

TRANSICIÓN DE MODULA-2 A JAVA EN UN CURSO DE INICIACIÓN A LA PROGRAMACIÓN

Xavier Franch¹, Joaquim Gabarró¹

¹L.S.I.-U.P.C. (Departament Llenguatges i Sistemes Informàtics,
Universitat Politècnica de Catalunya)
e-mail: franch|gabarro@lsi.upc.es

Resumen: Se presenta una experiencia piloto desarrollada en la asignatura de iniciación a la programación impartida en la Facultat d'Informàtica de Barcelona, durante el curso 1998/99. Dicha experiencia se centra en el uso de Java como lenguaje de programación, en sustitución de Modula-2.

1.- MOTIVACIÓN

Como en tantas otras facultades y escuelas de informática, el aprendizaje de la programación en la asignatura de *Iniciación a la Programación (IniPro)* de la Facultat d'Informàtica de Barcelona (F.I.B.) de la U.P.C. se ha desarrollado tradicionalmente bajo el paradigma imperativo, alrededor de los conceptos clásicos de notación algorítmica, esquemas de programación y diseño descendente, usando Modula-2 como lenguaje de programación en las prácticas. Los programas desarrollados en las clases de teoría, problemas y prácticas responden al patrón de tratamiento de secuencias y tablas. El entorno de trabajo ha consistido fundamentalmente en el uso de miniordenadores de la casa Digital, si bien últimamente los alumnos también utilizan PCs para el desarrollo de la práctica.

En los últimos años, se constatan dos hechos que interfieren con esta aproximación clásica al aprendizaje de la programación.

a) Aparición de la programación orientada a objetos.

La programación orientada a objetos (O.O.) disfruta hoy en día de un éxito indudable tanto en el mundo académico como en el industrial. Si bien algunos de los conceptos propios de este paradigma ya se habían formulado en el contexto de la programación con tipos abstractos de datos, hay que convenir que no han sido universalmente aceptados hasta su auge.

Creemos que la adopción de este paradigma puede reportar ciertas ventajas en el conjunto de asignaturas de primer ciclo de una ingeniería informática, al usarse los conceptos de herencia, genericidad, vinculación dinámica, etc., para facilitar el aprendizaje. Por lo que se refiere a *IniPro*, aunque no se trata de la asignatura más beneficiada por el cambio, opinamos que la utilización de la O.O. permite plasmar de manera simple algunos conceptos presentados en la asignatura. En especial, la estructuración de programas en clases nos permite sustituir el clásico diseño descendente, que presenta el consabido desequilibrio en el tratamiento de las acciones y los datos, por un diseño modular mediante clases, más equilibrado.

b) Cambio del perfil del alumnado.

La evolución de la informática a todos los niveles y su inserción en la sociedad provoca que el alumno que se matricula en *IniPro* posea unos conocimientos y tenga unas expectativas muy diferentes del alumno tipo de hace unos pocos años. Se corre el riesgo, por lo tanto, de que pueda llegar a desmotivarse no por el contenido mismo del curso, sino por el "envoltorio": ¿dónde están las interfaces gráficas con menús, botones, ventanas y colores?, ¿por qué no se utiliza el *mouse*? Es importante explicar lo que se debe explicar pero de manera que el alumno aprenda lo que quiere aprender, y no siempre es fácil combinar ambos objetivos.

Por todo ello, se decidió iniciar una experiencia piloto en *IniPro* dirigida a paliar ambos efectos. Esta experiencia se traduce en la elección de Java como lenguaje de programación, y se describe en las secciones siguientes.

2.- CONTENIDO DEL CURSO

Antes que nada, debe describirse el programa de la asignatura sin considerar la experiencia piloto. A grandes rasgos, el temario se divide en tres bloques:

- Conceptos introductorios: se describen las nociones de especificación, algoritmo y código, y se enseñan los conceptos básicos asociados a estos elementos. Se insiste en la necesidad de escribir especificaciones pre-post (no necesariamente formales) de los programas, y también en la necesidad de escribir el algoritmo como paso previo de la codificación. Se introducen las funciones y las acciones como mecanismo de encapsulado.
- Esquemas de programación y colecciones de datos: se introduce la noción de esquema como elemento de clasificación y resolución de los programas que se tratan en la asignatura. Se introducen los esquemas de recorrido, búsqueda y mixtos, con diversas variantes, tanto sobre secuencias como sobre tablas.
- Diseño descendente: se muestra esta técnica para construir programas por refinamientos sucesivos. En realidad, el diseño descendente que se enseña en la asignatura se ha orientado a un tratamiento equilibrado de los datos, para paliar el desequilibrio antes mencionado. Eso sí, en ningún momento se ha definido el concepto de abstracción de datos como tal.

