

LA ENSEÑANZA DE JAVA EN LOS ESTUDIOS DE INFORMÁTICA

Elvira Albert¹, Javier Oliver¹, Germán Vidal¹

¹ *Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
e-mail: {ealbert, ffoliver, gvidal}@dsic.upv.es*

RESUMEN: Java es un lenguaje de programación orientado a objetos, concurrente, robusto, portable y adecuado para el desarrollo de aplicaciones distribuidas. Pese a todas sus buenas características, cabe preguntarse si resulta adecuado incluir una asignatura en el plan de estudios de informática que contemple su enseñanza. En este artículo, analizamos esta cuestión y presentamos las características básicas de un curso centrado en la enseñanza de Java. Finalmente, describimos la experiencia en la implantación de una asignatura concreta que sigue las directrices expuestas.

1.- INTRODUCCIÓN.

Actualmente, el lenguaje de programación Java se encuentra en un proceso de continua evolución. La primera versión oficial data de Febrero de 1996. Desde entonces, el lenguaje se ha visto sujeto a un continuo cambio; fundamentalmente, se han ido añadiendo nuevas capacidades: conectividad a bases de datos, *JavaBeans*, RMI (*Remote Method Invocation*), *applets* firmadas, etc. Este dinamismo se ha visto reflejado en los temarios de los cursos de programación en Java, que se han ido actualizando conforme evolucionaba el lenguaje. Pese a todo, el núcleo del lenguaje se encuentra lo suficientemente maduro como para establecer una propuesta de contenidos para un curso de programación basado en Java.

En este artículo nos centramos en los estudios de primer ciclo en Informática. La primera cuestión que se plantea es si resulta adecuado plantear una asignatura en dichos estudios centrada en la enseñanza del lenguaje de programación Java, o si, por el contrario, esta materia es más propia de un curso externo al plan de estudios (por ejemplo, un curso de postgrado).

Por un lado, es indudable el impacto que Java ha conseguido en el ámbito industrial, donde se suelen requerir conocimientos de Java en los candidatos a puestos de trabajo relacionados con tareas de programación (y, especialmente, en áreas relacionadas con Internet). Por otro lado, el lenguaje Java puede también reportar a los alumnos una serie de conocimientos beneficiosos para cursar otras asignaturas del plan de estudios (como, por ejemplo, asignaturas relacionadas con la programación orientada a objetos, distribuida, concurrente o la asignatura *Proyecto Fin de Carrera*, en la que los alumnos a menudo muestran su interés hacia temas relacionados con la realidad empresarial y, por tanto, con el uso de un lenguaje como Java).

Cabe destacar también que, en los últimos años, se está llevando a cabo una remodelación de los contenidos de las asignaturas relacionadas directamente con la programación para incorporar, en muchos casos, las aportaciones que conlleva el uso de Java. Las Escuelas de Informática del

estado español se encuentran en estos momentos en un proceso de reforma de sus planes de estudio. Observando con detalle la información que sobre sus planes de estudio ponen accesible en Internet las diferentes universidades, detectamos que muy pocas han trasladado ya el uso de Java a los cursos básicos sobre programación. En algunos casos, es posible encontrar asignaturas (de carácter optativo, generalmente) centradas en la enseñanza de Java con los nombres más diversos: Programación Orientada a Objetos, Lenguajes de Programación o Fundamentos de Programación II. Con un objetivo distinto, cabe mencionar la experiencia llevada a cabo por la Universitat Politècnica de Catalunya sobre el uso de Java en una primera asignatura de "Introducción a la Programación". Como los profesores implicados en la misma describen en [FG99], el resultado ha sido ampliamente positivo.

Se podría argumentar que nos encontramos actualmente en un transitorio en los contenidos de las asignaturas de Programación en las Escuelas de Informática. La propuesta de asignatura que vamos a presentar se enmarca en ese intervalo de tiempo. Pensamos que en un futuro próximo no será de utilidad incluir una asignatura centrada en Java, porque los contenidos estarán ya desarrollados en diversas materias pero, hasta ese momento, la Universidad debe dar respuesta a la demanda que el mercado laboral exige.

