

P2PSP: un protocolo de red sencillo como herramienta para el aprendizaje basado en proyectos

Cristóbal Medina-López, L. G. Casado, and Vicente González-Ruiz

University of Almería (CeiA3), Spain.
04120, La Cañada de San Urbano S/N, Almería, Spain.
{cristobalmedina,leo,vruiz}@ual.es

Resumen Este artículo describe la experiencia de aplicar la metodología de Aprendizaje Basado en Proyectos en la asignatura Redes de Computadores impartida en Ingeniería Informática de la Universidad de Almería. Concretamente, se describe el proceso docente llevado a cabo en referencia al proyecto P2PSP, un protocolo específicamente diseñado para hacer streaming de audio/vídeo a través de Internet usando el paradigma P2P, que ha permitido, por una parte, la introducción de una forma natural y motivada de los contenidos teóricos y prácticos relacionados con la materia, y por otra, poner en práctica dichos conceptos sobre una herramienta que presentaba muchos retos tecnológicos. Otro aspecto importante es que, aunque el procedimiento docente seguido supuso que algunas partes de la materia recibieran más atención que otras, el alumnado participó activamente en la definición del protocolo, lo que ayudó a motivar significativamente el seguimiento y aprendizaje de la asignatura. Gracias fundamentalmente a dicha motivación, el proyecto P2PSP.org se ha convertido en un proyecto real de software libre que en la actualidad cuenta con colaboraciones de diferentes programadores alrededor del mundo.

Keywords: P2PSP, ABP (Aprendizaje Basado en Proyectos), docencia, redes P2P (Peer-to-Peer), Redes de Computadores.

1. Introducción

En el presente trabajo se analiza la experiencia de aplicar el Aprendizaje Basado en Proyectos (ABP) llevado a cabo en las asignaturas de Redes de Computadores (RC) que pertenecen al tercer y último curso de las extinguidas Ingenierías Técnicas en Sistemas y Gestión, así como en el tercer curso de la Ingeniería Informática de la Universidad de Almería. Cuando se habla de la Informática es difícil predecir hasta donde llegará o cuales serán los nuevos retos a los que habrá que hacer frente en el futuro. Lo que parece claro es que estamos tratando con uno de los campos del conocimiento que avanzan con mayor rapidez, y las RC no son una excepción sino todo lo contrario. Sorprendentemente, esta realidad no parece aplicarse, al menos al ritmo que cabría esperar,

en la actualización de competencias que el alumno debe adquirir en el entorno universitario.

Por otra parte, cada vez son más las compañías que en sus ofertas de empleo valoran positivamente las competencias del candidato en cuanto a liderazgo, su integración en un grupo de trabajo o su creatividad, así como el conocimiento de las tecnologías más actuales. En cualquier rama de conocimiento es necesario adquirir capacidades de aprendizaje continuo [8], pero aún más cuando se trata de disciplinas como la Informática [16]. Uno de los enfoques más citados en la literatura y llevados a la práctica en diferentes ámbitos con el fin de conseguir que los estudiantes adquieran las competencias mencionadas es el ABP [3] [4] [7], que puede ser considerado una variante del Aprendizaje Basado en Problemas [17]. Una comparativa entre ambos enfoques puede encontrarse en [15].

El resto de este documento tiene la siguiente estructura. En la Sección 2 se muestran las características generales del ABP. La Sección 3 describe nuestra experiencia en la aplicación de la metodología ABP en las asignaturas de RC. La Sección 4 presenta los resultados obtenidos como consecuencia de la aplicación de esta metodología. Por último, las principales conclusiones se muestran en la Sección 5.

2. Aprendizaje basado en proyectos

El ABP (o BPL, Based-Project Learning, en inglés), es un modelo en el que el aprendizaje gira en torno al desarrollo de uno o varios proyectos para la adquisición de nuevos conocimientos, de tal modo que involucra al estudiante en el desarrollo del proyecto y el tutor actúa como un guía durante el proceso de aprendizaje.

