Facultad de Informática fue sede del Torneo Argentino de Programación (TAP 2013), realizado en el marco de la competencia ACM-ICPC¹ auspiciada por IBM a nivel internacional. La realización de este primer torneo permitió vislumbrar su gran potencial en la enseñanza de programación, dando origen a un proyecto de extensión denominado “*Entrenamiento en la programación de la computadora a partir de una aplicación para competencias*”. Su objetivo principal es

¹ACM-ICPC (ACM- International Collegiate Programming Contest).
<http://icpc.baylor.edu/>

la formación de una comunidad de programadores en la región a partir de la realización de torneos análogos, contemplando diferentes niveles de exigencia, acordes a quienes se inician en las tareas de programación.

Se persigue además la generación de espacios de intercambio y actualización de la comunidad docente de las escuelas medias, en relación a los nuevos paradigmas de la programación de computadoras y a los contenidos propios del nivel universitario, facilitando la incorporación de nuevos saberes en los contenidos de las asignaturas trabajadas. A través de la conformación de grupos de estudio de docentes y estudiantes de la escuela media y facultad, se busca abordar problemáticas del campo de la enseñanza de la programación de computadoras además de aumentar las expectativas de los estudiantes, mejorando su preparación para la inserción y permanencia en la universidad.

Para la primera etapa del proyecto se tuvo en cuenta una audiencia conformada por estudiantes de escuelas medias de la región, cuyos planes de estudio contemplan la asignatura programación. En principio se buscó reforzar conocimientos en los docentes participantes tendientes a facilitarles su actuación como entrenadores en las competencias de programación planificadas. Por esto se comenzó con la capacitación y la realización de actividades de divulgación sobre metodologías de enseñanza y paradigmas de programación.

1.1 Enseñar a programar a través de torneos de programación

Más de dos décadas de trabajo en materias introductorias a la programación nos han permitido vivenciar que la enseñanza y el aprendizaje de la programación no es una tarea sencilla, especialmente para los estudiantes novatos, que a pesar de ser nativos digitales, necesitan ser motivados de manera adecuada con el fin de lograr aprendizajes efectivos y duraderos.

Tradicionalmente la enseñanza de programación se ha efectuado como una actividad fundamentalmente individual; sin embargo en los últimos se han adoptado diferentes enfoques priorizando prácticas de aprendizaje colaborativo, basado en proyectos y aprendizaje competitivo. Si bien mediante este último se puede llevar a una mayor participación de los estudiantes, deben tenerse en cuenta las diferencias en la motivación y los sentimientos generados entre los que resultan ganadores o perdedores. Para evitar la frustración de los estudiantes se debe enfocar la actividad más en el aprendizaje y sus aspectos lúdicos (disfrutar jugar más que ganar), y definir claramente los criterios de evaluación.

Hay varios sitios Web donde los estudiantes pueden descargar problemas de programación y presentar sus soluciones para ser evaluados de manera automática; uno de los más populares es "UVA on-line Judge" (<http://uva.onlinejudge.org/>). Su objetivo es capacitar a los usuarios a medida que participan en competencias de programación abiertas al mundo entero. Cuenta con una gran cantidad de problemas, siendo utilizado tanto por estudiantes como por docentes de programación.

Este sistema sirvió de base a EduJudge², sistema distribuido para el aprendizaje de la programación, que cuenta además con un repositorio de objetos de aprendizaje y

² <http://eduvalab.uva.es/en/projects/edujudge-project>

puede ser integrado a la plataforma Moodle mediante la actividad QUESTOURnament³. Este último recurso permite a los profesores configurar entornos de trabajo (individuales/grupales) donde se habilita un conjunto de desafíos intelectuales con el fin de ser resueltos por los estudiantes en un tiempo limitado. Posteriormente el estudiante es recompensado en función de la calidad de su trabajo, de acuerdo a la adecuación del reto propuesto y la evaluación de las respuestas presentadas por los otros estudiantes participantes. Una vez que un desafío es contestado correctamente, la puntuación comienza a disminuir de forma que el alumno primero en responder consigue la máxima puntuación.

