

Cómo motivar al alumnado entrelazando las asignaturas Programación Avanzada y Tecnología de Computadores

Pedro Pablo Garrido Abenza
Héctor Francisco Migallón Gomis
Dept. Física y Arquitectura de Computadores
Universidad Miguel Hernández
03202 Elche (Alicante)
e-mail: pgarrido,hmigallon@umh.es

Resumen

En este artículo se presenta una experiencia realizada en las titulaciones técnicas de telecomunicaciones de la Universidad Miguel Hernández de Elche, con el fin de motivar al alumnado entrelazando dos asignaturas de su plan de estudios. Actualmente esta experiencia también se quiere implantar en la titulación de Ingeniero Técnico en Informática de Gestión (ITIG) de la misma universidad. Concretamente esta experiencia se basa en que mientras un alumno de ITIG cursa la asignatura de primer curso Tecnología de Computadores (TC), utiliza para el desarrollo de las prácticas de esta asignatura el software desarrollado por un compañero suyo en la asignatura optativa de tercer curso Programación Avanzada (PA), quien tiene la ventaja de haber sido en primer lugar un usuario y de alguna manera el que ha realizado el test de funcionamiento de un software diferente pero con las mismas funciones.

1. Introducción

A la mayoría de estudiantes de ITIG o de las distintas ingenierías técnicas en telecomunicaciones, el temario de la asignatura TC, o de las diferentes asignaturas de similar temario, les parece que está apartado del conjunto de conocimientos que tienen que adquirir en el transcurso de sus estudios superiores. Esto supone una traba para el profesorado que tiene que impartir una materia que no es del agrado del estudiante, y para el estudiante porque tiene la sensación de que no le va a servir de nada lo que

está aprendiendo. Obviamente no existe duda alguna de que esto es erróneo.

Por otro lado el estudiante de PA se encuentra con una asignatura optativa de último curso de ITIG de la cual tiene buenas expectativas, ya que las asignaturas optativas son las que le van a dotar de conocimientos específicos y de las que espera contenidos que puedan aplicar directamente en su cercana vida profesional, por tanto es complicado cubrir esas expectativas.

Pensamos que un modo correcto de satisfacer esas expectativas es realizar una aplicación informática de simulación desde el principio hasta el final. Esto, además, es un buen incentivo para el alumno por ser la primera aproximación a un primer trabajo de desarrollo completo que puede utilizar como experiencia previa al Proyecto Fin de Carrera, sin confundir la diferencia de envergadura de ambos. Hay que destacar que, por lo general, el estudiante se encuentra con que sus trabajos prácticos normalmente son tareas concretas, independientes unas de otras y con una extensión en el tiempo que no suele superar las cuatro sesiones de dos horas. En nuestro caso proponemos un único trabajo que extenderemos durante todas las prácticas de la asignatura, consistente en el desarrollo de un simulador de circuitos digitales completo, el cual se realiza en fases sucesivas de forma continua, comenzando por el análisis y el diseño del mismo, y continuando con su implementación en el lenguaje de programación Java. Esto hace que también vayan perdiendo dependencia del grupo en conjunto y que se trabaje de verdad en un grupo reducido de personas, lo cual está más ajustado a la realidad que se van a encontrar una vez finalicen sus estudios.

2. Descripción de la asignatura “Tecnología de Computadores”

La asignatura TC [1], [2] basa su contenido en el estudio de:

- Sistemas de representación numéricos.
- Funciones lógicas.
- Circuitos combinacionales.
- Circuitos secuenciales.

Tanto los sistemas de representación numérica como las funciones lógicas no suponen ningún problema ni en su docencia ni en la predisposición del alumnado, que considera que esta parte sí es necesaria como base para las asignaturas de programación. De hecho las prácticas de la asignatura sí usan el contenido de esta primera parte de la asignatura pero no es necesario el refuerzo de dicha parte con sesiones prácticas específicas. Sin embargo, tanto la parte correspondiente a circuitos combinacionales, como la de circuitos secuenciales, es más difícil atraer la atención y la curiosidad del alumnado. Esto se debe a diversos motivos. El fundamental es que el estudiante considera que no le aporta nada, y en segundo lugar, que es poco tangible para él. Todo esto, unido a lo expresado anteriormente, nos lleva a utilizar todas las sesiones prácticas para desarrollar esta parte del temario de la asignatura.

