

# La programación declarativa en los estudios de Ingeniería Informática

Alberto Gómez Mancha

Departamento de Informática  
Universidad de Extremadura  
agomez@unex.es

## Resumen

*En este trabajo se tratará la asignatura 'Programación Declarativa' que se imparte desde hace dos años en los estudios de Ingeniería Informática de la Universidad de Extremadura en su Escuela Politécnica de Cáceres.*

*En primer lugar se comentarán las características de esta asignatura (créditos, número de alumnos, situación dentro del plan de estudios, etc.), objetivos, contenidos, métodos de evaluación, así como su relación con otras asignaturas dentro de nuestro plan de estudios.*

*Para situar estos contenidos, se realiza una comparación de las asignaturas de programación declarativa en otras universidades españolas, a través de los datos disponibles en sus páginas WEB (quizás por este motivo la información no es completa, aunque sí significativa).*

*Para terminar, se harán algunos comentarios sobre el estudio realizado y se tratarán las experiencias observadas en la docencia de esta materia.*

## 1 Introducción

En este trabajo se va a tratar la enseñanza de programación declarativa en los estudios de Informática en algunas universidades españolas y especialmente en la Universidad de Extremadura. El objetivo es comparar la información que sobre las asignaturas podemos encontrar en Internet para ver si existe un acuerdo en los contenidos, las prácticas y la bibliografía entre los docentes de

esta materia, o si, por el contrario, existen muchas discrepancias.

Bajo el nombre de programación declarativa se suelen agrupar dos paradigmas de programación distintos a la programación imperativa habitual: la programación lógica y la programación funcional.

A diferencia de la programación imperativa, que está basada en el modelo de máquina en el que se ejecutan los programas, la programación lógica y funcional tienen una base matemática: la lógica y el lambda-cálculo y las funciones, respectivamente.

A la hora de resolver un problema con un lenguaje de programación lógica no se especifica paso a paso la lista de instrucciones que hay que realizar para solucionarlo (como se hace en un lenguaje de programación imperativo), sino que se declaran los hechos y las relaciones entre ellos, de manera que luego se pueda deducir automáticamente si otro hecho es cierto o no a partir de las premisas que forman el programa.

En los lenguajes de programación funcionales, la única estructura con la que se trabaja es la función. Todo programa es una función que, a partir de unos datos de entrada devuelve un resultado. La composición de funciones es la herramienta básica para construir los programas.

El campo de la programación declarativa es bastante activo actualmente y se realiza mucha investigación, aunque quizás no sean demasiado utilizados en aplicaciones reales fuera de la universidad.

El segundo apartado de este trabajo trata sobre la asignatura 'Programación declarativa' en los estudios de informática de la Universidad de Extremadura. En el siguiente se estudia y compara

la información que sobre esta asignatura se encuentra en las páginas que las universidades ponen en Internet. En el apartado 4 se comentan algunos de los resultados obtenidos en el estudio anterior. Para terminar, el autor expone sus experiencias en la docencia de esta asignatura y propone posibles mejoras según el estudio realizado.

## 2 La programación declarativa en la EUP de Cáceres

Hace un año aproximadamente terminaron sus estudios los alumnos de la primera promoción de Ingeniería Informática en la Escuela Politécnica de Cáceres de la Universidad de Extremadura. Las asignaturas del 2º ciclo se están impartiendo este curso por tercera vez en el caso de las troncales y obligatorias. Por tanto, los contenidos son bastante variables y se están modificando y mejorando cada año.

La asignatura de 'Programación declarativa' se empezó a impartir el curso 96-97. En el plan de estudios aparece como una asignatura optativa o de libre elección para el segundo ciclo.

Es una asignatura cuatrimestral (primer cuatrimestre) que aparece en el horario del cuarto curso, aunque hay alumnos de último año que también la eligen.

La carga docente es de 6 créditos, de los cuáles 3 son de teoría y 3 de práctica. Esto supone 2 horas de clases teóricas y 2 horas en un laboratorio a la semana. Las clases teóricas las reciben en un solo grupo; para las clases prácticas se dividen en dos grupos. En los dos años que se ha impartido, se han matriculado 60 y 50 alumnos, respectivamente.

En las clases teóricas se explican los conceptos fundamentales sobre la materia y se realizan ejercicios para ilustrarlos. El temario está dividido en dos bloques principales; el primero dedicado a la programación lógica y el segundo a la programación funcional. También hay un tema previo introductorio sobre la programación declarativa en general.

