

# **HISTORIA Y TIPOS DE LENGUAJE DE PROGRAMACIÓN**

Marina Aguilar  
André Ormset  
José Antonio Ríos  
René Hofboer  
Damián Zamorano  
Bernard Garcia  
Elementos de Programación  
E.T.S.I. Telecomunicación, 1º C  
Curso 03/04

# Índice

## 1.- Lenguajes de programación: una visión general:.....pág.2

- Lenguajes declarativos:.....pág.2
- Lenguajes de alto nivel:.....pág.2
- Lenguajes ensambladores:.....pág.3
- Lenguajes máquina:.....pág.3

## 2.- Perspectiva histórica de los lenguajes de programación:.....pág.3

## 3.- Diferentes lenguajes de programación a través de la historia:.....pág.4

- PASCAL:.....pág.4
- FORTRAN:.....pág.5
- COBOL:.....pág.5
- PL/I:.....pág.5
- SNOBOL:.....pág.6
- APL:.....pág.6
- LISP:.....pág.6
- PROLOG:.....pág.6
- C y C++:.....pág.7
- MODULA-2:.....pág.7
- DELPHI:.....pág.8
- JAVA:.....pág.8

### Lenguajes de programación: una visión general

Hay, al menos, dos formas fundamentales desde las que pueden verse o clasificarse los lenguajes de programación: por su nivel y por sus principales aplicaciones. Además, estas visiones están condicionadas por la evolución histórica por la que ha transcurrido el lenguaje. Además, hay cuatro niveles distintos de lenguajes de programación, como indicamos en la figura.

Lenguajes declarativos

Lenguajes de alto nivel

Lenguajes ensambladores

Lenguajes máquina

Los lenguajes declarativos son los más parecidos al castellano o inglés en su potencia expresiva y funcionalidad y están en el nivel más alto respecto a los otros. Son fundamentalmente lenguajes de órdenes, dominados por sentencias que expresan “lo que hay que hacer” en vez de “cómo hacerlo”. Ejemplos de estos lenguajes son los lenguajes estadísticos como SAS y SPSS y los lenguajes de búsqueda en bases de datos, como NATURAL e IMS. Estos lenguajes se desarrollaron con la idea de que los profesionales pudieran asimilar más rápidamente el lenguaje y usarlo en su trabajo, sin necesidad de programadores o prácticas de programación.

Los lenguajes de alto nivel son los más utilizados como lenguajes de programación. Aunque no son fundamentalmente declarativos, estos lenguajes permiten que los algoritmos se expresen en un nivel y estilo de escritura fácilmente legible y comprensible por otros programadores. Además, los lenguajes de alto nivel suelen tener la característica de “transportabilidad”. Es decir, están implementados sobre varias máquinas, de forma que un programa puede ser fácilmente “transportado” (transferido) de una máquina a otra sin una revisión sustancial. En este sentido, se llaman “independientes de la máquina”. Ejemplos de estos lenguajes de alto nivel son PASCAL, APL y FORTRAN (lenguajes de programación utilizados para aplicaciones científicas), COBOL (para aplicaciones de procesamiento de datos), SNOBOL (para aplicaciones de procesamiento de textos), LISP y PROLOG (para aplicaciones de inteligencia artificial), C y ADA (para aplicaciones de programación

de sistemas) y PL/I (para aplicaciones de propósito general). Todos estos lenguajes de alto nivel los desarrollaremos más detenidamente en otro apartado de la presentación.

Los lenguajes ensambladores y los lenguajes máquina son dependientes de la máquina. Cada tipo de máquina tiene su propio lenguaje máquina distinto y su lenguaje ensamblador asociado. El lenguaje ensamblador es simplemente una representación simbólica del lenguaje máquina asociado, lo cual permite una programación menos tediosa que con el anterior. Sin embargo, es necesario un conocimiento de la arquitectura mecánica subyacente para realizar una programación efectiva en cualquiera de estos niveles de lenguajes.