3. Descripción de la asignatura “Programación Avanzada”

La asignatura PA basa su contenido en el estudio de un lenguaje de programación avanzado como es JAVA [3]. Si nos centramos en las características de la titulación de ITIG, antes de llegar a cursar esta asignatura se han adquirido los conocimientos básicos de programación en la asignatura Fundamentos de Programación (FP), donde han trabajado con el lenguaje C, viendo prácticamente la totalidad de sus características y prestaciones. La asignatura FP se imparte en primer curso. Posteriormente, en la asignatura de Estructura de Datos (ED) de segundo curso, se refuerzan los conocimientos de C y se entra a estudiar nociones de Programación Orientada a Objetos (POO) trabajando con C++. El estudio de la POO [4] es muy importante para el desarrollo

de la asignatura de PA, ya que permite dedicarnos de pleno a las características avanzadas del lenguaje JAVA. Así pues, el temario de la asignatura lo distribuiremos en los siguientes temas o unidades didácticas:

- Conceptos básicos del lenguaje JAVA.
- Programación Orientada a Objetos.
- Conceptos avanzados de programación.
- Aplicaciones gráficas.
- Flujos de datos.
- Comunicaciones e Internet.

En el primer punto se presenta los elementos básicos del lenguaje Java, tales como las palabras clave y estructuras de control, muy similares a las del lenguaje C. En el segundo punto se hace un repaso de la POO de forma genérica, es decir, sin particularizar para ningún lenguaje concreto. Puesto que, como ya hemos adelantado, el alumnado ya ha adquirido estos conocimientos, se puede basar el contenido de la asignatura en los conceptos avanzados del lenguaje, que son los necesarios para la realización completa del software de simulación.

4. La “no” relación entre ambas asignaturas

Aunque es evidente que ambas asignaturas no tienen ningún tipo de relación que las una en el temario, es posible establecer la comunicación entre ambas asignaturas.

El hecho de que existe una tendencia a la desaparición de las incompatibilidades entre asignaturas, debemos tener en cuenta este factor a la hora de relacionar asignaturas. Podemos generar un problema a determinados estudiantes que por motivos cualesquiera lleguen a PA sin haber superado TC. Por tanto ha de comunicarse a los alumnos, en el programa de la asignatura que es aconsejable haber superado la asignatura TC. No obstante, lejos de ser un obstáculo, esta relación podría producir el efecto contrario, esto es, el desarrollo de las prácticas de PA podría ayudar al alumno en el aprendizaje de TC.

5. Justificación de la experiencia

La finalidad de esta experiencia es la de que el alumnado encuentre motivaciones para estudiar y trabajar con interés en ambas asignaturas.

Se obtiene un doble beneficio. El primero de ellos con sobre los alumnos de TC. La experiencia que tenemos hasta el momento es que el estudiante de TC acoge con mayor agrado el software de simulación creado por sus compañeros -aunque sólo sea por ver de lo que serán capaces de hacer y mejorar- que cualquier otro software comercial o de libre distribución estándar, los cuales suelen incluir muchas más funciones de las necesarias, dificultando así las fases iniciales del aprendizaje. El segundo de los beneficios se obtiene con los alumnos de PA, que agradece el desarrollo de un trabajo con ya determinada entidad, factor este muy importante en el desarrollo profesional del alumno.

Una característica importante a reseñar es que intentar motivar al alumnado en el desarrollo de TC se debe, en gran parte, a la inexistencia de laboratorios de electrónica en las titulaciones de informática, o la gran tasa de uso de éstos en las titulaciones de telecomunicaciones. Esta carencia, aunque es lógica, provoca que la asignatura TC tenga la problemática de que el alumnado no experimenta realmente con los circuitos integrados, y nunca llega a saber realmente como se trabajaría con ellos. El disponer de estos laboratorios sería un gran aporte para la asignatura TC, pero debe comprenderse que ésta sería una de las últimas prioridades para una universidad o escuela politécnica.