En las clases prácticas, llevadas a cabo en un laboratorio de PC, se proponen nuevos ejercicios para que los alumnos trabajen con los conceptos tratados en las clases de teoría.

Las prácticas de la parte de programación lógica se realizan en el lenguaje más utilizado de este paradigma, Prolog, concretamente en SWI-Prolog, un intérprete de la Universidad de Amsterdam de libre acceso [Prolog].

Para las prácticas de programación funcional se usa Gofer, un lenguaje funcional (también gratuito) que implementa un subconjunto de Haskell [Gofer].

La evaluación de la asignatura se realiza a través de un examen escrito y de la entrega de prácticas para cada uno de los dos bloques en los que se divide la asignatura. Es necesario sacar un mínimo de 4 en cualquiera de las pruebas para hacer media. Las prácticas representan una tercera parte de la nota final.

La asignatura de 'Programación declarativa' aparece un poco apartada del resto de las asignaturas del plan de estudios actual. Podría ser de gran ayuda para 'Inteligencia Artificial', pero al ser ésta una asignatura troncal de 4º curso, y 'Programación declarativa' optativa, no se puede suponer que los alumnos tengan conocimientos de Prolog para realizar las prácticas. (La solución en los cursos pasados fue realizar seminarios de Prolog para los alumnos de 'Inteligencia Artificial'.) Hay algunas asignaturas en el último curso de la Ingeniería Informática en las que se tratan temas relacionados ('Arquitecturas avanzadas', 'Procesadores de lenguajes', 'Bases de datos avanzadas'), pero al no haber un sistema de prerequisites entre las asignaturas, no se pueden aprovechar las materias explicadas en otras asignaturas, con la consiguiente pérdida de tiempo (hay que explicar lo mismo en distintas asignaturas sin poder profundizar demasiado) e interés (para los alumnos que sí han cursado las asignaturas).

## 3 La programación declarativa en las universidades españolas

¿Hay una serie de temas comunes que deben incluirse en el temario de un curso introductorio de programación declarativa? Ésta es la pregunta que surgió inicialmente y a la que este trabajo quiere responder. En este apartado se va a comparar la información que sobre la asignatura 'Programación declarativa' se puede encontrar en las páginas WEB de las distintas universidades españolas. De ahí se intentará establecer qué temas

se suelen explicar, los lenguajes de programación más utilizados, los libros que más se recomiendan, etc.

El principal problema a la hora de realizar este estudio ha sido lo heterogéneo de la información y de la manera de presentarla en las distintas páginas. No todas las universidades incluyen información detallada sobre las asignaturas: algunas simplemente presentan el nombre y algún dato académico complementario; en otros casos, no aparece información de ningún tipo. Este estudio no puede considerarse completo ni representativo de lo que pasa en todas las universidades, al no tener información completa y homogénea de todas ellas, aunque sí proporcionará una idea general a partir de un conjunto suficientemente grande (11) de asignaturas.

Los datos con los que se trabaja en este artículo se recogieron durante la primera semana de junio de 1998. Sólo se han incluido las asignaturas de aquellas universidades que presentaban una información detallada (datos académicos, temario, bibliografía).

De las once asignaturas, de otras tantas universidades, elegidas para elaborar el informe, 7 pertenecen a los estudios de Ingeniería Informática, y las cuatro restantes a los de Ingeniería Técnica en Informática de Sistemas.

Se han encontrado algunas asignaturas íntegramente dedicadas a programación funcional [EHU] o lógica [UV] que no se han incluido en el trabajo.

En el primer subapartado se presentan las universidades que se van a considerar, junto con los datos académicos de las asignaturas. En el siguiente se incluye un temario que refleja las materias tratadas en la mayoría de las universidades, tanto para el bloque de programación lógica como para la parte de funcional. En el tercer subapartado se comentan las prácticas y los métodos de evaluación empleados. Para terminar, se presentan los libros que se suelen recomendar en las distintas universidades.

### 3.1 Datos académicos

En la siguiente tabla aparecen los datos de la asignatura 'Programación declarativa' de las

distintas universidades que se han tenido en cuenta en este trabajo.

De cada una de ellas se presenta el nombre de la universidad, junto con la abreviatura que usaremos a lo largo del estudio para referirnos a ella, el carácter que tiene dentro del plan de estudios (Obligatoria, Optativa o de Libre Elección), el número de créditos, separados en créditos teóricos más créditos prácticos, el curso al que se adscriben y la duración anual o cuatrimestral (semestral según algunas universidades, aunque la duración es equivalente).