Una ventaja del motor de evaluación de EduJudge es que permite proporcionar una evaluación matizada (no se limita a indicar correcto / incorrecto) y tiene en cuenta diferentes tipos de problemas de programación: un solo caso de prueba, varios casos de pruebas y problemas interactivos en los que la solución debe interactuar con otro programa. Este sistema es compatible con una amplia variedad de lenguajes en los que se destacan Pascal, Java y C.

El repositorio de problemas denominado crimsonHex ofrece una mejora en cuanto a la accesibilidad y usabilidad de enunciados de problemas a resolver mediante el diseño de programas. Dispone de una amplia variedad de problemas de programación que se pueden buscar por palabra clave, autor, tipo, nivel de instrucción, el nivel de dificultad y el lenguaje [1].

1.2 Motivación y aprendizaje

En [2] se presenta una iniciativa importante para entender los factores clave que afectan la motivación y el aprendizaje del estudiante en cursos de programación. Los factores pueden ser intrínsecos (se centran en los individuos, actitudes, expectativas, objetivos, emociones, etc.) o extrínsecos (ambientales), estos últimos derivan del entorno externo al aprendizaje. Algunos de los factores a tener cuenta son:

- **Actitud y expectativas individuales:** las personas tienen diferentes expectativas y niveles de confianza acerca de lo que son capaces de hacer, difieren sus percepciones y valores emocionales acerca de los resultados o recompensas obtenidas.
- **Objetivos y emociones:** las metas personales son importantes para determinar el rendimiento, estudios han establecido la correlación entre la motivación intrínseca y el compromiso con el logro de metas.
- **Dirección clara:** un aprendizaje efectivo en la educación superior se asocia con la percepción y certeza de los estudiantes sobre la tarea a realizar.
- **Recompensa y reconocimiento:** la teoría del refuerzo, hace hincapié en la relación entre la conducta y sus consecuencias. El reconocimiento implica un cierto grado de evaluación externa del desempeño que puede afectar la motivación de los estudiantes e influir posteriormente en el interés despertado por la tarea propuesta.
- **Castigo:** mientras que la motivación positiva como incentivos parece tener sentido, la gente responde a la expectativa de los castigos, también.

³Módulo QUESTOURnament. <https://moodle.org/mod/>

- **La presión y la competencia social:** la presión de los compañeros y la competencia también afectan el aprendizaje.
- **Auto-eficacia:** se refiere a lo que una persona cree que puede hacer en una tarea de aprendizaje particular y está relacionado con el rendimiento académico. Las personas con un alto nivel de auto-eficacia tienden a establecer metas altas y a presentar un buen desempeño.

Por lo descrito anteriormente resulta primordial destacar que la competencia es una instancia más de aprendizaje, por lo tanto la puntuación (alta/baja) y el resultado (ganador/vencedor) deberá tomarse como una cuestión secundaria. Se debe hacer hincapié en que el estudiante logre una adecuada comprensión de los conceptos trabajados, pueda reconocer sus errores y fallas, además de identificar las estrategias que pueden ayudarlo a superarse y mejorar su habilidad de programación. Hay muchas experiencias de torneos en programación con objetivos similares a los que perseguimos como es el descrito en [3], donde se destaca el fortalecimiento de los lazos creados entre los estudiantes, así como un alto nivel de motivación.

2 Descripción del Contexto y Actividades desarrolladas

Las experiencias se llevaron a cabo con estudiantes de las instituciones de nivel medio Colegio AMEN (Neuquén), CEM 14 (Fernández Oro) y Colegio Almafuerte (Cipolletti). En total participaron aproximadamente 60 estudiantes, en promedio fueron 20 alumnos por colegio. La participación de alumnos fue de carácter voluntario y en el caso del colegio CEM14, el límite de alumnos quedó establecido por la capacidad del transporte. En el caso de las otras instituciones, los estudiantes y docentes concurren por sus propios medios. Esto es interesante de destacar debido a que el número de alumnos se ha mantenido constante. En los encuentros presenciales se ha contado con la participación de todos los miembros del proyecto de extensión y han colaborado además docentes del Departamento de Programación y alumnos avanzados de nuestra Facultad.

