

Experiencias en Taller de Proyecto II

Fernando G. Tinetti¹, Gastón A. Marón, Alan F. Castelli, Julián Delekta

Facultad de Informática, Universidad Nacional de La Plata, La Plata, Argentina

¹Comisión de Investigaciones Científicas de la Provincia de Bs. As., La Plata, Argentina
fernando@info.unlp.edu.ar

Resumen. En este trabajo se presentan los lineamientos generales y experiencias recogidas en una de las asignaturas del último semestre de la Ingeniería en Computación de la Universidad Nacional de la Plata: Taller de Proyecto II. En esta asignatura se integran muchos de los conocimientos adquiridos en las asignaturas anteriores y los estudiantes tienen contacto directo con un proyecto completo “de punta a punta” en un ambiente que se intenta asemejar más al de producción en el cual se desarrollarán una vez graduados. A lo largo del desarrollo de la asignatura se tiene contacto directo con partes de hardware, sistemas/protocolos de comunicaciones y desarrollo de software (no solo referido a la programación necesaria para el funcionamiento de hardware y comunicaciones, sino también a otros aspectos como problemas de tiempo real y de interfaz de usuario). Desde el punto de vista de las actividades de los estudiantes, se plantea un ambiente similar al de un usuario o empresa que determina los requerimientos del proyecto. Esto sin dejar de lado la tarea docente o el ambiente propio de una asignatura, en la que los estudiantes tienen acceso a explicaciones y adquisición de conocimientos y habilidades en función de las tareas que deben resolver.

Keywords: Desarrollo de Proyectos, Sistemas de Tiempo Real, Sistemas Web.

1 Introducción

La asignatura Taller de Proyecto II (a la que se hará referencia a partir de este punto con TdPII) se dicta en el último semestre de la Ingeniería en Computación [4] de la Universidad Nacional de La Plata. El objetivo en general es que los estudiantes trabajen sobre un proyecto integrador con una aplicación concreta de los conocimientos adquiridos, tanto en términos de hardware como de software. Esto permitiría complementar la formación en trabajos/aplicaciones que incluyen hardware y software, incluyendo nuevos temas o problemas que engloban varios temas relacionados. Se tiene al menos una reunión (usualmente informal) anual para articulación con la asignatura Taller de Proyecto I.

TdPII incorpora y combina de manera integrada diferentes contenidos de asignaturas previas de la carrera, aunque en este caso aplicados a contextos específicos de: a) Arquitectura de sistemas de cómputo (con microprocesadores y/o microcontroladores), con sus características y limitaciones [2] [3], b) Redes y sistemas distribuidos [6] [10], normalmente desde el punto de vista de una interfaz web, no siempre provista por un

sistema completo de cómputo como una computadora de escritorio o servidor web “estándar”, c) Desarrollo/ingeniería de software. Es importante destacar que no solo es importante la funcionalidad, sino otros aspectos relativamente heterogéneos como los relacionados con cómputo de tiempo real [1] [5], documentación e interfaces de usuario [8] [7]. Con estos lineamientos, a lo largo del tiempo de la cursada cada estudiante está involucrado en el desarrollo de un proyecto específico.

Cada proyecto se debe llevar a cabo por un grupo reducido de estudiantes (usualmente dos o tres estudiantes por grupo), donde no solamente se reafirman los temas propios de cada proyecto en particular, sino que permite tener una identificación más sistemática y ágil en el contexto del trabajo en grupos. El desarrollo en grupo tiene en cuenta el trabajo de/en producción que es mayoritario en el trabajo profesional futuro. El planteo de una relación que a la vez de ser docente-estudiante se asimila a una relación de usuario o empresa que define los requerimientos, se pretende como aporte en cuanto a la justificación técnica de las decisiones tomadas e implementadas en el contexto del proyecto específico que se desarrolla.