2.- CARACTERÍSTICAS DE LA PROPUESTA.

A continuación presentamos las principales características de una asignatura de "programación en Java" enfocada a cursos avanzados de la enseñanza de programación. Como se ha discutido en la introducción, la materia tiene un carácter (implícitamente) transitorio. Por tanto, pensamos que el carácter más adecuado para la misma es que sea optativa.

Una primera cuestión es el tema de los *prerrequisitos* de la asignatura. Por un lado, sería deseable que los alumnos tuvieran unos conocimientos suficientes de programación (en cualquier paradigma, por ejemplo, en un lenguaje imperativo). Además, ya que Java es un lenguaje orientado a objetos, sería conveniente que tuvieran algunas nociones básicas sobre orientación a objetos (herencia, clases, objetos, etc). Sin embargo, pensamos que este último punto no es imprescindible, ya que nuestra experiencia con alumnos que no poseen dichos conocimientos ha resultado satisfactoria (dedicando relativamente poco esfuerzo a introducir los conceptos más relevantes). Por ello, pensamos que la ubicación más adecuada para la asignatura es el cuarto o quinto cuatrimestre de la titulación.

Como ya se ha comentado, la rápida evolución del lenguaje Java puede dificultar el diseño de contenidos para la asignatura. Asimismo, se trata de un lenguaje muy extenso, por lo que resultaría imposible contemplar su totalidad en el marco de una asignatura optativa típica (6 créditos). Teniendo en cuenta todos los factores apuntados, a continuación describimos las características básicas de nuestra propuesta.

a) Objetivos.

Los objetivos docentes de la asignatura son los siguientes:

- Introducir las nociones básicas asociadas al paradigma de la Programación Orientada a Objetos (POO).
- Desarrollar soluciones algorítmicas a problemas de dificultad media utilizando POO e implementarlas en el lenguaje Java.

- Aprender a construir aplicaciones distribuidas compuestas por código móvil que se ejecuta a través de una red.
- Proporcionar al alumno los conocimientos básicos sobre interfaces de usuario visuales con gestión de eventos.
- Presentar al alumno las características del lenguaje Java que permiten desarrollar aplicaciones relacionadas con Internet.

b) Contenidos Básicos.

Los siguientes contenidos son básicos y deberían contemplarse en el programa de una asignatura cuyo objetivo principal sea proporcionar al alumno los conocimientos suficientes para abordar el desarrollo de aplicaciones industriales en el lenguaje Java.

- *Introducción al paradigma de programación orientada a objetos y al lenguaje Java.* Aquí incluimos características esenciales de la POO, como los conceptos de *clase* y *objeto*, así como los mecanismos de *herencia*, *polimorfismo* y *genericidad*. Respecto al lenguaje Java, consideramos interesante introducir las principales diferencias con otros lenguajes de programación, el concepto de *applet* y su sintaxis básica.
- *Flujos de entrada y salida.* La entrada y salida en Java está basada en el uso de *flujos*. Su conocimiento resulta esencial para el manejo de ficheros, buffers de memoria e, incluso, la entrada/salida estándar. Los flujos son secuencias de bytes que se transfieren de una fuente a un destino a través de un canal de comunicación. La potencia de los flujos se debe a que abstraen los detalles de las operaciones de entrada y salida que se utilizarán en la comunicación.
- *Trabajo en red.* A menudo, entre las aplicaciones cliente/servidor que se encuentran en Internet, sólo un pequeño grupo es utilizado. Entre ellas, se incluyen: ftp, telnet, e-mail, http. En este bloque, se pretende que el alumno aprenda a desarrollar programas que implementen dichas aplicaciones cliente/servidor en red usando las facilidades del lenguaje Java.
- *Aplicaciones distribuidas.* Una aplicación distribuida es una aplicación cuyo procedimiento se distribuye por múltiples computadoras de una red. Las aplicaciones distribuidas pueden servir a la vez a múltiples usuarios y, dependiendo del diseño, conseguir un uso más optimizado de los recursos del sistema. En esta parte, se proporcionará al alumno la información necesaria para comprender las capacidades de programación distribuida a través de Java (mediante la RMI).
- *Programación WWW.* En este bloque se estudiará el acceso al WWW a través de Java, analizando las facilidades proporcionadas por los paquetes estándar del lenguaje. En particular, se debe introducir el desarrollo de *applets*, es decir, pequeños programas Java que se transmiten dinámicamente a través de la red como cualquier otro elemento de un documento HTML.
- *Interfaces de usuario.* El paquete awt es la aproximación que usa Java para la creación de interfaces de usuario independientes de la plataforma con la que se trabaja. Por lo tanto, resulta esencial para todo programador Java aprender cómo se organiza el awt y cubrir las clases que este paquete proporciona.