Se planificaron y realizaron encuentros con los integrantes del proyecto donde se presentaron diferentes herramientas para trabajar la enseñanza de programación, resultando seleccionada la herramienta PSeInt, no antes trabajada en las instituciones mencionadas.

Se diseñó un aula virtual con el fin de realizar actividades previas a los encuentros presenciales, facilitando de esta manera la interacción entre los estudiantes de las distintas instituciones participantes. Esto resultó fundamental dado que las competencias se realizaban en grupos constituidos por 2 ó 3 estudiantes de cada colegio. La primera actividad dentro del aula virtual fue la participación en un foro de presentación, donde se los invitaba a colocar además de sus datos personales, sus preferencias musicales, deportivas, etc.

El aula contó con diferentes espacios destinados a la comunicación (información de actividades, evacuación de dudas, ocio) y a la presentación de contenidos con el objetivo de unificar los temas a trabajar. En la figura 1 se puede observar la interfaz inicial del curso mencionado.



Fig. 1. Espacio Virtual dentro de la plataforma Moodle de la UNCo.

Los datos relevados en la actividad de presentación permitieron diseñar dos primeras actividades que ponían a prueba sus conocimientos musicales y refranes populares. Estas actividades además de introducir a los estudiantes en un ambiente de competencias perseguían fomentar la relación e interacción entre los mismos, además de la participación.

3 Torneos de programación: favoreciendo la colaboración y la motivación

El concurso enfrenta a equipos conformados por estudiantes de tres instituciones de la región, que deben resolver problemas del mundo real en nivel de dificultad progresivo. Los competidores corren contra el reloj en una batalla donde además de la lógica y el empleo de estrategias, es importante la colaboración y comunicación de los integrantes de cada equipo. Una vez que un equipo resuelve el problema, un jurado de expertos es el encargado de evaluar el diseño y funcionamiento del algoritmo con un banco de prueba que acompaña el problema.

Previo a cada instancia de competencia, se realizó un repaso a través de exposiciones y ejercitación, en las que se hizo hincapié sobre uso de nombres significativos para identificadores, uso de comentarios e indentación para facilitar la lectura del código, correcta elección de estructuras de control, etc. Si bien en esta etapa la corrección se realiza como una caja negra, creemos importante resaltar y evidenciar las buenas prácticas de programación.

En caso que la solución propuesta sea correcta se le informa la puntuación obtenida al grupo y posteriormente el jurado en su devolución da pautas sobre como se podría mejorar el diseño de algoritmo. Cuando la solución es incorrecta, hay una penalización de tiempo que se calcula en forma automática, dando la posibilidad a cada equipo que siga intentando encontrar una solución en el tiempo que resta. El

equipo que resuelve más problemas en el menor tiempo acumulado es declarado ganador. A continuación se describen algunas de las actividades más relevantes llevadas a cabo.

3.1 Actividades de precalentamiento

Se realizaron actividades de precalentamiento previas a la realización de los torneos presenciales con el objetivo de introducir a los estudiantes en un ambiente de competencias y de repasar los conceptos que integran los problemas en la competencia presencial.

La metodología adoptada consistió en presentar el enunciado de un problema a resolver y mediante opciones de entrega de tareas se fijaba el plazo límite para realizar la resolución. En la devolución realizada a cada estudiante además de indicar si la respuesta era correcta, se realizaban comentarios acerca de como mejorar el diseño enviado. Posteriormente, un reloj reflejaba el tiempo restante para la finalización de la actividad. Una vez concluido el plazo de tiempo establecido se presentaba un ranking con las posiciones logradas por los estudiantes. Se dejó disponible además material de lectura complementario y foros abiertos para consultas y comentarios. En estas actividades los participantes que resolvían los problemas iban subiendo la solución a través de un recurso habilitado para tal fin, específicamente se habilitó una tarea, y quienes tenían dudas podían consultar a través de los foros.