En el curso piloto, el temario ha sufrido diversas modificaciones, destacando:

- Se introduce el concepto de clase como mecanismo de encapsulado de datos, y también las nociones asociadas de atributo y método. En cuanto a las variables, se sustituyen por la noción de objeto. Incluso, para poder explicarles alguna biblioteca de Java en la parte de laboratorio, se introduce de manera somera la noción de herencia. La aparición de estos conceptos ha provocado una ampliación de la notación algorítmica usada en las clases de teoría y problemas.
- El diseño descendente se sustituye ahora sí por diseño con clases, donde los datos adquieren ya una entidad propia bien definida. Éste es un punto especialmente delicado puesto que, dados los objetivos de la asignatura y los conocimientos de los estudiantes, no se pretende que éstos sean capaces de realizar diseños complicados. Por lo tanto, los problemas deben ser de complejidad tal que la descomposición en clases sea meridianamente clara, e incluso debe proporcionárseles pistas en caso necesario, para que puedan centrarse en la parte de algorítmica que es el objetivo de la asignatura. Este hecho ha provocado ciertos cambios en el tipo de problemas resueltos en las últimas semanas de clase.

El curso piloto impacta de manera especial en la parte de laboratorio, que cambia de manera radical por la utilización de Java. Destacamos:

- Una pequeña introducción a diversas herramientas para trabajar en la red y facilitar el trabajo desde casa: algo de Internet y HTML, y *telnet* y *ftp*.
- La presentación de parte de la clase AWT. Se ven subclases fáciles de usar como *Button* y *TextField*. Como se explican a principio de curso, los alumnos pueden ir profundizando progresivamente en su conocimiento.
- La traducción de las secuencias directamente a ficheros Java, usando las clases *InputStream*, *OutputStream*, *Read* y *Write*.

3.- IMPLANTACIÓN

Para empezar, algunas cifras. El número de estudiantes matriculados en el primer cuatrimestre superó los 600 y en el segundo, fue cercano a los 500, distribuidos en 6 y 5 grupos, respectivamente. Por lo que respecta al cuerpo docente, el número de profesores asignados a la asignatura el curso 98/99 alcanzó la cifra de 14, más tres becarios docentes encargados de la parte de prácticas y elaboración de material.

Por todo ello, se decidió abrir un único grupo piloto durante el primer cuatrimestre, con dos profesores involucrados de manera voluntaria, y dos becarios docentes creando material a destajo. El devenir del curso, si bien exigente en cuanto a ritmo de trabajo, fue satisfactorio en líneas generales.

Una tarea que merece mencionarse explícitamente fue la coordinación de este grupo piloto con el resto de grupos, pues uno de los objetivos desde el punto de vista de coordinación era asegurar que los conocimientos adquiridos fueron fundamentalmente idénticos a los del resto de grupos. Como prueba palpable de esta uniformidad, el examen y la práctica fueron prácticamente iguales en todos los casos: en los exámenes, los enunciados diferían básicamente en unas pocas palabras (*método* en vez de *acción*, *objeto* en vez de *variable*, etc.); la práctica fue idéntica con una pequeña simplificación para tener en cuenta la mayor carga soportada por los alumnos del grupo piloto durante el curso. Un aspecto positivo fue la posibilidad de efectuar un pequeño examen adicional sobre terminal, para evaluar la asimilación de la parte más práctica de la asignatura.

En el segundo cuatrimestre, la experiencia se amplió a dos grupos pilotos. De nuevo, los profesores involucrados se ofrecieron voluntariamente. Gracias a la existencia del material básico, fue posible elaborar apuntes de curso de la parte de teoría y complementar la parte de laboratorio con más ejemplos.

Debe mencionarse también la transición de los alumnos del grupo piloto del primer cuatrimestre a la asignatura *Programación Metódica (ProMet)* de segundo cuatrimestre. El cuerpo docente de dicha asignatura se ha esforzado en crear el soporte necesario para permitir que estos alumnos pudieran utilizar Java en la parte de prácticas de la asignatura, minimizando así el “aterizaje” desde el grupo piloto.