Universidad	Ob/Op/LE	Créditos (teóricos+prácticos)	Curso	Duración
UCLM Castilla-La Mancha	Op	9	2 *	Anual
UCM Complutense de Madrid	Ob	12 (8+4)	2C	Anual
UDC Coruña	Ob	6 (3+3)	3	Cuatr.
UDG Girona	Op	4.5 (3+1.5)	3 *	Cuatr.
UEX Extremadura	Op	6 (3+3)	2C	Cuatr.
UGR Granada	Ob	6 (3+3)	3	Cuatr.
UMA Málaga	Ob	6	3	Cuatr.
UNED Educación a Distancia	Op	7	3 *	Cuatr.
UPC Politécnica de Catalunya	Op	6 (3+3)	2C	Cuatr.
UPM Politécnica de Madrid	LE	6	3 *	Cuatr.
UPV Politécnica de València	Op	6 (3+3)	3	Cuatr.

Las asignaturas en las que aparece un asterisco en la columna de curso (UCLM, UDG, UNED, UPM) son de estudios de Ingeniería Técnica de Informática de Sistemas. El resto pertenecen a la Ingeniería Informática.

No todas las páginas visitadas ofrecen información de las asignaturas que deben estudiarse antes o a la vez que ésta. Algunas, al ser de segundo ciclo, ya tienen asignaturas del primer ciclo como prerequisites [UEX] [UPC]. En general, si se detalla este dato, se incluyen asignaturas de programación como 'Estructuras de datos' [UCM] [UPM] o 'Introducción a los esquemas algorítmicos' [UPC], y asignaturas de introducción a la lógica [UDG] [UPC].

Sólo en 4 universidades la asignatura es obligatoria en los planes de estudio; en el resto de

los casos es una asignatura optativa o de libre elección.

En la mayoría de los casos esta asignatura es cuatrimestral y con 6 créditos de carga docente. Sólo en los dos casos en que se superan los 9 créditos, la asignatura es anual.

### 3.2 Temarios

La parte más interesante de este estudio es la comparación de los temarios de esta asignatura en las distintas universidades consideradas, para comprobar si existe una cierta uniformidad en los contenidos expuestos.

Aunque la información con la que se contaba no mantenía el mismo nivel de detalle en todos los casos, se puede considerar que los temarios son bastante similares en casi todos los casos, con diferencias de contenido y enfoque mínimas.

Generalmente, tras un tema introductorio a la programación declarativa (paradigmas fundamentales, características, utilización), se divide la asignatura en dos bloques diferenciados, uno dedicado a la programación lógica y otro a la funcional.

Estudiando la parte de los temarios dedicada a la programación lógica, podemos obtener este temario básico:

#### Programación lógica

- Repaso de lógica de primer orden
- Fundamentos de programación lógica
- Cláusulas
- Unificación y resolución. Algoritmos de resolución
- Semántica de programas lógicos
- Semántica para la negación
- El modelo de ejecución de Prolog
- Introducción al lenguaje Prolog
  - Hechos, predicados, reglas, variables
  - Unificación
  - Backtracking
  - Aritmética
  - Entrada/salida
- Control en Prolog
  - Corte y fallo
  - Usos y problemas con el corte

- Negación como fallo
- Estructuras de datos en Prolog
  - Listas
  - Árboles

El lenguaje utilizado en todos los casos como vehículo para explicar los conceptos es Prolog.

Como temas adicionales de programación lógica que aparecen sólo en algunas universidades:

- Gestión de la base de reglas
- Técnicas avanzadas de programación en Prolog
- Gramáticas lógicas
- Metaintérpretes Prolog
- Uso de Prolog en la Inteligencia Artificial
- Programación lógica con restricciones
- El lenguaje Visual Prolog

En el caso de la programación funcional, también existe bastante uniformidad en las materias expuestas, aunque hay mayor discrepancia en el lenguaje utilizado para aplicarlas.

#### Programación funcional

- Características de los lenguajes funcionales
- Elementos de un lenguaje funcional
  - Definición de funciones
  - Tipos
  - Evaluación perezosa (*lazy*) e impaciente (*eager*)
  - Currificación
- Sistema de inferencia de tipos
- Funciones de orden superior
- Sobrecarga de funciones
- Listas y tuplas
- Estructuras de datos infinitas

Dependiendo del lenguaje funcional que se emplee en las prácticas, se incluyen explicaciones adaptadas al lenguaje en particular.