c) Prácticas de Laboratorio.

Las prácticas de laboratorio se deberían ajustar a los contenidos del apartado anterior, de forma que el alumno pueda llevar al terreno experimental los conceptos vistos en las sesiones de teoría.

Una cuestión relevante es el entorno utilizado para las sesiones de prácticas. El sistema que proponemos utilizar en el laboratorio para compilar programas Java es el JDK (Java Developer's Kit) cuya distribución es gratuita. El hecho de que Java sea gratuito resulta atractivo para las universidades que se enfrentan a presupuestos escasos. Es más, a medida que aparezcan nuevas versiones de Java, éstas estarán disponibles de inmediato a través de Internet, de modo que las universidades pueden mantener su *software* de Java actualizado. No obstante, se le puede ofrecer al alumno la posibilidad de utilizar un entorno de desarrollo más avanzado. Por ejemplo, el entorno de desarrollo Kawa aparece como una capa sobre el JDK. Por tanto, para poder utilizarlo, es necesario disponer del JDK. Entre las ventajas de Kawa, destacamos que puede trabajar con cualquier versión del JDK (por tanto, es fácilmente actualizable) y que tiene un precio asequible.

d) Bibliografía.

Existe una gran cantidad de material adecuado para el desarrollo de los contenidos de esta propuesta. De entre las distintas referencias, vamos a seleccionar algunas, que agrupamos en dos bloques:

- Las referencias [Joy98] y [Pet87] corresponden a textos introductorios al paradigma de programación orientada a objetos.
- El manual de Naughton [Nau96] ofrece una visión general del lenguaje Java, aunque las nuevas versiones del JDK han adoptado un enfoque diferente con respecto a ciertos temas (e.g., la entrada/salida). Una referencia más actualizada es [DD98]. Como manuales de referencia de Java, se recomiendan los textos de Jaworski [Jaw97, Jaw98]. Por último, citamos [AG97] y [Fla97] como material que ofrece una base sólida en estrategias y técnicas de programación en Java.

3.- UNA EXPERIENCIA CONCRETA.

Esta propuesta ha sido llevada a la práctica en la asignatura optativa PVI (Programación Visual para Internet) impartida a alumnos de tercer curso (primer cuatrimestre) de la Escuela Universitaria de Informática de la Universidad Politécnica de Valencia. El programa que se presenta ha sido elaborado para cubrir un total de 6 créditos que se distribuyen en dos partes: 3 para el desarrollo de teoría (y problemas en el aula) y los 3 restantes para prácticas de laboratorio.

El temario concreto de la asignatura PVI es el siguiente:

Tema 1. Introducción a Java

- Introducción a la programación en Java. Sintaxis, tipos, operadores. Control de flujo.*
- Tema 2. Programación Orientada a Objetos
Clase / Objeto. Mecanismos fundamentales de la POO. Otras características.
- Tema 3. Interfaces
Diferencias con clases abstractas. Variables en interfaces.
- Tema 4. Paquetes
Restricción de la visibilidad. Protección de acceso.
- Tema 5. Gestión de cadenas
Constructores. Sintaxis de cadenas. Comparación en cadenas. La clase StringBuffer.
- Tema 6. Gestión de excepciones
Tipos de excepciones. Excepciones asíncronas. Excepciones creadas manualmente. Subclases de excepción propias.
- Tema 7. Entrada / Salida
Archivos como flujos secuenciales de bytes. Entrada y Salida de archivos.
- Tema 8. Trabajo en red
InetAddress. Conexiones mediante URLs. Puertos y conectores.
- Tema 9. Conexiones a Bases de Datos
Estándar JDBC. Acceso mediante SQL.
- Tema 10. Acceso a objetos remotos (RMI)
El modelo distribuido de Java. Implementación de la RMI.
- Tema 11. Applets
HTML básico. Incorporación de datos en un applet. Imágenes y sonido.
- Tema 12. El AWT
Componentes. Organización. Frames. Programación dirigida por eventos.