6. Motivación para el alumnado

Como hemos dicho, la finalidad de relacionar ambas asignaturas es motivar al alumnado. Esta motivación debemos llevarla a cabo en ambas asignaturas. Es decir, debemos intentar que, aunque una asignatura se apoye en la otra, exista motivación en ambas.

Dicha motivación la llevamos a cabo para la asignatura TC por medio de las siguientes premisas:

- Cada estudiante utiliza una herramienta creada por un compañero suyo, por tanto, ya no se parte de la idea de que es un software

perfecto, pudiendo encontrar, por tanto, posibles fallos. Esto le lleva a experimentar, analizar y comprobar la herramienta.

- El conocer de primera mano, como usuario final de la aplicación, cuáles han sido los fallos o características no adecuadas de la herramienta, para más adelante utilizar los resultados de esta experimentación en el desarrollo de su propio simulador de circuitos digitales.

Y para la asignatura PA por medio de las siguientes premisas:

- El saber que la herramienta que está realizando puede ser utilizada por un compañero suyo hace que intente desarrollar un software robusto para ponerle difícil a su compañero el encontrar los posibles fallos.
- Será la primera vez que se comience una aplicación y se finalice por completo, de modo que esta experiencia puede ser utilizada como medio de aprendizaje para la realización de proyectos o para iniciarse en todas las fases de las que debe constar un proyecto.

Por último, hay una motivación espontánea en los alumnos de ambas asignaturas, relacionada con un objetivo psicomotor establecido en su programa. Este objetivo, no es más que el de dar buen ejemplo por parte de los profesores, haciéndoles ver a los alumnos la capacidad que pueden llegar a alcanzar, así como hacerles ver las ventajas de desarrollar herramientas software propias, que aunque no sean tan perfectas y completas como otras herramientas comerciales estándar, aquellas están más personalizadas y se pueden mejorar constantemente.

7. Prácticas a realizar en TC

En la asignatura TC las prácticas consistirán en la experimentación con puertas lógicas y en el diseño, realización y comprobación de diferentes circuitos secuenciales y combinacionales. La realización de estos circuitos y la comprobación de los mismos, por las características de las titulaciones de informática y sus dotaciones de laboratorios (aulas de informática), han de desarrollarse necesariamente bajo un software de simulación. Con este software de simulación los

alumnos manejarán todo tipo de puertas lógicas, realizarán sus circuitos y comprobarán el funcionamiento de los mismos.

En particular el simulador estará preparado para trabajar con:

- Puertas AND.
- Puertas OR.
- Puertas NAND.
- Puertas NOR.
- Puertas XOR.
- Puerta NOT.
- Entradas Digitales.
- Salidas Digitales.

Como el simulador con el que se trabaja no será ninguna versión que haya pasado un filtro de test la primera y segunda sesión de prácticas (4 horas) consistirán en poner a prueba dicho simulador, es decir, en hacerle un test de funcionamiento. Con esto conseguimos que se consoliden los conocimientos que se están adquiriendo en la asignatura, ya que para poner en duda el funcionamiento del simulador, se debe dudar realmente de los resultados mostrados por éste. Se orientará cómo comenzar la fase de test, empezando por las funciones lógicas básicas con puertas lógicas de dos entradas, construyendo puertas básicas de más de dos entradas a partir de varias puertas básicas (comprobando su equivalencia), construyendo puertas no incluidas en el simulador... Esto se realiza en la primera sesión de prácticas. Es necesario hacer constar que, aun siendo software diferente, sí existe una uniformidad en las tareas básicas, lo cual facilita la explicación común del software al conjunto del alumnado.

La segunda sesión se dedicará, a modo de reto o de juego, consistente, a la búsqueda del fallo. Debe ser cada estudiante el que se invente diferentes circuitos, el que los monte en el simulador, el que calcule cómo deben funcionar y el que compruebe si dicho funcionamiento real coincide con lo que había calculado.