En la literatura pueden encontrarse discrepancias en cuanto a la definición exacta del modelo [22]. Donde si hay acuerdo, es en las competencias que se pretende que adquiriera el estudiante como resultado de aplicar esta metodología. Algunas de estas competencias son:

- Capacidad de trabajo en equipo.
- Promover la toma de decisiones.
- Mejora en la habilidad para analizar y resolver problemas.
- Compromiso y responsabilidad.
- Fomento de la curiosidad y la investigación.

3. ABP aplicado al proyecto P2PSP

3.1. Definición del proyecto

La asignatura comienza con la planificación del proyecto, incluyendo la metodología y los objetivos (direccionamiento IP, puertos, protocolos TCP y UDP, NAT, HTTP, servidor de streaming, formatos de vídeo, etc.) que se deberían alcanzar tras la finalización del proyecto. Más concretamente, se estableció un

proyecto que incluía todos estos conceptos el desarrollo de un protocolo sencillo de streaming de contenidos multimedia mediante redes peer-to-peer.

A continuación se presenta un extracto de documentación que el profesor utilizó para definir el proyecto y para guiar a los alumnos en su realización:

Este proyecto, que se ha denominado P2PSP (Peer to Peer Straightforward Protocol), plantea el desarrollo de un protocolo de comunicación entre procesos de computadores para el streaming (de audio y/o vídeo) usando la arquitectura Peer to Peer (P2P). Este protocolo de comunicación se basa en dos ideas básicas:

1. A pequeña escala, todos los peers se comunicarán entre sí, todos con todos, utilizando mensajes punto a punto (unicast).
2. Un peer deberá ser tan solidario con el resto de peers como éstos lo son con él, o de lo contrario, no podrán formar parte de la red P2P. Dicha solidaridad se expresa en términos de ancho de banda, lo que significa que para que un peer forme parte de *La Red* debe de enviar a ésta en promedio tanto como recibe de ella, durante cada cierto intervalo de tiempo.

El P2PSP se definirá especialmente para realizar streaming de secuencias de vídeo en directo, es decir, puede ser utilizado para transmitir a través de una red de comunicaciones basada en la conmutación de paquetes (como lo es la red Internet) un evento que se está produciendo en ese momento, aunque también puede ser aplicado en la difusión de secuencias ya almacenadas. Las redes P2P establecen una red de comunicaciones que se puede considerar como superpuesta sobre la red real. La red superpuesta del protocolo P2PSP se muestra en la Figura 1 y estará formada por las siguientes entidades:

- **Source:** el productor del stream de datos que es retransmitido en la red P2PSP. Normalmente será un servidor de streaming que usa el protocolo HTTP.
- **Player:** solicita y consume el stream, lo decodifica y reproduce.
- **Splitter:** esta entidad recibe el stream desde el source, lo divide en trozos (chunks) del mismo tamaño y envía los trozos a los peers.
- **Peer:** recibe los chunks desde el splitter y desde otros peers, ordena los trozos y se los envía como un stream al player.
- **Team:** un splitter y uno o más peers.

Además, se definieron una serie de sub-proyectos (hitos) que sumados entre sí, resultarían el P2PSP en su forma más básica. Dichos hitos y su temporización fueron:

1. **Semana 0:** Introducción al proyecto.
2. **Semana 1:** Emisión de un stream usando una arquitectura cliente/servidor.
3. **Semana 2:** Diseño de un cliente de streaming (primera versión del splitter).
4. **Semana 3:** Transmisión desde el splitter a un peer.