4.- EVALUACIÓN DE LA EXPERIENCIA

Desde un punto de vista de contenidos, debe empezarse diciendo que los alumnos no han aprendido “más” algorítmica, sino que han utilizado una metodología algo diferente para adquirir básicamente los mismos conceptos, con una componente de “modernidad” añadida, tanto en el paradigma de programación como en el desarrollo de programas. Eso sí, se valora positivamente la transición del diseño descendente a diseño en clases. Debe destacarse también que existen algunos cabos sueltos en la formulación de especificaciones formales manipulando el concepto de objeto.

Por lo que se refiere a los alumnos, debe destacarse la buena acogida general, incluso en el primer cuatrimestre, pese a los problemas lógicos de puesta en marcha. Esta conclusión la extraemos dados los comentarios de los alumnos, y también de la opinión vertida en la revista de los estudiantes de la F.I.B.

Por lo que se refiere a los resultados académicos, son ligeramente mejores que en el resto de grupos. En estos momentos, no estamos en condiciones de afirmar si esta diferencia es realmente debida a los cambios introducidos, o bien a la asistencia más personalizada debida a la mayor implicación del cuerpo docente, o simplemente a las leyes del azar.

Por último, no podemos dejar de mencionar la cita que el diario "El País" hizo de la experiencia en su suplemento de informática *Ciberpaís* (1/10/98).

Pueden encontrarse más detalles en el report del departamento LSI-99-18-R, accesible en <http://www.lsi.upc.es/dept/techreps/1999.html>.

5.- EL FUTURO

El desarrollo de la experiencia piloto durante el curso 98/99 se considera suficientemente satisfactoria, y por ello ya se ha decidido su generalización para los cursos siguientes.

En el curso 99/00, el enfoque presentado en este artículo ya se va a aplicar a todos los grupos de la asignatura *IniPro*. Existe suficiente material docente de soporte de la asignatura, y los profesores de la misma ya han dispuesto de tiempo para estudiar Java. Esta adopción definitiva va a simplificar la tarea de coordinación de la asignatura. Eso sí, dado el volumen de alumnos matriculados en *IniPro*, es indudable que se perderá este seguimiento más individualizado del que han disfrutado los alumnos de los cursos piloto.

Por lo que se refiere al resto de asignaturas de programación de primer ciclo, se pretende implantar progresivamente la transición a la programación O.O. con Java, de la manera que corresponda en cada asignatura. Ya se ha comentado en la sección 3 que la asignatura *ProMet* ha empezado a preparar dicha transición para la primera oleada de alumnos, y durante el primer cuatrimestre del curso 99/00 se seguirá trabajando en esta dirección. Se prevé que en el segundo cuatrimestre del curso 99/00, la sustitución de Modula-2 por Java será mayoritaria.

Sin duda, la asignatura donde el impacto del cambio puede ser más importante es *Estructuras de Datos y Algoritmos*. Actualmente, los tipos abstractos de datos que se estudian son presentados de manera independiente, mientras que la utilización de los conceptos propios de la O.O. nos permitiría engazarlos mediante una jerarquía de herencia, que mostraría claramente sus interrelaciones. Por otro lado, el uso de la vinculación dinámica y del polimorfismo confiere un alto grado de flexibilidad en la programación, y especialmente en la definición de las implementaciones de un tipo abstracto como clases efectivas, herederas de una misma clase abstracta que

representa la especificación del tipo. Se pretende preparar el cambio durante el curso 99/00, para efectuar la transición el curso 00/01.

Y, por último, no puede dejar de citarse que la F.I.B. está actualmente inmersa en un proceso de revisión de los planes de estudio, que podría incidir en alguno de los aspectos mencionados.

6.- AGRADECIMIENTOS

Especialmente, a los alumnos Alberto Vázquez y Javier Vázquez, becarios docentes de *IniPro* en el curso 98/99, cuya labor ha sido crucial en el desarrollo de la experiencia. A Julio Valdés, profesor de *IniPro* durante el segundo cuatrimestre, por su implicación en el proyecto. También a Antonia Gómez, del Laboratorio de Cálculo de la F.I.B., por el soporte técnico proporcionado. Finalmente, a los alumnos que habéis sufrido en vuestras carnes la transición a Java, gracias por la paciencia y los comentarios.