Esta sesión debe acabar con la emisión, por parte de cada estudiante, de un informe con los fallos detectados, si es que los hubiere, de manera que para la siguiente sesión de prácticas se sepa si el simulador con el que debe trabajar es adecuado, o es más recomendable la utilización de la versión desarrollada por nosotros denominada *Simulin*. Dicha versión no sólo ha sido testada por nosotros, sino que también lo ha sido por los

propios alumnos de TC, como cualquier otro simulador desarrollado por alumnos de PA. Además, dispone otras funciones que vamos añadiendo continuamente, con el objetivo de cubrir la totalidad del temario de TC. Por ejemplo, podemos mencionar módulos para realizar conversiones entre sistemas de numeración, simplificación de funciones mediante mapas de Karnaugh (K-Maps), etc. Se les ofrece esta versión para evitar que la carencia de confianza en el software perjudique el desarrollo de sus prácticas.

Como se ha explicado, el inicio de trabajo con esta filosofía de reutilizar el software de otros compañeros, es completamente diferente al inicio con un software estándar, de modo que se incentiva el inicio de trabajo. Se constata que en las sesiones iniciales de prácticas la asistencia de alumnado es alta, y que es en este momento en el que hay que intentar motivarlos de tal modo que no desesperen, y para que continúen en el desarrollo completo de las prácticas. Con nuestra experiencia conseguimos disminuir el porcentaje de alumnado, que deja la asignatura para futuras convocatorias ya en el inicio de la asignatura.

Según nuestra experiencia, los problemas encontrados por los alumnos se deben más a la utilización de los diferentes eventos de ratón y teclado, que a problemas de funcionamiento correcto del simulador.

Se tiene en cuenta que, el software desarrollado por los alumnos de PA que se distribuye al alumnado de TC para realizar sus prácticas, ha sido evaluado por el profesor o profesores responsables. En esa evaluación se debe de marcar el software que sí es válido para la realización de las prácticas de TC, descartando, así mismo, el software que no sea válido. Este último caso no implica el suspenso en la asignatura de PA, sino que por diferentes motivos, ya sea por no ser lo suficientemente robusto, dificultad en el uso, documentación de ayuda en línea pobre,... no se considera adecuado para el uso de sus compañeros. Aun no significando un suspenso si debe verse penalizado en la nota.

8. Práctica a realizar en PA

En la asignatura PA dedicaremos todo el tiempo asignado a laboratorio a realizar un entorno de simulación de circuitos digitales, incluyendo

puertas lógicas, circuitos secuenciales y combinatoriales, dotado de las características básicas de las que consta cualquier simulador de este tipo. Debe de cumplir, por tanto, con las características mencionadas para el simulador utilizado en el desarrollo de las prácticas de TC y por supuesto debe cumplir con unas reglas fijadas, puesto que, como ya hemos dicho, este software de simulación será utilizado por sus compañeros de TC.

La práctica realmente se desarrolla en dos fases claramente diferenciadas. En la primera de ellas se realizará un simulador modo texto, y tendrá las siguientes características:

- Permite todos los tipos de puertas antes mencionados.
- Todas las puertas lógicas, excepto la NOT, pueden tener dos ó más entradas.
- Los circuitos se construyen de forma programada, esto es, para implementar un circuito distinto tendremos que modificar el código fuente y recompilar.
- Obtención de la función lógica equivalente en notación prefija, como mecanismo de comprobación del circuito.
- Obtención de la tabla de verdad para la comprobación de resultados.

En la segunda fase, se dotará al simulador de una interfaz gráfica, la cual ya será la versión utilizable por sus compañeros en la asignatura TC. En esta versión se debe añadir las herramientas para visualizar las puertas, las conexiones y los módulos de entrada y salida a modo de circuito digital convencional, así como la creación de circuitos mediante el ratón y presentación gráfica de resultados (tabla de verdad).

La distribución de las prácticas en estas dos fases es importante, ya que se debe empezar desde la primera sesión con la realización del simulador, y, sin embargo, no será hasta avanzado la mitad del cuatrimestre cuando se adquieran los conocimientos básicos para el desarrollo del entorno gráfico. Por tanto, la base de la programación de las clases necesarias se debe ir desarrollando de manera que pueda ir probándose su corrección de algún modo, como por ejemplo, obteniendo la función lógica equivalente o la tabla de verdad del circuito. Se desarrollará los diferentes objetos, de los que debe constar el simulador, a nivel de comportamiento lógico y cómo relacionarlos con el resto de objetos, así

como los módulos de obtención de resultados. Esta parte se puede desarrollar completamente prescindiendo del entorno gráfico y con la ayuda para la depuración de un lenguaje texto de simulación. Finalizada esta parte de desarrollo ya se dispone de los conocimientos necesarios para dotar a los objetos de su característica gráfica, y la construcción de aplicaciones gráficas.