### Perspectiva histórica de los lenguajes de programación

El programador, diseñador e implementador de un lenguaje de programación deben comprender la evolución histórica para poder apreciar por qué presentan características diferentes. La misma historia nos permite ver la evolución de familias de lenguajes de programación, ver la influencia que ejercen las arquitecturas y aplicaciones de las computadoras sobre el diseño de lenguajes y evitar futuros defectos de diseño aprendiendo las lecciones del pasado. En el siguiente diagrama que presentamos resumimos algunos de los más importantes avances en el diseño de lenguajes de alto nivel desde sus comienzos hasta finales de la década de los 80. Este diagrama representa sólo una pequeña muestra de todos los lenguajes de programación. Existen en uso, hoy en día, más de 150 lenguajes de programación diferentes, con nombres tan diversos como AMBIT, BASEBALL, LOGO y MAD. Muchos son dialectos de los que se muestran en el diagrama. Los que describimos en nuestra presentación los hemos elegido debido a su mayor influencia y amplio uso entre los programadores, así como por sus distintivas características de diseño e implementación.

El diagrama también contiene las líneas que relacionan a varios lenguajes. Las líneas continuas muestran ascendencia directa, mientras que las líneas discontinuas muestran una fuerte influencia. Por ejemplo, vemos que FORTRAN I es un ascendente directo de FORTRAN II, mientras que FORTRAN, COBOL, ALGOL 60, LISP, SNOBOL y los lenguajes ensambladores, influyeron en el diseño de PL/I. Varios lenguajes representados en el diagrama están prefijados por las letras ANS. Con esto queremos decir que el American National Standards Institute ha adoptado esa versión del lenguaje como el estándar nacional.

Finalmente, creemos importante denotar algunas influencias anteriores a 1960 en el diseño de los lenguajes de programación. Incluimos como importantes la notación algebraica ordinaria, por ejemplo, que influyó fuertemente en el diseño de FORTRAN y ALGOL; el inglés como principal influencia en el desarrollo de COBOL; el lambda cálculo de Church, que dio los fundamentos de la notación funcional de LISP, mientras que el algoritmo de Markov motivó el estilo de reconocimiento de formas de SNOBOL. En último lugar, la arquitectura de computadores de von Neumann, la cual fue una evolución de la máquina más antigua de Turing, es el modelo básico de la mayoría de los diseños de computadoras hasta hoy en día. Estas

máquinas no sólo influyeron en los primeros lenguajes sino que también suministraron el esqueleto operacional sobre el que evolucionó la mayoría de la programación de sistemas.