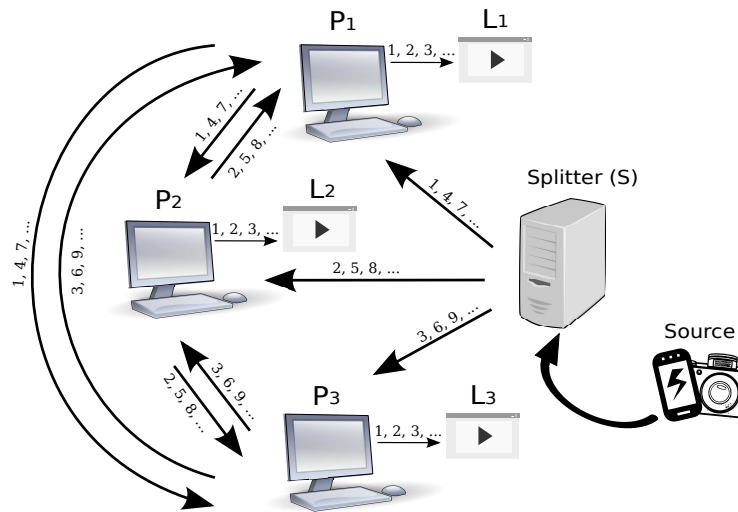


Figura 1. Un team P2PSP. Los números en las flechas indican los números de chunks (bloques de datos) retransmitidos.

5. **Semanas 4 y 5:** Transmisión desde el peer al player (cliente HTTP).
6. **Semanas 6 y 7:** La llegada de otros peers al cluster.
7. **Semana 8:** Buffering para acomodar el jitter ¹ de la red.
8. **Semana 9:** Control de los abandonos voluntarios.
9. **Semana 10:** Control de los abandonos involuntarios.
10. **Semanas 11 y 12:** Análisis del protocolo.

Cada uno de estos hitos se acompañaba de una descripción general del comportamiento esperado. Además, se proponían una serie de preguntas y experimentos que los estudiantes debían llevar a cabo.

3.2. Herramientas para facilitar el trabajo colaborativo

En la actualidad existen una gran cantidad de herramientas que facilitan el trabajo en grupo o colaborativo. En el caso específico del desarrollo de software es muy común hacer uso de repositorios de código, por ejemplo Launchpad [12] o GitHub [11], junto con sus respectivas herramientas para la gestión de versiones, Bazaar o Git. En los últimos años los repositorios han evolucionado incorporando otras características útiles para el trabajo colaborativo como son: wikis, páginas webs, gráficos y estadísticas, *reportar* bugs, asignación de tareas, hitos, e incluso algunas han llegado a darle un toque social con conceptos como seguidores o favoritos.

¹ jitter: variación de retardos entre paquetes en una misma comunicación de red.

En nuestro caso elegimos como repositorio Launchpad y como herramienta de gestión de versiones Bazaar, aunque en la actualidad se ha migrado a GitHub, ya que este cuenta con una interfaz más moderna que junto a otras características hace que sea una plataforma más atractiva para los estudiantes [5]. Más concretamente, durante el desarrollo del proyecto los estudiantes hicieron uso de las siguientes herramientas:

1. **Wiki:** era común para todos los alumnos y servía como plataforma de documentación del proyecto. En particular se hizo uso de MediaWiki [18], un servidor de software libre para crear wikis desarrollado por WikiMedia y muy conocido por ser el motor del proyecto Wikipedia [19].
2. **Repositorio y control de versiones** (Launchpad y Bazaar): únicamente era necesario crearse una cuenta o acceder mediante OpenID [9] para alojar el código en el repositorio. Después, bastaba con descargarse la utilidad `bzr` y usarlo mediante la línea de comandos.
3. **Lista de correo pública** (alojada en Launchpad): unirse a la lista era opcional, pero era interesante ya que se discutían temas relacionados con el proyecto. Los alumnos compartían y discutían sus ideas con el resto de compañeros y con el profesor.

3.3. Distribución de los grupos de trabajo

Una vez definido el problema, se solicitó que los alumnos hicieran grupos de hasta un máximo de 4 personas. Cada grupo, diez en total, tendría que trabajar en el proyecto desarrollando el protocolo en el lenguaje de programación de su preferencia. Para aquellos que no tuvieran un lenguaje preferente, se les invitó a usar Python [10], ya que es un lenguaje muy popular, de alto nivel, muy útil para la programación de Socket y además, ampliamente usado en el ámbito laboral. Uno de los requisitos derivados de seguir un protocolo común era que todos los desarrollos deberían ser compatibles entre sí, de modo que los elementos desarrollados por los diferentes grupos, como por ejemplo un peer, pudieran interactuar entre ellos. Por lo demás, los estudiantes podrían implementar el protocolo como quisieran, siempre que cumplieran las reglas básicas de funcionamiento.

