



Óscar Belmonte Fernández<sup>1</sup>,  
Carlos Granell Canut<sup>2</sup>

<sup>1</sup>Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I de Castellón; Coordinador de la sección técnica "Lenguajes de Programación" de Novática <sup>2</sup>European Commission - Joint Research Center

<oscar.belmonte@lsi.uji.es>,  
<carlos.granell@jrc.ec.europa.eu>

El lenguaje nos hace humanos, o al menos es una característica que en los humanos es más notable que en individuos de otras especies. El lenguaje permite la comunicación entre individuos de la misma especie. E incluso, y aunque de forma limitada, entre individuos de distintas especies, como por ejemplo en el caso del aprendizaje y uso de palabras inglesas por chimpancés.

Otro rasgo distintivo de nuestra especie es el uso y elaboración de herramientas. De nuevo, este rasgo no es único de nuestra especie pero sí que está notablemente más desarrollado que en el resto de especies de nuestro planeta. Y, de entre las herramientas más complejas que el hombre ha creado, los ordenadores o computadores ocupan un lugar destacado por el nivel de su complejidad.

La lengua madre de los ordenadores es el código binario, secuencias de ceros y unos ininteligibles para un humano pero que interpretadas por un ordenador le indican, de manera inequívoca, la secuencia de acciones a realizar. La brecha existente entre el lenguaje que utilizamos los humanos y el que "entienden" los ordenadores se ha estrechado con la invención de los lenguajes de programación. Estos lenguajes nos permiten a los humanos expresar, de manera razonablemente flexible y rica, una serie de instrucciones a un ordenador que, tras un proceso de codificación a lenguaje binario, los ordenadores ya son capaces de ejecutar.

Podemos pues decir que los lenguajes de programación sin lugar a dudas caracterizan la profesión de la Informática. Son en esencia la *lingua franca* para la descripción y comunicación de algoritmos para la resolución de procesos.

El origen de los lenguajes de programación se remonta a los propios inicios de la computación, lo cual da una clara idea de su importancia dentro de las ciencias de la computación. Desde entonces, son muchos los lenguajes que han visto la luz hasta nuestro días. Es cierto que muchos de ellos con poco éxito quedando incluso relegados como proyectos de investigación en laboratorios. Otros, sin embargo, han perdurado años e incluso décadas, dando lugar a derivaciones de familias enteras de lenguajes a partir de una raíz común como es el caso de Lisp o C. Este hecho claramente denota una característica funda-

# Presentación. Lenguajes de programación en perspectiva

## Editores invitados

**Oscar Belmonte Fernández** es doctor en Ciencias Físicas por la Universitat de València, profesor titular en el departamento de Lenguajes y Sistemas Informático de la Universitat Jaume I de Castellón y miembro activo en el Instituto de Nuevas Tecnologías de la Imagen (iNIT) donde desarrolla actualmente su investigación dentro del grupo de Sistemas de Información Geográfica, Geoinfo. Ha participado en distintos proyectos de investigación europeos así como dirigido proyectos en el ámbito nacional y regional. Su línea actual de investigación principal se centra en la Internet de las cosas (IoT), y las redes de sensores. Anteriormente, su principal línea de investigación se centraba en la Informática Gráfica en tiempo real. Es coordinador de la sección técnica "Lenguajes de Programación" de *Novática*.

**Carlos Granell Canut** se licenció como Ingeniero en Informática por la Universidad Jaume I de Castellón en 2000 y recibió el grado de doctor por la misma Universidad en 2006. Durante diez años ha ocupado diversos puestos como investigador en el departamento de Lenguajes y Sistemas Informáticos y en el Instituto de Nuevas Tecnologías de la Imagen (iNIT). Desde principios de 2011 se encuentra como investigador postdoctoral en la unidad *Digital Earth and Reference Data del European Commission - Joint Research Center* (EC-JRC), Ispra (Italia). Sus principales líneas de investigación tienen que ver con la aplicación de las tecnologías de información geográfica en el desarrollo de software para plataformas Web que involucran la composición y reutilización de servicios Web, modelos y servicios de geo-procesamiento. Ha participado en diversos proyectos de investigación con fondos públicos, de los cuales destacan los proyectos nacionales España Virtual y los europeos ACE-GIS, AWARE, EuroGEOSS y ENVIROFI. Además, ha realizado diversas estancias de investigación en centros europeos como SINTEF (Noruega), la *Faculty of Geo-Information Science and Earth Observation* de la Universidad de Twente (Holanda) y el *Center for Geospatial Science* de la Universidad de Nottingham (Reino Unido).

mental de los lenguajes de programación: son entidades vivas, que evolucionan, y se adaptan a las nuevas necesidades del entorno computacional.

