

# Entorno de programación Java para la docencia y desarrollo de procesadores de lenguajes.

Miguel Mascaró, Catalina Orell

Dept. de Matemàtiques i Informàtica  
Universitat de les Illes Balears  
07071 Palma de Mallorca  
e-mail: [mascport@uib.es](mailto:mascport@uib.es)

## Resumen

El objetivo principal de este artículo es presentar un conjunto de herramientas, que dan soporte a la docencia en las áreas que tocan temas relacionados con los procesadores de lenguajes (compiladores). Este conjunto de herramientas se basa en el lenguaje de programación Java, y consigue integrar en un CD-ROM todo el *software* y la documentación requeridos para que el alumnado disponga de una forma rápida y eficaz, del material necesario para poner en práctica los conocimientos teóricos adquiridos.

## 1. Introducción

Uno de los principales problemas existentes en la docencia de aquellas asignaturas que requieren la confección de una práctica laboriosa y compleja, es la elección del *software* a utilizar por parte del alumnado para desarrollar los trabajos durante el curso.

Por supuesto podríamos decantarnos por la opción más sencilla, dejando en manos del alumnado la elección del *software* a utilizar. La experiencia docente en una asignatura de estas características, como es el caso de *Procesadores de Lenguajes*, nos demuestra que si bien esta opción siempre debe de ofertarse, la complejidad de la realización de un compilador requiere de una atención y un soporte teórico y práctico constante por parte del profesor, y es impensable que este conozca a fondo todas las herramientas existentes en el mercado. Por ello, una buena opción sería ofrecer al alumnado la facilidad de que en todo momento

tenga garantizado un soporte técnico total por parte del profesor.

De la experiencia docente se deduce que para escoger el *software* de nuestro entorno de trabajo, deben de tenerse en cuenta una serie de características básicas que dicho entorno de desarrollo debe poseer:

- Las herramientas que integren el paquete de trabajo deben ser de libre utilización.
- Deben tener una documentación asociada que sea completa y abundante.
- Deben basarse en un entorno de programación práctico, moderno y fácil de utilizar.
- En la medida de lo posible el conjunto de herramientas a utilizar debe de ser de fácil instalación, para evitar la generación de problemas no derivados directamente de la práctica de la asignatura.
- A ser posible la utilización del paquete docente no debe depender de la plataforma *hardware* o *software* utilizada por el alumnado.

Como fruto de la elección de un *software* que cumpla todos los requerimientos enumerados anteriormente, se ha realizado un CD-ROM que contiene todos los elementos necesarios para dar soporte a la docencia de la asignatura Procesadores de Lenguajes [12].

A continuación expondremos todos los aspectos particulares del paquete para el desarrollo de procesadores de lenguajes “*Analizadores de información*”, acompañando todos los puntos detallados, con referencias a las experiencias reales obtenidas durante la realización de la docencia de dicha asignatura.

Esta ponencia se complementa con un CD-ROM que incluye el paquete docente presentado.

## 2. El paquete de ayuda a la docencia “Analizadores de Información”

El paquete de *software* presentado es una recopilación de herramientas de programación en Java [1],[5], que se integran en un único paquete de desarrollo, con la finalidad de dotar al usuario de dicho entorno, es decir al alumnado, de un sistema didáctico y completo que oriente sus esfuerzos a la puesta en práctica de los conocimientos adquiridos en las clases teóricas, sin sobrecargar el trabajo de este con complicaciones colaterales como pueden ser problemas derivados de la instalación, integración de paquetes dedicados a la obtención de *Interfaces Gráficas*, compatibilidad del *software* con el sistema operativo, etc.

El entorno de desarrollo “Analizadores de Información” consta de los siguientes elementos:

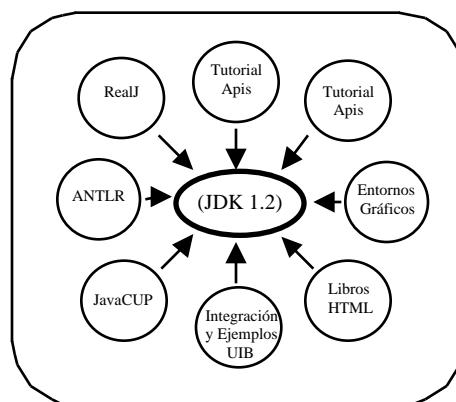
- Entorno de desarrollo en Java, JDK 1.2.2, [5],[11].
- Entorno integrado de desarrollo *RealJ* para la programación en Java, [7].
- ANTLR: Herramienta de Generación de Analizadores de Información (*parsers*) de tipo  $LL(n)^1$ , [8].
- JavaCUP: Herramienta de Generación de Analizadores de Información (*parsers*) de tipo  $LR(n)$ -LALR<sup>2</sup>, [9].
- Tutoriales de Java. Versión 1.3 en formato *html*, [11].
- Apis completos de Java. Versión 1.3 en formato *html*, [11].
- Tutorial de ANTLR en formato *html*, [8].
- Entorno de desarrollo gráfico, Java 2D, [11].
- Entorno de desarrollo de Interfaces Gráficas *Swing*, [6],[11].
- Versión inglesa del libro “*Thinking in Java, 2<sup>nd</sup> edition (rev. 9)*”, [1].

<sup>1</sup>  $LL(n)$ : Analizador descendente con  $n$  símbolos de anticipación.

<sup>2</sup>  $LR(n)$ : Analizador ascendente con  $n$  símbolos de anticipación.  $LR(0) \{ LALR \} LR(1)$ .

- Documentación y ejemplo de un visor de *html* construido usando ANTLR, [12].
- Documentación y ejemplos de JavaCUP, [9].
- Intérprete de código de tres direcciones (pseudo ensamblador) programado en Java.

Además el CD-ROM tiene función de auto instalación si se coloca en una máquina con sistema operativo *Windows*, i se posee otro sistema operativo puede instalarse la jerarquía del disco directamente sin ningún problema.



El paquete Analizadores de Información no es solo una herramienta para la generación de compiladores, es un potente entorno de programación multiplataforma.

## 3. Adaptación a los requerimientos por parte del paquete presentado

Veamos los diferentes aspectos del paquete de *software* presentado.

Todos los elementos integrados en el CD-ROM son de libre utilización para el uso docente. El entorno de programación *JDK 1.2*, el entorno integrado de desarrollo *RealJ*, y todos los *Apis* y tutoriales sobre estos son propiedad de *Sun Microsystems* [11], y este ha liberalizado su uso para tareas docentes. Las herramientas de generadores de *parsers* *ANTLR* y *JavaCUP* son propiedad de *Magelang Institute* [8],[9], y de la Universidad de *Princeton* respectivamente, siendo también de libre utilización para el uso docente. El libro “*Thinking in Java*” incluido en el paquete docente es de libre utilización para fines u objetivos no lucrativos [1]. El resto de elementos

incluidos en el CD-ROM han sido elaborados por la Universitat de les Illes Balears [12], siendo también de libre utilización para tareas dirigidas a la docencia.

La documentación existente para cada una de las herramientas es más que suficiente. Además de existir manuales y ejemplos para cada una de las herramientas, el *software* dispone también de todos los *Apis* (descripción de librerías de programación) de Java SUN, que en todo momento dan soporte a la programación Java. Para terminar el entorno de programación presentado, dispone de una publicación electrónica que es uno de los manuales de Java más completos que existen. Todo ello por supuesto de libre uso para tares docentes.

Todo el conjunto de herramientas descrito se basa en el lenguaje de programación Java. El lenguaje Java es un lenguaje moderno, orientado a objetos, de propósito general, y dotado de una gran cantidad de librerías disponibles para todo tipo de fines y utilidades. El JDK de *Sun Microsystems* es el entorno de desarrollo estándar para la programación Java. En nuestro caso el JDK 1.2 posee todas las herramientas necesarias para desarrollar aplicaciones que funcionen sobre una máquina virtual Java, pero para facilitar la creación de proyectos se ha incluido en el CD-ROM el entorno de programación gráfico interactivo *RealJ*, cuyo funcionamiento es similar al de la mayoría de entornos de programación *IDE*<sup>3</sup>.