Las prácticas de la asignatura complementan los temas anteriores. Se trata de sesiones de prácticas dirigidas, de dos horas semanales, en las que los alumnos deben desarrollar pequeñas aplicaciones relacionadas con la materia que se esté estudiando en las sesiones de teoría.

Con respecto a la evaluación, ésta viene determinada por un único examen en el que se valoran los conocimientos adquiridos por el alumno. El examen incluye tanto preguntas acerca de los contenidos teóricos desarrollados a lo largo del curso, como ejercicios de carácter práctico enfocados a la implementación de programas en Java (implícitamente relacionados con el trabajo desarrollado en el laboratorio). Por lo que se refiere a resultados académicos durante el presente curso 1999/2000 (el primero en el que se ha implantado la asignatura PVI), cabe destacar que han sido muy satisfactorios. En la primera convocatoria se obtuvo un 72% de alumnos aprobados (de entre los presentados). El porcentaje ascendió ligeramente en la segunda convocatoria, llegando a un 84% de alumnos aprobados.

Para una información más detallada sobre la asignatura PVI, puede consultarse la URL:

<http://www.dsic.upv.es/users/elp/calbert/pvi.html>

4.- CONCLUSIONES.

Como ya hemos argumentado, el conocimiento del lenguaje de programación Java es, en la actualidad, prácticamente imprescindible para los estudiantes de Informática. Esto se debe, principalmente, a su gran implantación y demanda en la industria (debido, sobre todo, al auge que están tomando Internet y las tecnologías relacionadas). Puesto que aún no se encuentra totalmente asentado en los planes de estudio, la existencia de una asignatura específica para su

desarrollo es, en estos momentos, vital. Desde nuestra experiencia en la impartición de la asignatura PVI, podemos concluir que una propuesta como la que hemos presentado tiene una gran aceptación por parte de los alumnos, que se encuentran muy motivados y eso ha llevado a que los resultados obtenidos hayan sido muy satisfactorios. Pensamos, como ya se ha dicho, que esta asignatura cubre un hueco de forma transitoria, pero necesaria, y que en un tiempo cercano estos contenidos serán posiblemente desarrollados en otras asignaturas a lo largo de la titulación. Hasta ese momento, una asignatura con estas características parece fundamental en cualquier plan de estudios, al menos con carácter optativo.

5.- BIBLIOGRAFÍA.

- [AG97] Arnold, K. y Gosling, J. *El lenguaje de programación Java*. Addison-Wesley/DOMO, 1997.
- [Anu96] Annuff, E. *Java Source Book: A complete guide to creating java applets for the Web*. John Wiley & Sons, Inc, 1996.
- [DD98] Deitel, H. y Deitel, P. *Cómo programar en Java*. Prentice-Hall, Englewood Cliffs, NJ, 1998.
- [FG99] Xavier Franch y Joaquim Gabarró. *Transición de Modula-2 a Java en un curso de iniciación a la programación*. JENUI'99, páginas 319-325.
- [Fla97] Flanagan, H. *Java in a nutshell*. O'Reilly, 1997.
- [Jaw97] Jaworski, J. *Java 1.1 Developer's Guide*. Prentice-Hall, 1997.
- [Jaw98] Jaworski, J. *Java 1.2 al descubierto*. Prentice-Hall, 1998.
- [Joy98] Joyanes, L. *Programación Orientada a Objetos*. McGraw-Hill, 1998.
- [Nau96] Naughton, P. *The Java Handbook*. Osborne-McGraw-Hill, 1996.
- [Pet87] Peterson, G. *Object-Oriented Computing. Volume 1: Concepts*. Computer Society Press of IEEE, 1987.