En esta líneas de presentación no pretendemos dar una visión histórica de los lenguajes de programación, puesto que las contribuciones incluidas en la presente monografía ya dan buena cuenta de este aspecto. Queremos realizar un breve ejercicio de imaginación, con la dificultad que esto entraña, e intentar anticipar qué nos deparará el futuro en cuanto a lenguajes de programación desde el punto de visto de la evolución del entorno.

Los lenguajes de programación describen algoritmos que se ejecutan en un entorno computacional, digamos un ordenador. Este entorno ha cambiado paulatinamente desde los *mainframes*, ordenadores de escritorio, la web, la nube, hasta llegar recientemente a los dispositivos móviles.

Pero ya hay nuevos entornos computacionales a la vista (como los drones, las gafas, los coches, las redes sociales, o potencialmente cualquier objetivo urbano presente en ciudades inteligentes), que reclaman a voces la necesidad de una evolución en los lenguajes de programación, tal vez en forma de especiali-

zaciones de lenguajes existentes, o lenguajes con un mayor nivel de abstracción, o lenguajes que hibriden varios paradigmas de programación, o incluso un giro drástico hacia nuevos modelos de computación. El tiempo dirá si los lenguajes actuales serán capaces de explotar las nuevas características de estos nuevos entornos computacionales y otros tantos aún por aparecer, pero estamos seguros de que los lenguajes de programación seguirán teniendo un rol destacado en la Informática del futuro.

Con esta premisa en mente, hemos pedido a los autores de las contribuciones que aparecen en esta monografía sobre *Lenguajes de Programación* que concluyeran sus contribuciones esbozando alguna predicción futura en cada uno de sus respectivos campos. Sabemos que se trataba de una petición espinosa, y en este sentido agradecemos nuevamente a los autores su buen hacer. Esperamos pues que los lectores disfruten con la presente monografía y ésta motive suficientemente su imaginación en cuanto a los retos que nos pueden deparar los lenguajes de programación en los próximos años.

Esta monografía que ahora presentamos tiene sus limitaciones y no pretende ser exhaustiva con respecto a los lenguajes de programa-

ción, dada la cantidad, variedad y larga historia de los mismos. Pero sí hemos intentado plasmar en esencia una perspectiva histórica de la evolución de los lenguajes de programación de la mano de una serie de artículos junto con la transcripción de los discursos de la investidura como Doctor *Honoris Causa* de **Tony Hoare** en la Universidad Complutense de Madrid: La "*laudatio*" a cargo de **Ricardo Peña Marí** y el discurso de agradecimiento del propio Hoare. Creemos que siendo Tony Hoare un personaje que ha resultado tan importante en la historia y evolución de la algorítmica y la programación, los artículos citados constituyen una perfecta culminación de la presente monografía.

Sin duda, un tema tan apasionante como éste se extiende en el tiempo y esperamos que, en nuevos números de **Novática**, sigan apareciendo contribuciones alrededor de los lenguajes de programación, su formalización, características, nuevos entornos de aplicación, así como estudios sobre su uso y difusión entre los profesionales de la Informática.

Este monográfico sobre los lenguajes de programación se abre con una contribución de **Ricardo Peña Marí**. Con una claridad de exposición envidiable, Ricardo Peña, autor del libro "*De Euclides a Java : historia de los algoritmos y de los lenguajes de programación*", nos cuenta los orígenes de los lenguajes de programación de alto nivel hasta nuestros días.

El artículo parte desde el concepto fundamental de algoritmo para luego introducir, con sutileza, las analogías y diferencias entre los modelos de cómputo que principalmente determinan si un lenguaje de programación pertenece a la familia de lenguajes imperativos, lógicos y funcionales.

Sin embargo, esta clasificación tradicional por familias no se adecúa tan claramente a la situación actual debido a la aparición de nuevas características y aspectos que son transversales a dichas familias y juegan un papel clave para caracterizar lenguajes de programación que pueden pertenecer a familias distintas.