Todo el *software* integrado en el CD-ROM es de fácil instalación, y esta viene detallada ampliamente en un documento de ayuda en formato *html*. En el caso de que poseamos un SO de la familia *Windows* el software se instala automáticamente al introducir el CD-ROM en el ordenador. Si el equipo de que disponemos ya posee soporte de programación Java JDK 1.2 o superior, bastará con arrastrar el árbol de directorios dentro del *home* de este, para que todo el sistema de generación de analizadores léxicos y sintácticos que hemos preparado comience a funcionar. Todas estas facilidades han reducido considerablemente el tiempo de preparación del soporte práctico de la asignatura, ya que el

alumnado no pierde tiempo en problemas secundarios, derivados de la instalación y de la configuración de variables de entorno necesarias. En concreto basándonos en la experiencia práctica del curso actual, es la primera vez en la que se han podido realizar experimentos prácticos a los 10 días de haber empezado la docencia de la asignatura.

A todo ello hay que añadir que al no depender el entorno de trabajo de la plataforma *software* o *hardware* de que se disponga, el alumnado puede traspasar sus programas fuente de un sistema de programación a otro, sin que ello modifique para nada ni la estructura de compilación y ejecución de los programas, ni el resultado ni la apariencia de estos.

#### **4. Utilización del software docente presentado en la asignatura procesadores de lenguajes**

El paquete de desarrollo “*Analizadores de Información*” ha sido puesto en práctica por primera vez en el curso actual 2000/2001. Transcurridas las tres cuartas partes del curso los beneficios obtenidos son evidentes.

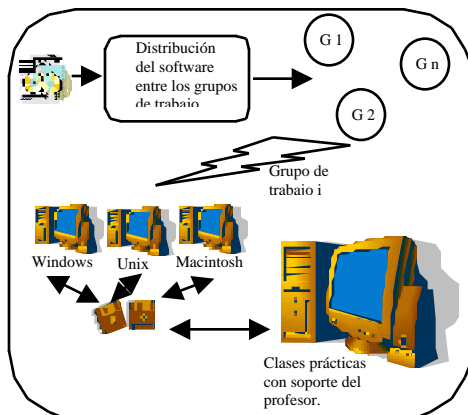
En la etapa de instalación del *software* los resultados han sido muy satisfactorios por tres motivos fundamentales. El primero de ellos es que el estar todo el paquete integrado en un solo CD-ROM, ha posibilitado que en el transcurso de una semana todo el alumnado tuviera el sistema, ya que a partir de una serie de copias iniciales, estas se han ido distribuyendo de manera rápida y sencilla, sin problemas de tamaño, conexión, licencia, etc. El segundo es que la instalación del entorno de trabajo se hace de forma automática en entornos que disponen de un sistema operativo de la familia *Windows*, y en el resto también se realiza de manera sencilla con unas pocas declaraciones de variables de entorno, lo cual ha facilitado que en la primera semana del curso todo el mundo dispusiera del entorno de trabajo en su ordenador personal. La tercera causa es la de que el paquete de *software* funcione de manera idéntica con independencia del sistema informático que se posea, lo cual posibilita que el alumnado con un simple *disquete* pueda llevar consigo los ficheros fuente de su proyecto, y

---

<sup>3</sup> Integrated Development Environment.

pueda reproducir el proceso de compilación y desarrollo de forma idéntica en los ordenadores que tiene asignados en la Universidad.

Por otra parte al realizarse las prácticas en grupo, los integrantes de estos pueden dividirse perfectamente el trabajo sin tener en cuenta para nada el sistema informático que se posea.



Proceso generalizado de distribución y puesta en funcionamiento, del entorno de programación aplicado a las prácticas de la asignatura "Procesadores de Lenguajes".

Para enumerar las ventajas que aporta el paquete de *software* docente a la asignatura "Procesadores de lenguajes", lo que haremos es examinar las etapas del proceso de generación de un compilador [2],[3],[4]:

- **Análisis léxico:** En la etapa del análisis léxico, se debe de separar un fichero de entrada que consta de una serie de caracteres ASCII en *tokens*, es decir en grupos de caracteres que tengan un significado propio e indivisible, y al mismo tiempo almacenar información sobre estos. Esto supone programar una serie de estructuras de almacenamiento (Tabla de Símbolos), y los autómatas finitos deterministas correspondientes. En el CD-ROM presentado se halla todo un entorno de generación de analizadores léxicos, el JLEX, el cual a partir de las expresiones regulares de los *tokens* es capaz de generar el autómata finito determinista correspondiente. Por otra parte algunas de las características del JDK 1.2, como la existencia de una clase java que implementa una tabla de *hashing*, facilitan la

creación de las estructuras alternativas (tabla de símbolos, listas, etc...).