Fue muy alentador ver que antes de que los profesores alcanzaran a responder muchas de las consultas planteadas otros participantes intentaban ayudar a sus compañeros, contándoles cual había sido la idea de ellos o aportando sus conocimientos. En la figura 2 se puede observar un ejemplo de actividades presentadas.

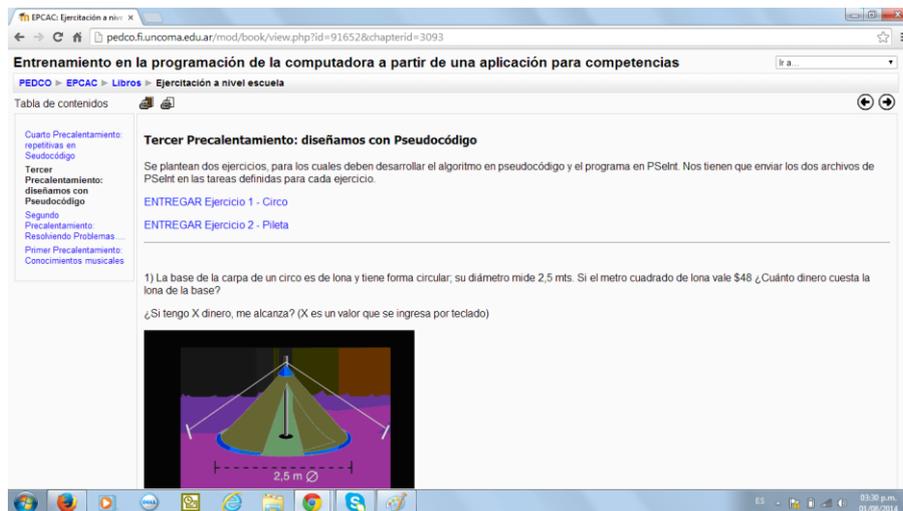


Fig.2. Actividades de precalentamiento.

3.2 Competencias presenciales

Se han realizado dos encuentros presenciales en espacios provistos por la Facultad de Informática. Para cada encuentro se planificaron charlas y actividades ejercitativas previas a las competencias, con el objetivo de unificar los contenidos a trabajar y ofrecer una instancia de repaso.

La metodología de trabajo contempló actividades grupales, donde cada grupo contaba con integrantes de las tres escuelas participantes. Al iniciar el torneo cada grupo debía decidir un nombre y un logo para representarlos. Esta actividad es una instancia de integración que permite a los estudiantes volver a acercarse a los compañeros e integrarse. Una vez realizada esta actividad se presentaba el problema a resolver. La solución debía diseñarse mediante la herramienta PSeInt, que al poseer una interfaz gráfica, amigable e intuitiva, permite trabajar con pseudocódigo de una manera muy simple

Tanto alumnos como docentes mostraron mucho entusiasmo durante estas jornadas. Se elaboró un reglamento con el fin de asegurar un sano desarrollo de las competencias y un buen clima de trabajo. Previo al inicio de la competencia se leía el reglamento y se entregada una copia a cada grupo participante.

3.3 Competencias online

Actualmente se está desarrollando una herramienta Web para facilitar la realización de competencias de programación online. Esta herramienta, bautizada con el nombre “Hornero”, fue diseñada teniendo como meta dos premisas: *“que se pueda programar/competir empleando cualquier lenguaje”* y *“que pueda jugar el que quiera”*. Para poder cumplir con la primera, se diseñó un protocolo de comunicación vía web service REST [4], lo que permite la interacción con lenguajes con interfaz de socket TCP. Esto abarca a la mayoría de los lenguajes actuales pero quedan afuera los lenguajes que no tienen interfaz de socket TCP. Para no excluirlos, se desarrolló un wrapper en C que encapsula la comunicación con el servidor y llama al archivo desarrollado por el programador enviando y recibiendo los parámetros por entrada y salida estándar. De esta manera se logra extender la comunicación a todos los lenguajes de programación.