La cantidad reducida de estudiantes de cada grupo está orientada a que todos los integrantes del grupo tengan contacto directo con todas las partes del proyecto. Con este objetivo, el alcance/profundidad de cada proyecto se dimensiona de manera tal que se pueda resolver en el tiempo que impone el calendario académico para la asignatura, que es de un semestre. En cuanto a la dimensión/complejidad del proyecto y la restricción de tiempo impuesta por el calendario académico, por un lado, se aprovecha la experiencia de la cátedra y por el otro, también se acuerda con cada grupo una funcionalidad y características mínimas para el proyecto, a partir de la cual el grupo puede definir su continuación siempre que el tiempo lo permita.

2 Definiciones Generales y Desarrollo Cronológico

Tanto para homogeneizar el tipo de trabajo a realizar como para dar un contexto común a todos los grupos de estudiantes, la estructura general de cada proyecto se define esquemáticamente como lo muestra la Fig. 1. A través del ejemplo de la Fig. 1 a) se muestran por un lado los flujos de información (Fig. 1-a) y por el otro los contenidos generales que todos los proyectos tienen: hardware, comunicaciones e interfaz/acceso web (Fig. 1-b). Para cada una de las partes, se consolidan conceptos de hardware, comunicaciones y redes y al menos en parte sistemas distribuidos y sistemas web.

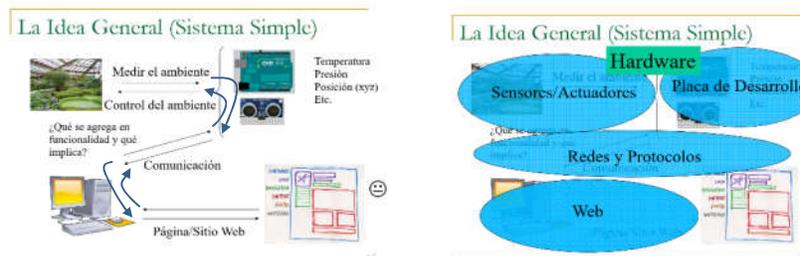


Fig. 1. Estructura General de los Proyectos.

En el tiempo de cursada se suceden cuatro etapas, donde cada una de ellas concluye con al menos un documento entregado por cada grupo de estudiantes: 1) Definición del Plan de Proyecto, 2) Entrega/Informe de Avance I que incluye código fuente, si ya han desarrollado, 3) Entrega/Informe de Avance II que incluye código fuente, no necesariamente completo y 4) Entrega Final, proyecto completo. La sucesión y alcance/contenido de estas cuatro etapas se ha definido en función de la experiencia de la cátedra y se ha mantenido al menos en los últimos años. Como mínimo, la entrega final del proyecto implica una presentación en clase por parte de los estudiantes. En todos los casos, se evalúan las entregas en general y los informes/documentación asociada en particular.

3 Resultados

A partir de un catálogo de proyectos inicial, los estudiantes definen grupos y seleccionan según sus preferencias. Para los casos (no muy frecuentes) en que más de un grupo selecciona un mismo proyecto del catálogo, se definen variantes para cada grupo, que son suficientemente heterogéneas como para que el solapamiento de dos proyectos a realizar sea mínimo. Esto por un lado genera una gran variedad de proyectos y, por otro lado, dado que las exposiciones se realizan en clase para todos los estudiantes, el intercambio de ideas y posibilidades alcanza a todos los grupos. La mayor cantidad de inconvenientes que se han presentado en casi todas las cursadas se pueden resumir en: 1) Generación de documentación técnica, y 2) Cumplimiento de plazos de entrega. En la mayoría de los casos, los proyectos no solo cumplen los requisitos mínimos, sino que a medida que se avanza en el desarrollo, cada grupo suele proponer y acordar con los docentes sus propias ideas a implementar.