3.4. Seguimiento y evaluación

Cada semana, el profesor tomaría el rol de jefe de proyecto revisando que los hitos marcados para esa semana se habían alcanzado y un representante de cada grupo debía entregar al profesor las repuestas a las preguntas planteadas en ese hito. Además, se siguió la misma filosofía que se sigue en los desarrollos de software libre, por lo que todos los repositorios de software eran accesibles por todos los grupos, estando permitido tomar ideas de los desarrollos de otros grupos, al igual que también se podían *reportar* bugs encontrados en los códigos de otros grupos para que consideraran arreglarlos.

El uso de los repositorios de software, el control de versiones, el uso de wiki, la lista de correo y las cuestiones planteadas por el profesor, convirtieron la tarea

de evaluación en un proceso objetivo. Estas herramientas permiten hacer un seguimiento de las contribuciones realizadas por los alumnos, ya que se guarda un histórico de las actualizaciones llevadas a cabo. Más concretamente, para la evaluación final se tuvieron en cuenta los siguientes puntos:

1. Aportaciones al código en el repositorio del grupo. Se tuvo en cuenta la calidad del código así como la creatividad en la resolución del problema.
2. Calidad de las soluciones a las cuestiones y experimentos planteados en cada hito.
3. Contribuciones del alumno a la wiki del proyecto.
4. Participación en la lista de correo pública del proyecto.
5. Colaboración entre grupos.

Por otro lado, se valoró positivamente las ideas que, aunque no se llegaron a implementar, aportaban valor al proyecto y podrían mejorar el protocolo una vez implementadas.

4. Resultados de la experiencia

En general, los resultados de esta experiencia han sido muy positivos. Los alumnos formaron rápidamente los grupos de trabajo y empezaron a investigar el problema y a buscar posibles soluciones. Un 26 % de los alumnos usó la lista de correo, y un 50 % participó en la documentación utilizando la wiki. Se entregaron el 100 % de los proyectos al final de la asignatura y todos cumplían los requisitos mínimos exigidos, la nota media de la asignatura se vio incrementada en 2,79 puntos sobre 10, siendo la del año del proyecto 8,4, lo que pone de manifiesto la gran motivación de los estudiantes en el desarrollo de este proyecto. De hecho:

1. Los alumnos eligieron a la asignatura RC como la mejor valorada de toda la titulación.
2. Uno de los alumnos, que posteriormente realizó un PFC relacionado con el P2PSP, recibió el premio nacional de software libre [6] y el premio al mejor proyecto fin de carrera de la titulación.

Como resultado de esta motivación, algunos estudiantes siguieron trabajando en el proyecto después de terminar el curso. Este hecho hizo que el proyecto siguiera evolucionando fuera del aula, lo que permitió que otros colaboradores externos aportasen código al mismo. Actualmente, el proyecto ha crecido de tal forma que:

1. El protocolo P2PSP es un proyecto de software libre en el que están implicados desarrolladores voluntarios de diferentes partes del mundo [20].
2. En 2014 se creó una empresa de base tecnológica que está explotando comercialmente la implantación del protocolo P2PSP [1].

3. En 2015, la organización P2PSP.org [21] ha sido seleccionada como una de las 137 organizaciones para participar en el programa Google Summer of Code [2], un programa de Google a nivel mundial que remunera a los estudiantes que completen un proyecto para una organización de software libre durante el verano. Para el programa, se han propuesto 16 ideas, algunas de las cuales, han sido aportadas por los propios estudiantes.
4. Se ha creado una línea de investigación paralela al proyecto que estudia los aspectos más innovadores del P2PSP [13] [14].
5. La metodología ABP aplicada al P2PSP ha seguido usándose en asignaturas afines con igual éxito.

5. Conclusiones

Nuestra experiencia nos ha mostrado que la metodología ABP no sólo es una herramienta útil para la enseñanza superior en las escuelas de ingeniería, sino que es tal la motivación que genera en comparación con las clásicas prácticas establecidas en los antiguos modelos de enseñanza, que consigue que los estudiantes sigan trabajando en el proyecto incluso tras la finalización de la asignatura. Por lo tanto, se puede afirmar, que la supervisión directa del trabajo colaborativo del alumnado en un proyecto produce excelentes resultados.