El artículo hace un recorrido sobre esos aspectos y nos permite conocer cómo funcionan y cómo influyen en los diferentes paradigmas de programación. El autor concluye con su personal visión de la evolución de los lenguajes de programación del futuro, que seguro será de gran interés para los estudiosos de la materia.

A continuación, tras familiarizarnos con las principales familias de lenguajes de programación, nos sumergimos de lleno en una de las familias de lenguajes de programación que actualmente están resurgiendo con fuerza: los lenguajes funcionales.

De la mano de **Manuel Montenegro Montes**, quien nos introduce en la historia de la programación funcional y sus principales características, y nos cuenta los actuales usos de los lenguajes de programación funcionales.

El venerado lenguaje Lisp ha sido el origen de muchos dialectos y sucesores de lenguajes funcionales actuales aunque otros tantos como Scala y F# han surgido de forma independiente. En consonancia con la tesis del primer artículo, los lenguajes funcionales actuales tienden a ser multiparadigma e incluyen aspectos que son más propios de los lenguajes imperativos, lo cual favorece que la programación funcional se atreva en el futuro en otros entornos como plataformas móviles, más allá de la programación tradicional de aplicaciones de escritorios.

No descubrimos nada nuevo al decir que el desarrollo para aplicaciones en red en entornos distribuidos, y en especial para la web, está recibiendo un creciente interés en detrimento de otros entornos "tradicionales" como son las aplicaciones de escritorio. A pesar de las dificultades para el desarrollo de aplicaciones en entornos distribuidos, las enormes ventajas que estos entornos ofrecen ha supuesto un acicate en la investigación y desarrollo de lenguajes de programación especialmente adaptados a estos entornos.

Por otro lado, existe cierto "peligro" en la explotación de nuevos terrenos inexplorados. Un peligro con el que muchos programadores tenemos que lidiar es con la falta de estandarización, cada nuevo explorador de esos terrenos vírgenes marca una senda distinta de exploración, a veces, completamente divergente del resto de sendas.

Afortunadamente, cada vez somos más conscientes del beneficio de la estandarización consensuada, ya que no merma la creatividad sino que aúna esfuerzos en la misma dirección, lo que no supone un pensamiento único, sino un esfuerzo compartido. Gracias a los esfuerzos de estandarización en el mundo de la web por parte del *World Wide Web Consortium* (W3C), hoy podemos decir que muchos de los desarrolladores de aplicaciones para la web caminan sobre una misma y amplia senda.

**Carlos Blé Jurado** nos presenta en su contribución una visión panorámica de los estándares web, desde el propio lenguaje de marcas HTML hasta el lenguaje JavaScript, base para la creación de aplicaciones ricas en los navegadores web. Carlos Blé presta especial atención a los esfuerzos de estandarización por parte del W3C. También tienen cabida en su contribución las herramientas de desarrollo fundamentales para la creación de código por parte de los programadores. Acaba su

contribución con una visión sobre el desarrollo futuro de los lenguajes de programación orientados al entorno web.

Las ciencias de la computación se construyen, como otros muchos logros humanos, a hombros de gigantes. Y de entre los gigantes que en las ciencias de la computación son y han sido, Sir Charles Antony Richard Hoare ocupa, sin duda, un lugar destacado. Desde nuestro humilde punto de vista creemos sobradamente justificado que haya sido investido Doctor *Honoris Causa* por la Universidad Complutense de Madrid.

La *Laudatio* de **Ricardo Peña Marí**, catedrático del departamento de Sistemas Informáticos y Computación de la Universidad Complutense de Madrid, es una muestra de respeto y admiración por la labor y dedicación a las Ciencias de la Computación de Hoare. Tarea en extremo complicada resumir en un discurso las muchas contribuciones de Hoare a las Ciencias de la Computación, que Ricardo Peña ha resuelto de manera muy acertada, desde nuestro punto de vista.

La respuesta a la *laudatio* de **Sir Charles Antony Richard Hoare** es un claro ejemplo del valor del aprendizaje sobre la fuente del error y el esfuerzo. Emociona la humildad del discurso de Hoare. Impresiona la búsqueda de la sencillez en las ideas subyacentes a la teoría, la belleza en la simplicidad de los conceptos.