Dado que la cátedra define de antemano el catálogo de proyectos, en general también los materiales han sido comprados y provistos por la cátedra de antemano. En algunos casos, los grupos han gestionado compras, cuyo costo ha sido también provisto por la cátedra. La gestión de compras de materiales por un lado incorpora algunos aspectos de logística en los grupos y por el otro también permite a la cátedra mantener los proyectos, de manera tal que se puedan extender en cursadas posteriores. La decisión de proveer materiales que los propios grupos mantienen en su poder a lo largo del tiempo de cursada genera riesgos de pérdida o destrucción, pero agrega responsabilidad a los grupos. Más del 95% de los materiales se reutiliza.

La evolución de los proyectos a lo largo de dos o tres años de cursada tiene varias ventajas: 1) asimila de alguna manera el trabajo en producción, donde en la mayoría de los casos los profesionales se incluyen en proyectos ya desarrollados o cuyo desarrollo ya se ha iniciado, 2) se puede acentuar la importancia de la generación de documentación técnica precisa, dado que cada grupo eventualmente recibe un proyecto documentado o genera uno que va a recibir un grupo siguiente, y 3) se pueden agregar funcionalidades o mejorar aspectos de diseño de los proyectos, para los que no habría tiempo en un único semestre. Como un ejemplo de evolución puede mencionarse un vehículo móvil controlado por interfaz web: inicialmente solo se implementó el sistema completo con comando básicos desde la interfaz, en un año siguiente se incorporó la corrección de errores de movilidad de motores para el movimiento en línea recta y control de

colisiones, incorporando sensores de distancia. En varios casos a partir de una implementación inicial de un proyecto se cambió el hardware, justificando y documentando los cambios. Uno de los ejemplos que se ha repetido es el de cambiar la combinación microcontrolador + módulo de comunicaciones WiFi (Arduino+ESP8266-01) por placas de desarrollo que contienen ambas funcionalidades integradas (NodeMCU con ESP8266 o placas de desarrollo con ESP32). Estos cambios usualmente involucran también agregados funcionales y decisiones más cercanas al propio hardware como la distribución de alimentación. Algunos ejemplos y documentación asociada que se ha generado se pueden encontrar en [9].

4 Conclusiones

La asignatura TdPII introduce algunos aspectos que los propios estudiantes identifican como integradores de conceptos a partir de proyectos completos, de “punta a punta”. La evolución en el tiempo de los proyectos ha generado que los estudiantes se involucren en proyectos ya desarrollados que se modifican de acuerdo a nuevos requerimientos. También, integrarse en un proyecto ya desarrollado permite a los estudiantes identificar la importancia de la documentación recibida y generada. La decisión de responsabilizar a los estudiantes por los materiales fuera de un laboratorio o aula ha demostrado una muy buena respuesta a identificación de problemas nuevos o diferentes, de logística, a resolver. El análisis de documentación recibida por los estudiantes y la responsabilidad de generar mejor y más precisa documentación pone de relieve que los proyectos no solo tienen que satisfacer requisitos funcionales, sino que deben identificar y justificar las decisiones implementadas.

Referencias

1. G. Buttazzo, *Hard Real-Time Computing Systems*, Springer, 2011.
2. F. M. Cady, *Microcontrollers and Microcomputers Principles of Software and Hardware Engineering*, Oxford University Press, 2009.
3. J. L. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2017.
4. Ingeniería en Computación - UNLP, “IC Página Principal”, <https://ic.info.unlp.edu.ar/>
5. H. Kopetz, W. Steiner, *Real-Time Systems: Design Principles for Distributed Embedded Applications 3rd Ed.*, Springer, 2022.
6. J. Kurose, K. Ross *Computer Networking*, 8th Ed., Pearson, 2020.
7. R. Pressman, *Ingeniería de Software. Un enfoque práctico*, 9na Ed., McGraw-Hill, 2021.
8. I. Sommerville, *Ingeniería de Software*. 10ma Edición, Pearson, 2016.
9. F. G. Tinetti, “ftinetti - Real-Time & Embedded Systems”, <http://fernando.scienceontheweb.net/links/embed-rt/>
10. M. van Steen, A.S. Tanenbaum, *Distributed Systems*, 3rd Ed., CreateSpace Independent Publishing Platform, 2017.