Temas adicionales que no están presentes en todos los temarios:

- Introducción al lambda-cálculo
- Árboles
- Corrección de programas

En general, la presencia de la programación declarativa se limita a una asignatura de

introducción en todo el plan de estudios. Sin embargo, en dos de las universidades estudiadas hay asignaturas complementarias que profundizan en este campo.

En los estudios de Ingeniería Informática de la Universidad de Málaga se puede cursar la asignatura optativa de segundo ciclo 'Programación declarativa II', cuatrimestral con 4.5 créditos [UMA2].

En la Universidad Politécnica de Valencia se cuenta con tres asignaturas optativas de 6 créditos cada una: 'Programación funcional', 'Programación lógica' y 'Programación lógica visual', dentro de una especialización en los estudios de Ingeniería Informática [UPV2].

### 3.3 Prácticas y evaluación

Como complemento a la teoría explicada, en todos los centros se realizan prácticas con determinados lenguajes para que el alumno comprenda y aplique los conceptos teóricos.

Universidad	P. Lógica	P.funcional	Evaluación
UCLM	Prolog	Gofer/LISP	Perezosa
UCM	Prolog	Miranda	Perezosa
UDC	Prolog	ML	Impaciente
UDG	Prolog	Miranda	Perezosa
UEX	Prolog	Gofer	Perezosa
UGR	Prolog	LISP	---
UMA	Prolog	Haskell	Perezosa
UNED	Prolog	Gofer	Perezosa
UPC	Prolog	Miranda	Perezosa
UPM	Prolog	CAML	Impaciente
UPV	VisualProlog	Gofer	Perezosa

La utilización de Prolog es unánime como lenguaje para la programación lógica.

En el caso de la programación funcional la diversidad de lenguajes es mayor. Podemos establecer dos grandes grupos, según el lenguaje elegido sea de los que realizan evaluación perezosa o 'lazy' de los argumentos de la función (sólo se evalúan si se necesitan, como Miranda,

Gofer o Haskell) o bien evaluación impaciente, voraz o 'eager' (antes de aplicar la función se evalúan todos los argumentos, se vayan a utilizar o no, como en los lenguajes de la familia de ML).

Sólo en un caso [UGR] se usa como lenguaje funcional LISP, que según muchos autores no puede considerarse como un lenguaje totalmente funcional, aunque es el precursor de todos ellos. En la universidad de Castilla-La Mancha se estudia LISP como complemento a Gofer. (Para mayor información sobre lenguajes funcionales se puede consultar [FAQ]).

Aunque en todos los casos se emplea una prueba escrita para evaluar los conocimientos adquiridos por el alumno, las prácticas son obligatorias en la mayoría de los casos, en otros son optativas [UCM] y en algunos parece que no se tienen en cuenta a la hora de la evaluación final [UNED].

### 3.4 Bibliografía

Otro punto interesante de este estudio es la comparación de los libros que se utilizan como referencia en esta asignatura en las universidades consideradas. De las referencias bibliográficas que aparecen al final del artículo, las más repetidas son las siete primeras, cuatro de las cuáles tratan de programación lógica y Prolog ([1] a [4]), y tres de programación funcional ([5] a [7]).

Los libros más utilizados para programación lógica son el de Clocksin y Mellish [1], un libro relativamente antiguo pero que sigue siendo fundamental para el estudio de Prolog y de la programación lógica; y el de Sterling y Shapiro [4], más reciente y que profundiza en la programación en Prolog.

En cuanto a la literatura de programación funcional, el libro más referenciado es el de Bird y Wadler [5], uno de los considerados fundamentales en este campo.

En la tabla siguiente se muestran las universidades (excepto UPM, que no presentaba información bibliográfica en la página WEB) junto con los libros recomendados como básicos (marcados con una B) o como complementarios (con una C). Los números que aparecen en la columna *Otras* son referencias a otros libros que sólo se proponen en la universidad correspondiente.

Universidad	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Otras
UCLM	B		B	B	B	B		8,9,10
UCM				C	B	C	C	11,12
UDC				B				13
UDG			B				B	10
UEX	B	B		C	B	C		
UGR	B							14,15
UMA	B	B	B	B	B			16, 17
UNED					B			18, 19
UPC	B	B				B	B	
UPV	B		B	B				20

#### 4 Comentarios al estudio

Aunque la información con la que se ha realizado este estudio no es completa ni homogénea, ya que no se pueden encontrar los mismos datos de esta asignatura en las páginas WEB de todas las universidades, se pueden extraer algunas conclusiones útiles sobre la programación declarativa en los estudios de Informática.