Afortunadamente para la comunidad científica, y en particular para los investigadores en lenguajes de programación, Hoare sigue en activo, y desde mismo modo que grandes científicos anteriores a él, persigue una teoría unificadora de los lenguajes de programación con una fuerte componente algebraica.

En resumen, este monográfico trata de dar una amplia visión de la evolución de los lenguajes de programación desde sus comienzos. Cabe recordar que en el próximo número 223 de **Novática** se presentará una nueva edición de la sección "Visiones" dedicada esta vez al mundo de la programación, la cual complementará los artículos de la presente monografía con una serie de artículos breves y de opinión personal sobre la profesión de programador en nuestros días.

## Referencias útiles sobre "Lenguajes de programación"

A continuación se proporcionan algunas referencias, que complementan a las incluidas en cada uno de los artículos, y que permitirán al lector profundizar más en los distintos enfoques sobre lenguajes de programación tratados en esta monografía.

### Libros

- **Thomas J. Bergin, Richard G. Gibson (editores).** *History of programming languages*. Vol I y II. Addison-Wesley, 1996.
- **Ricardo Peña Marí.** *De Euclides a Java: historia de los algoritmos y de los lenguajes de programación*. Editorial Nivola. Colección ciencia abierta, 2006.
- **James Rumbaugh.** *El Lenguaje unificado de modelado: manual de referencia: UML*. Addison-Wesley, 2007.
- **Benjamin C. Pierce.** *Types and programming languages*. MIT Press, 2002.
- **Jason R. Briggs.** *Python for Kids: A Playful Introduction to Programming*. No Starch Press, diciembre 2012.
- **Ravi Sethi.** *Programming languages: concepts and constructs*. Addison-Wesley, 1996.

- **Brian W. Kernighan, Dennis M. Ritchie.** *El Lenguaje de programación C*. Prentice Hall Hispanoamericana, 1991.
- **Ken Arnold, James Gosling, David Holmes.** *El Lenguaje de programación Java*. Addison Wesley, 2001.
- **Bjarne Stroustrup.** *El Lenguaje de programación C++*. Addison Wesley, 2002.
- **Stephen G. Kochan.** *Programming in Objective-C 2.0*. Addison Wesley, 2009.

### Enlaces web

- **C.A.R. Hoare.** *Unified Theories of Programming*. <[http://research.microsoft.com/pubs/68626/unified\\_theories.pdf](http://research.microsoft.com/pubs/68626/unified_theories.pdf)>.
- **TIOBE Software.** *Ranking de uso de los lenguajes de programación*, <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.htm>>.
- **Scratch.** *Lenguaje de programación orientado a los niños*, <<http://scratch.mit.edu/>>.

### Congresos/eventos

- **POPL 2013.** *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. Roma (Italia) 23-25 enero 2013, <[\[sws.org/2013/\]\(http://sws.org/2013/\)>.](http://popl.mpi-</a></li></ul></div><div data-bbox=)

- **ECOOP 2013.** *European Conference on Object-Oriented Programming*, Montpellier (Francia), 1-5 julio 2013, <<http://www.ecoop.org/>>.
- **ICLP 2013 (JICSLP).** *International Conference on Logic Programming/Joint International Conference and Symposium on Logic Programming*. Estambul (Turquía) 24-29 agosto 2013, <<http://www.iclp2013.org/en/>>.
- **OOPSLA 2013.** *Conference on Object-Oriented Programming Systems, Languages, and Applications*. Indianápolis (USA) 26-31 octubre 2013, <<http://splashcon.org/2013/cfp/618>>.
- **ICFP 2013.** *The International Conference on Functional Programming*, ACM SIGPLAN. Boston, Massachusetts (USA) 25-27 septiembre 2013, <<http://icfpconference.org/icfp2013/index.html>>.

### Asociaciones

- **The Association for Logic Programming.** Asociación para el desarrollo de la Programación Lógica, <<http://www.cs.nmsu.edu/ALP/conferences/>>.

¿Estudiante de Ingeniería Técnica o Ingeniería Superior de Informática?

Puedes aprovecharte de las condiciones especiales para hacerte

socio estudiante de ATI

y gozar de los servicios que te ofrece nuestra asociación,

según el acuerdo firmado con la

Asociación RITSI

Infórmate en <[www.ati.es](http://www.ati.es)>

o ponte en contacto con la Secretaría de ATI Madrid