En este caso concreto, se ha podido comprobar cómo un proyecto que nació con un enfoque puramente docente, como es el P2PSP, se ha convertido en un proyecto de software libre real en el que actualmente colaboran personas de diferentes partes del mundo, en su mayoría estudiantes.

Referencias

1. Luxunda, S.L. <http://www.luxunda.es/>.
2. GSoC 2015. P2PSP en Google Summer of Code 2015. <https://www.google-melange.com/gsoc/org2/google/gsoc2015/p2psp>.
3. Hemachandra Bhat, J Sujatha, KSR Mohan, and Geetha Prakash. Project based learning: An industry perspective. *Journal of Engineering Education Transformations*, pages 302–305, 2015.
4. José Manuel Cortés Martín. El aprendizaje basado en problemas como técnica didáctica del derecho internacional público. *DOCENCIA Y DERECHO*, (2), 2010.
5. Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. Social coding in github: Transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, CSCW '12*, pages 1277–1286, New York, NY, USA, 2012. ACM.
6. Comité de evaluación de proyectos del CUSL. Premiados en la 8ª Edición del Concurso Universitario de Software Libre. <http://www.concursosoftwarelibre.org/1314/node/33>.
7. Ignacio de los Ríos, Adolfo Cazorla, José M. Díaz-Puente, and José L. Yagüe. Project-based learning in engineering higher education: two decades of teaching competences in real environments. *Procedia - Social and Behavioral Sciences*, 2(2):1368 – 1378, 2010. Innovation and Creativity in Education.

8. Agustín Fernández, Josep Llosa, and Fermín Sánchez. Estrategias para el diseño de laboratorios orientados al aprendizaje continuo. *JENUI*, 8:189, 2008.
9. The OpenID Foundation. Openid. <http://openid.net/>.
10. The Python Software Foundation. Python. <https://www.python.org/>.
11. GitHub Inc. GitHub. <https://github.com/>.
12. Canonical Ltd. Launchpad. <https://launchpad.net/>.
13. Cristóbal Medina-López, Juan Pablo García Ortiz, J.A.M Naranjo, L.G. Casado, and Vicente González-Ruiz. IPTV using P2PSP and HTML5+WebRTC. In *The Fourth W3C Web and TV Workshop (Web & TV Convergence)*, page Paper Submission 5, Munchen, Germany, March 2014. IRT (Institut für Rundfunktechnik), Munchen, Germany, The World Wide Web Consortium (W3C).
14. Cristobal Medina-López, J.A.M. Naranjo, Juan Pablo García-Ortiz, L. G. Casado, and Vicente González-Ruiz. Execution of the P2PSP protocol in parallel environments. In *Actas XXIV Jornadas de Paralelismo*, pages 216–221, Madrid, Septiembre 2013.
15. José A Domínguez Navarro, Eva-Sara Carod Pérez, and María Jesús Velilla Marco. Comparativa entre el aprendizaje basado en proyectos y el aprendizaje basado en problemas. In *II Jornadas de innovación docente, tecnologías de la información y de la comunicación e investigación educativa*, Zaragoza, 2008.
16. Javier Oliver. El futuro de la formación de los profesionales informáticos. *IX Jornadas de Enseñanza Universitaria de la Informática*, 2003.
17. Henk G Schmidt. Problem-based learning: An introduction. *Instructional science*, 22(4):247–250, 1994.
18. The MediaWiki Developers/Maintainers Team. Mediawiki.org. <https://www.mediawiki.org/wiki/MediaWiki>.
19. The MediaWiki Developers/Maintainers Team. Wikipedia. <https://www.wikipedia.org/>.
20. The P2PSP Team. Peer to Peer Straightforward Protocol. <http://p2psp.org/en/p2psp-protocol>.
21. The P2PSP.org Team. The p2psp.org project. <http://www.p2psp.org/>.
22. John W Thomas. A review of research on project-based learning. 2000.