En general, las características académicas de las asignaturas son similares. 'Programación declarativa' suele ser una asignatura cuatrimestral de 6 créditos, en 3º ó 4º curso, optativa. Normalmente es una asignatura aislada dentro del plan de estudios, aunque dos universidades ofrecen asignaturas para profundizar en estos temas [UMA] [UPV].

Los temarios de las asignaturas consideradas son bastante parecidos en cuanto a los temas que se tratan. En todos los casos, tras un tema introductorio a la programación declarativa, la asignatura se divide en dos bloques bien diferenciados, uno dedicado a la programación lógica y otro a la programación funcional.

La mayor diferencia se detecta en el lenguaje funcional elegido para realizar las prácticas, aunque puede apreciarse una preferencia por los lenguajes funcionales con evaluación perezosa. Con relación a la programación lógica, Prolog es el lenguaje utilizado en todas las universidades.

La bibliografía que se recomienda también es bastante similar en todos los casos. Los libros

que más se repiten son los considerados claves en la literatura declarativa.

Como trabajo futuro, sería bueno completar el estudio con todas las universidades donde se imparten cursos introductorios de programación declarativa (como asignatura única o dividida en varias asignaturas distintas). Los datos deberían tomarse a partir de un modelo uniforme, y no a través de la información a la que se puede acceder a través de Internet, que no es completa en algunos aspectos.

#### 5 Conclusiones

A la vista del estudio comparativo de las asignaturas que tratan la programación declarativa en otras universidades, la que se desarrolla en los estudios de Ingeniería Informática de la Escuela Politécnica de Cáceres es bastante similar al resto en cuanto a características académicas, temario, lenguajes de programación utilizados y bibliografía. El curso 97-98 es el segundo año en que se ha impartido esta asignatura, y por tanto todavía se están realizando cambios en el contenido, las prácticas, etc.

Este último curso se realizó una encuesta entre los alumnos para poder valorar si la asignatura había satisfecho las expectativas que tenían a la hora de matricularse. En general estaban contentos con el desarrollo de la asignatura, aunque el principal problema que veían era la poca aplicabilidad de los temas estudiados en aplicaciones reales. Una parte de culpa es de los ejercicios y prácticas desarrollados a lo largo del curso, que son breves y dedicados especialmente a presentar los principales conceptos. En cursos sucesivos habrá que realizar un esfuerzo para proponer ejercicios y prácticas más cercanos a aplicaciones reales.

La otra parte del problema es la poca coordinación existente entre las asignaturas dentro del plan de estudios. Al ser una asignatura optativa y no haber un sistema de prerrequisitos, los profesores de otras asignaturas que podrían utilizar las técnicas que aquí se presentan, no pueden suponer que los alumnos tienen estos conocimientos y, o bien presentan un enfoque distinto o bien explican brevemente y por encima los conceptos fundamentales (con el consiguiente aburrimiento para los alumnos que sí han cursado

‘Programación declarativa’, y el problema de no poder profundizar más en la materia, quedándonos siempre en meras introducciones).

El principal problema observado estos dos años es que a los alumnos les cuesta trabajo olvidar la programación imperativa a la que están acostumbrados. No aprovechan al máximo las ventajas de los paradigmas que aprenden sino que intentan adaptar lo nuevo a las estructuras que ya conocen. En las prácticas se ve claramente esta tendencia, porque están escritas en un lenguaje lógico o funcional pero casi todas ellas tienen un aspecto de programa imperativo. Esto nos llevaría a plantearnos la posibilidad de enseñar un lenguaje no imperativo en el primer curso de programación general. En el caso de los lenguajes funcionales, éstos parecen bastante adecuados para enseñar conceptos fundamentales como tipos, recursividad y modularidad, aunque esta discusión está fuera de los límites de este trabajo.

## Agradecimientos

Queremos agradecer el esfuerzo de las distintas universidades y de los profesores de las asignaturas que han puesto esta valiosa información al alcance de todos en las páginas WEB, y que se encargan de mantenerla actualizada.

También pedimos disculpas por no haber sabido encontrar los datos sobre ‘Programación declarativa’ que otras universidades pueden tener accesibles a través de Internet y no haberlos utilizado en este trabajo .