Hasta el momento se ha trabajado con los lenguajes: java, python, php, c, c++, pascal, javascript, c#, ciao-prolog, visual basic, perl, bash, lisp, ruby y smalltalk. Para facilitar el desarrollo de la experiencia se ha incorporado además la posibilidad de trabajar con pseudocódigo, en este caso el empleado en la aplicación PSeInt usada en el marco del proyecto de extensión mencionado.

La segunda premisa se logra brindando la posibilidad de que puedan participar todas aquellas personas interesadas, que sepan programar, independientemente de su condición y nivel (estudiantes de los diferentes niveles educativos, profesores, expertos en programación, autodidactas, etc.). Es decir, competencias abiertas, sin restricciones de edad y título. Lo únicos requisitos son tener conexión a Internet y saber programar.

Para poder jugar primero hay que registrarse en la aplicación web. Posteriormente hay que inscribirse en el torneo que se desee, en donde se podrá visualizar los enunciados propuestos. A continuación, el participante tiene que descargar desde la aplicación el Stub⁴ de su lenguaje de preferencia y modificarlo en función de los problemas a resolver. Por último se ejecuta la resolución en la máquina del programador, recibiendo los parámetros y enviando la respuesta desde y hasta el servidor.

Como ventaja principal, cada participante (o equipo) se entera al instante si la solución enviada es correcta, de manera que puede corregirla y reenviarla tantas veces como sea necesario. Se ofrece retroalimentación inmediata, aunque en una primera instancia las respuestas enviadas serán del tipo binario (correcto/incorrecto).

La aplicación está siendo desarrollada en el lenguaje PHP, con base de datos MySQL y el servidor de Web Apache. Esta aplicación está prevista utilizarla en la instancia final del proyecto. El último encuentro presencial contempla además realizar una competencia empleando la aplicación Hornero (en modalidad online), propiciando de esta manera que los diferentes grupos participen e interactúen dentro del mismo espacio físico.

Los avances en las tecnologías de redes de colaboración han permitido la creación de poderosos entornos de colaboración, mediante el uso de sistemas groupware y arquitecturas distribuidas. La Programación Colaborativa favorece el trabajo conjunto y sincrónico de programadores, principalmente para aquellos distribuidos en lugares geográficos distantes [5]. Esto nos lleva a pensar en un escenario futuro donde puedan adicionarse a la aplicación desarrollada mecanismos que permitan y favorezcan el trabajo de programación colaborativo.

4 Resultados preliminares

En este proceso de aprendizaje dinámico llevado a cabo hasta el momento, la adquisición e intercambio de conocimientos estuvo afectada por diversos factores: *medio ambiente (contexto), diferencias individuales, presión social de los compañeros*, entre otros.

Los dos encuentros presenciales se desarrollaron en un salón de la UNCo, lo que resultó un nuevo contexto para los alumnos y docentes del nivel medio. Estas escuelas están situadas en diferentes ciudades, por lo tanto para algunos estudiantes el paseo a la ciudad de Neuquén tuvo un valor agregado.

Los estudiantes compartieron sus notebooks, dándose en algunos casos situaciones donde debían utilizar una versión del software PSeInt diferente a la trabajada en sus prácticas habituales. Se interactuó con estudiantes y docentes de distintas escuelas, además de docentes y alumnos avanzados de la Facultad de informática. Esto último resultó alentador para aquellos estudiantes interesados en seguir una carrera

⁴ Se han desarrollado Stub de 13 lenguajes, que contienen los fuentes con las librerías necesarias para comunicarse con el servidor y las indicaciones necesarias para cada lenguaje.

informática, dado que pudieron observar que los docentes universitarios no son muy diferentes a sus profesores de la escuela media.

Con respecto a las diferencias individuales, se destaca la interacción de estos alumnos de 4 y 5 año de nivel medio, con conocimientos algunos avanzados en programación y otros recién iniciándose. Podría pensarse que el responsable de tipear la solución del grupo, los miembros recién iniciándose en programación comparados con los avanzados, podrían sufrir la presión social de sus compañeros. Considerando además la dificultad de trabajar con las distintas versiones de PSeInt y computadoras no propias. Si bien desde lo observado creemos que esto no fue así, el procesamiento de las encuestas realizadas nos permitirán acceder a las observaciones y sensaciones de los estudiantes.