Como acabamos de decir, la primera parte de la práctica implementa todas las funciones, y es manejable en modo texto. Tenemos que reseñar que sería aconsejable que el alumnado dedujera por sí mismo los esquemas de realización, ya que es una de las fases importantes en la realización de un proyecto, pero como esto supone un problema, se les ofrece una guía de cómo ir realizando el trabajo. El modo de utilizar las herramientas gráficas ya es una parte más creativa, en la cual puede aportar más ideas y contribuciones propias, fase en la cual no se ofrece guía de realización. Por último, la documentación de utilización del software está incluida en las tareas a realizar, y debe estar incluido en el software de simulación, de forma que el usuario posterior tendrá toda la documentación necesaria en el paquete de distribución.

En la evaluación de la práctica de PA debe de tenerse presente el hecho de que el software ha de poder reutilizarse, y es aquí donde se evitan los posibles problemas graves de reutilización del código. El código que cumple unas condiciones básicas es preparado para la realización de las prácticas de TC del siguiente curso. Puesto que la asignatura PA es de segundo curso, y TC es de primero, dispondremos de un mismo software de simulación para aproximadamente dos grupos (cuatro personas). Una opción que se desechó por problemas en el mantenimiento de las aulas de informática, era instalar una versión diferente en cada ordenador del aula.

9. Posibles mejoras

Una posible modificación que nos gustaría realizar en este planteamiento, es poder dejar a aquellos estudiantes aventajados que tienen las ideas muy claras, y un concepto diferente de cómo debe ser el simulador, libertad para la confección del mismo. Ello sólo sería posible, y desde nuestro punto de vista recomendable, si el tiempo disponible permitiera la confección de una ayuda

completa al usuario del software. De esta forma podríamos, al inicio de las clases prácticas de TC, impartir una breve introducción del programa, es decir explicar para qué sirve y qué es lo que queremos conseguir de él. Con esto y la ayuda en línea incorporada sería suficiente para que los alumnos de TC ya pudieran manejarse con el software de simulación.

Si es necesario la realización de una ayuda completa por el modo de enfocar las prácticas de TC, lo podemos conseguir dándoles comenzado el proyecto con determinados objetos ya parcialmente definidos. Así esta ganancia de tiempo la dedicarían a la implementación de una buena ayuda de usuario. Esta opción, que hemos desechado, lleva al alumnado a introducir el código entregado dentro de su software sin preocuparse de entenderlo, por lo que se decidió que desarrollaran el software completo.

10. Conclusión

De momento la experiencia nos dice que se ha conseguido la intención con la que comenzamos la planificación de este experimento, que no era más que motivar a nuestros estudiantes.

Además mostramos que es posible el trabajo común entre dos asignaturas no relacionadas en el

temario, consiguiendo beneficios por ambas partes.

Comprobamos también, cómo en la asignatura TC partiendo de un inicio algo lúdico en las sesiones prácticas, el alumnado de alguna manera gana seguridad en sus conocimientos, y comprende que el poner en duda lo que se le transmite es una forma de aprender.

Por último en la asignatura PA hemos notado cómo incentivo el tener un desarrollo de cierta entidad, que el estudiante sabe que puede llevar a buen término con la ayuda del profesor, aunque dejándole bastante libertad en su ejecución.

Referencias

- [1] Floyd, T.L. *Fundamentos de Sistemas Digitales*. Prentice Hall. 2000.
- [2] Hayes, John P. *Introducción al diseño lógico digital*. Addison-Wesley Iberoamericana. 1996.
- [3] Joyanes Aguilar, L., Zahonero Martínez, I. *Programación en Java 2*. McGraw-Hill. 2002.
- [4] Muñoz Caro, C., Niño Ramos A., Vizcaíno Barceló A. *Introducción a la programación con orientación a objetos*. Pearson Educación. 2002.