## Referencias en Internet

- [EHU] (Programación funcional)  
<http://www.ji.si.ehu.es/teaching/docente/Docente/12.3-4/!8056C.html>
- [FAQ] **FAQ for comp.lang.functional**  
<http://www.cs.nott.ac.uk/Department/Staff/gmh/faq.html>
- [Gofer]  
<ftp://ftp.cs.nott.ac.uk/nott-fp/languages/gofer>
- [Prolog]  
<http://www.swi.psy.uva.nl/usr/jan/SWI-Prolog.html>
- [UCLM]  
<http://www.info-ab.uclm.es/sec-ab/plan/progdecl.html>
- [UCM]

- <http://www.ucm.es/info/dsip/ESI137.html>
- [UDC]  
<http://ceu.fi.udc.es/asignaturas/PD.html>
- [UDG]  
<http://sdireps1.udg.es/scripts/assignaturesEPS/assignatura.idc?centre=3105&carrera=105&codi=100339>
- [UEX]  
<http://webepcc.unex.es/~agomez/pd.html>
- [UGR]  
<http://www-etsi2.ugr.es/cgi-bin/nivel.cgi?pagina=/nivel2/planes/i3.htm#podec>
- [UMA]  
<http://www.lcc.uma.es/docencia/ETSIIInf/progdec/pd.html>
- [UMA2]  
[http://www.lcc.uma.es/docencia/ETSIIInf/prog\\_declII.html](http://www.lcc.uma.es/docencia/ETSIIInf/prog_declII.html)
- [UNED]  
<http://www.ieec.uned.es/ieec/pd/>
- [UPC]  
<http://www-fib.upc.es/NovaGuia/Assignatures/PD.html>
- [UPM]  
[http://www.eui.upm.es/jesteinv/assign/pro\\_dec.htm](http://www.eui.upm.es/jesteinv/assign/pro_dec.htm)
- [UPV]  
<http://alumnat.upv.es/pla/asi/2554/index.html>  
<http://www.dsic.upv.es/asignaturas/facultad/pdd/prd.html>
- [UPV2]  
<http://www.dsic.upv.es/asignaturas/eui/l1p/l1p.html>  
<http://www.dsic.upv.es/asignaturas/facultad/prg/plo.html>  
<http://www.dsic.upv.es/asignaturas/eui/l1p/l1p.html>
- [UV] ( Programación lógica)  
[http://sdi.eleinf.uv.es/guia/asignatu/lp\\_llp.html](http://sdi.eleinf.uv.es/guia/asignatu/lp_llp.html)

## Referencias

- [1] W.F. Clocksin, C.S. Mellish. *Programación en Prolog*. Gustavo Gili, 1991.
- [2] Richard A. O’Keefe. *The Craft of Prolog*. The MIT Press, 1990.
- [3] J.W. Lloyd. *Foundations of Logic Programming (2nd edition)*. Springer-Verlag, 1987.
- [4] Leon Sterling, Ehud Shapiro. *The Art of Prolog: Advanced Programming Techniques, 2nd Ed*. The MIT Press, 1994.
- [5] R. Bird, P. Wadler. *Introduction to functional programming*. Prentice Hall, 1988.
- [6] A. Field, P. Harrison. *Functional programming*. Addison Wesley, 1988.
- [7] C. Reade. *Elements of Functional Programming*. Addison Wesley, 1989
- [8] C.L. Chang, R.C. Lee. *Symbolic logic and mechanical theorem proving*. Academic Press, 1973.

- [9] P. Henderson. *Functional programming. Application and implementation*. Prentice-Hall, 1980.
- [10] R. Kowalski. *Lógica, programación e inteligencia artificial*. Díaz de Santos, 1988.
- [11] K. Doets. *From Logic to Logic Programming*. The MIT Press, 1994.
- [12] S. Thompson. *Miranda. The Craft of Functional Programming* Addison-Wesley, 1995.
- [13] Åke Winkström. *Functional programming using standard ML*. Prentice Hall.
- [14] I. Braatko. *Prolog Programming for Artificial Intelligence*. Addison-Wesley, 1990.
- [15] H.P. Winston, Horn. *LISP (3<sup>rd</sup> edition)*. Addison-Wesley, 1991.
- [16] A. Davie. *An Introduction to Functional Programming Systems Using Haskell*. Cambridge Press, 1992.
- [17] B. Ruíz. *Programación funcional en el lenguaje Haskell*. Universidad de Málaga, 1994.
- [18] J. Fokker. *Programación Funcional*. Universidad de Utrecht, 1995.
- [19] R. Sethi. *Lenguajes de programación. Conceptos y constructores*. Addison-Wesley, 1992.
- [20] R. Apt Krzystof. *From logic programming to Prolog*. Prentice-Hall, 1997.