Los problemas dados a los estudiantes se seleccionaron de los abarcados en la primera práctica de programación de la carrera Licenciatura en Ciencias de la Computación. Algunos de los problemas presentados involucraban conocimientos de fórmulas básicas de geometría. Dado que estos conceptos casi no se trabajan en los dos últimos años del nivel medio, surgieron como inquietud en los estudiantes, motivando a los docentes a incluirlos y trabajarlos en sus prácticas.

5 Conclusiones

Las experiencias realizadas han permitido gestionar el desarrollo y ejecución de ejercicios de programación dentro dos contextos: el curricular y otro de aprendizaje no formal (precalentamientos, competencias de programación).

Se ha podido observar en los encuentros presenciales que el nivel de satisfacción de los estudiantes no dependía exclusivamente del nivel de conocimientos de informática, sino de la interacción y participación lograda dentro del grupo integrado. En cada encuentro se armaba una nueva conformación de equipos, estando la primera asignación de grupos a cargo de los coordinadores del proyecto. En el segundo encuentro, cada grupo eligió un nombre y logo representativo, lo que creemos generó un mayor sentido de pertenencia.

En los torneos presenciales, además de ofrecer una respuesta rápida, se trataba que esta fuera detallada inclusive cuando la solución era correcta, tratando de colaborar en la mejora de la calidad de los programas. Por ejemplo indicando y mostrando un uso correcto de estructuras, nombres de variables, comentarios, escritura del código entre otros aspectos. También se ofrecieron y mostraron soluciones alternativas a las resoluciones dadas por cada grupo de estudiantes. Lo que fue valorado de forma positiva.

La aplicación en desarrollo proporciona una base de datos de problemas con diferentes niveles de dificultad, en la cual se ha tenido en cuenta la redacción y el suministro de ejemplos para evitar que surjan posibles problemas de comprensión de enunciados. Se ha prestado cuidadosa atención en la elección de los problemas a incluir, dado que si éstos resultan demasiado difíciles o fáciles podrían contribuir a disminuir la motivación de los alumnos.

Si bien el proyecto no ha concluido, creemos que los resultados y la experiencia obtenida hasta el momento ha sido valorada en forma positiva por todos los

participantes e integrantes del proyecto, revelando el potencial de los torneos de programación como recurso estratégico para la enseñanza de programación.

Referencias

1. Elena Verdú, Luisa M. Regueras, María J. Verdú, José P. Leal, Juan P. de Castro, and Ricardo Queirós. 2012. A distributed system for learning programming on-line. *Comput. Educ.* 58, 1 (January 2012), 1-10. <http://dx.doi.org/10.1016/j.compedu.2011.08.015>
2. Kris M.Y. Law, Victor C.S. Lee, Y.T. Yu, Learning motivation in e-learning facilitated computer programming courses, *Computers & Education*, Volume 55, Issue 1, August 2010, Pages 218-228, ISSN 0360-1315, (<http://www.sciencedirect.com/science/article/pii/S0360131510000102>)
3. Moreno J., Pineda A., Montoya L. "Uso de un ambiente virtual competitivo para el aprendizaje de algoritmos y programación - Experiencia en la Universidad Nacional de Colombia" Brasil. 2013. XVIII Conferencia Internacional de Informatica en Educación - TISE 2013.
4. Roy Thomas Fielding, "Architectural Styles and the Design of Network-based Software Architectures" UNIVERSITY OF CALIFORNIA, IRVINE 2000. (<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>)
5. Crescencio Bravo, Rafael Duque, Jesús Gallardo, A groupware system to support collaborative programming: Design and experiences, *Journal of Systems and Software*, Volume 86, Issue 7, July 2013, Pages 1759-1771, ISSN 0164-1212, <http://dx.doi.org/10.1016/j.jss.2012.08.039>.
(<http://www.sciencedirect.com/science/article/pii/S0164121212002439>)