- **Análisis sintáctico y semántico:** Estas dos fases del proceso de compilación requieren de la creación de analizadores de frases con código semántico incorporado. Nuestro CD\_ROM contiene un conjunto de librerías de generación de analizadores ascendentes y descendentes. El programa java ANTLR sirve para construir en código java analizadores descendentes para gramáticas del tipo LL(n). Para analizadores del tipo ascendente (LALR), tenemos otra aplicación que genera *parsers* LALR en código java, y que enlaza con los analizadores léxicos producidos por el JLEX, esta aplicación sintáctica/semántica es el Java-CUP.
- **Código intermedio y optimización:** Para las etapas del código intermedio y de la optimización, se pueden utilizar el conjunto de librerías que ofrece el JDK 1.2. Este nos aporta toda una serie de facilidades para el manejo de ficheros, cadenas de caracteres, creación de estructuras de la información, etc. Además como la base del lenguaje Java es el C++, que generalmente es el que se usa para crear la mayoría de módulos de un compilador, la práctica nos acerca bastante a las herramientas utilizadas en el mercado, con la ventaja de que el Java libera al programador de todos los aspectos relacionados con el acceso y reserva de la memoria. Aunque el lenguaje Java tiene el inconveniente de que el código ejecutable es muy lento, ya que este se ejecuta sobre la máquina virtual, esto no tiene importancia en nuestro caso ya que no se trata de hacer un compilador comercial, sino de fabricar un compilador para entender correctamente el proceso de creación de este.
- **Generación de código:** Una vez que se ha manipulado el código intermedio a voluntad, el compilador debe de generar código máquina para una CPU en cuestión. Como la experiencia nos demuestra que es imposible en un solo curso, abarcar desde la etapa del análisis léxico hasta la generación de código CPU (óptimo), hemos optado por generar código intermedio optimizado para una CPU virtual. Para ello en el paquete de *software* se incluye un intérprete de ese código CPU

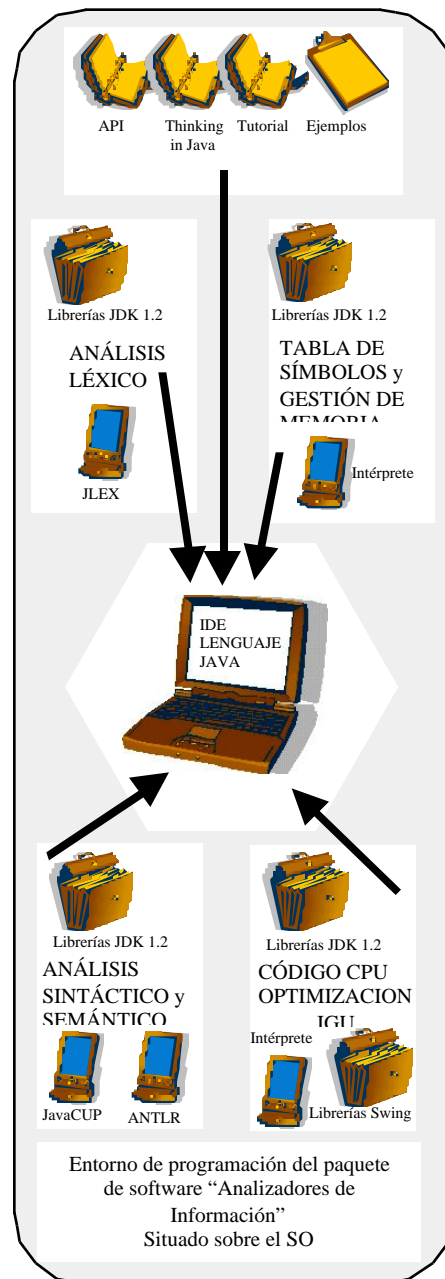
(virtual) realizado en Java. Con esta aplicación se solucionan una serie de factores:

- Las tareas de gestión de memoria del código generado las realiza el intérprete por sí mismo. Con ello se consigue reducir la carga de programación del alumnado, pero sin liberarlo de poseer los conocimientos necesarios para comprender perfectamente el funcionamiento de la gestión de memoria, ya que de no ser así no podría generar el código intermedio para la CPU (virtual) correspondiente.
- Al disponer el entorno de una CPU virtual, el sistema sigue siendo independiente de la plataforma *software* i *hardware* que se posea.
- Al disponer de una CPU virtual se desaparecen los aspectos dedicados a la asignación óptima de los registros de la CPU, *pipeline*, etc.
- Por último como ya hemos comentado anteriormente, el alumno queda liberado de todo el trabajo asociado a la programación del código CPU real. En este punto la experiencia demuestra que con esta técnica, terminan las prácticas en Junio aproximadamente un 30% más de alumnos.
- Integración de las fases del compilador: Para facilitar la integración del código Java resultante de todas las fases del compilador, el paquete presentado dispone de un entorno de desarrollo *IDE*, así como todo un conjunto de librerías que facilitan la programación de las interfaces gráficas del compilador.

### 5. Aplicación del paquete de software docente en la elaboración de un compilador

En el siguiente esquema se pueden observar como se aplican las diferentes características del *software* integrado en el CD-ROM, con el entorno de trabajo en la asignatura Procesadores de lenguajes: entorno de desarrollo *IDE*, compiladores, manuales, *Apis* de programación, documentos electrónicos, ejemplos, soporte para interfaces gráficas de usuario, CPU virtual, etc.

Todo el entorno funciona sobre el SO, con independencia total tanto del propio SO como del *hardware* utilizado.



En este gráfico podemos ver como se relacionan y se integran todas las aplicaciones y herramientas del CD-ROM presentado, con la finalidad de facilitar al alumno la programación de un compilador.

## 6. Conclusiones

En este artículo hemos presentado un entorno completo de desarrollo para cualquier asignatura relacionada con la generación de analizadores de información, aunque variando un poco el contenido del CD-ROM, puede adaptarse a cualquier otro tipo de asignatura que precise de un entorno de programación.

Los beneficios que aporta son cuantiosos, y además disponemos de un entorno de desarrollo completo integrado en un único CD-ROM, que a su vez es de fácil instalación.

Se aprovechan todas las características de un lenguaje de programación moderno, modular, y orientado a objetos, como es el Java. Pudiendo disponer a su vez de la gran cantidad de información que sobre este se dispone a través de la red.

Para finalizar, destacar además de todas las ventajas mencionadas a lo largo del artículo, el hecho de que la totalidad del contenido del CD-ROM es de libre utilización para tareas docentes, con lo cual se puede distribuir entre los alumnos que cursen la asignatura sin ningún tipo de inconveniente.

## Bibliografía

- [1] Thinking in Java, 2nd edition, rev. 9. Bruce Eckel. Año 2000. Publicación electrónica [www.BruceEckel.com](http://www.BruceEckel.com), (Editada también por Prentice Hall )
- [2] Compiladores. Principios, técnicas y herramientas. A.V. Aho, R. Sethi, J.D. Ullman. Addison Wesley Iberoamericana, 1990. ISBN: 0-201-62903-8.
- [3] Compiladores. Conceptos fundamentales. Teufel, Schmidt, Teufel. Addison Wesley Iberoamericana, 1995. ISBN: 0-201-65365-6.
- [4] Compiler Construction. Nicklaus Wirth. Addison Wesley, 1996. ISBN: 0-201-40353-6.
- [5] Descubre Java 1.2. Mike Morgan. Prentice Hall, 1999. ISBN: 0-7897-1627-5.
- [6] the Elements of user interface design. T. Mandel. Wiley Computer Publishing, 1997. ISBN: 0-471-16267-1.
- [7] Documentación del IDE de desarrollo RealJ en propiedad de Sun Microsystems. <http://www.realj.com>.
- [8] ANTLR Magellan Institute manual. <http://www.antlr.org>.
- [9] JavaCUP home page. Georgia Institute of Technology / Princeton University. <http://www.cs.princeton.edu/~appel/modern/java/CUP>.
- [10] Jlex Manual Home Page. Princeton University. <http://www.cs.princeton.edu/~appel/modern/java/JLex>.
- [11] Tutorial i Apis de programación de Sun Microsystems. <http://java.sun.com>.
- [12] Proyecto de final de carrera titulado “Entorn de Desenvolupament d’Analitzadors Léxics i Sintáctics en Java”. Catalina Orell Más – Miguel Mascaró Portells. Noviembre de 2000. UIB. Publicación Interna UIB.