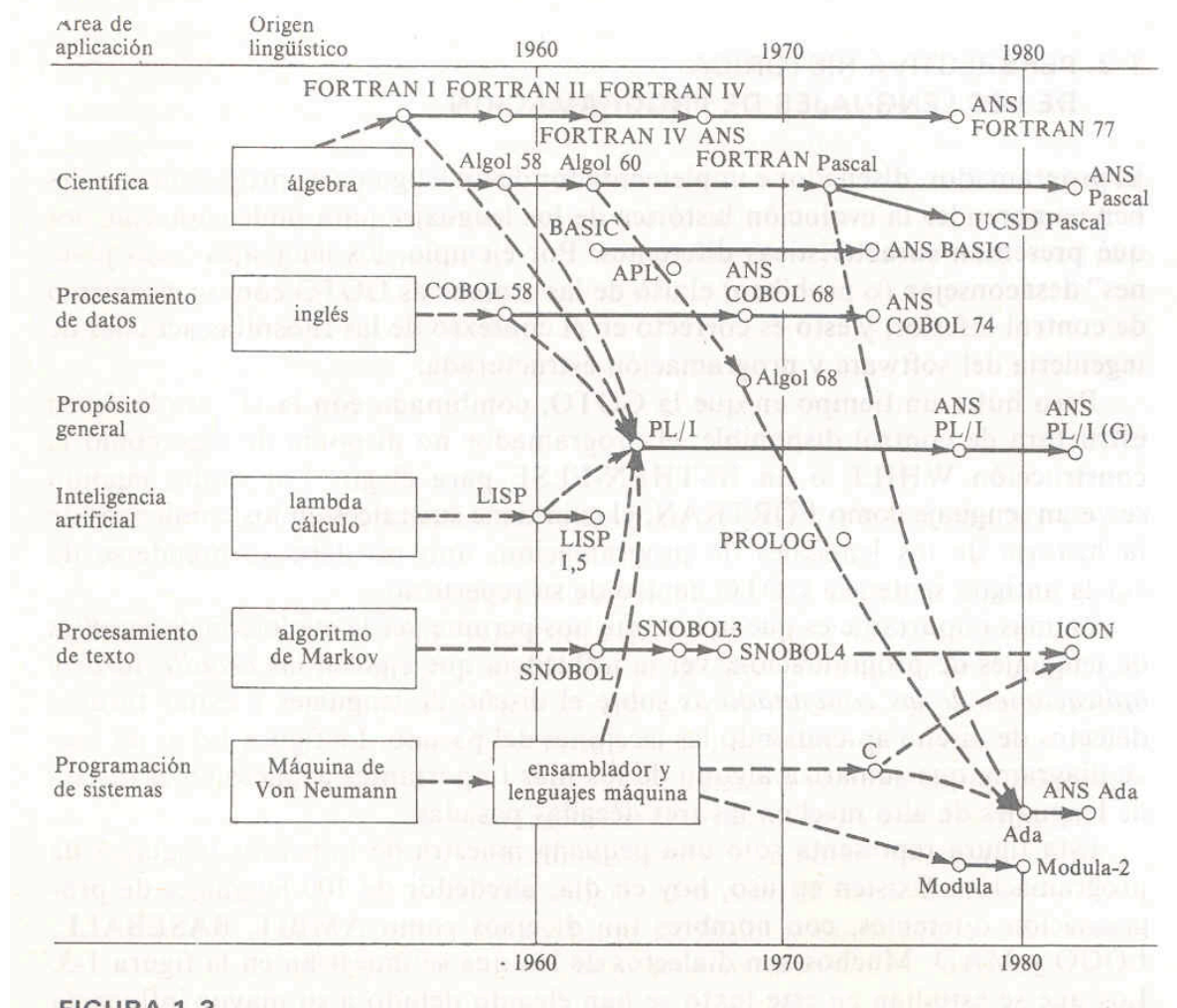


Diagrama evolutivo de los principales lenguajes de programación hasta finales de la década de los 80.

## Diferentes lenguajes de programación a través de la historia

- **PASCAL.** La historia de Pascal estaría incompleta sin trazar primero la historia del ALGOL, del cual Pascal es una evolución. La historia del ALGOL comenzó en 1.958, cuando un comité de representantes del GAMM (una organización europea de científicos en informática) y ACM (su contrapartida en USA) se reunieron en Zurcí y produjo un informe preliminar sobre un "International Algebraic Language", o IAL. Este lenguaje, conocido más tarde como ALGOL 58, atrajo mucho interés y fue implementado sobre varias computadoras. Los

representantes europeos y estadounidenses se reunieron de nuevo en París en 1.960 para considerar una versión completamente nueva de este lenguaje, conocida como ALGOL 60. Durante este período, ALGOL fue extremadamente popular entre los científicos informáticos, y su definición rigurosa marcó nuevos estándares para el diseño e implementación de lenguajes. ALGOL se convirtió en un lenguaje universal para la definición de algoritmos publicados en revistas. Con el paso del tiempo, fueron apareciendo nuevas versiones revisadas de ALGOL 60, como ALGOL W (desarrollado por Niklaus Wirth) o ALGOL 68, que fue intencionadamente un lenguaje de propósito general con aplicaciones en un amplio rango de interés, aunque rápidamente se reconoció como un lenguaje demasiado ambicioso para ser práctico. De esta manera, Wirth diseñó un sucesor más reducido del ALGOL 60 y lo llamó PASCAL. Su primer compilador se implementó en 1.970 y una versión revisada fue definida e implementada en 1.973. PASCAL fue claramente diseñado para servir como un lenguaje para enseñar diseño de algoritmos y metodología de programación. Como el ALGOL, PASCAL ha jugado un papel único como el principal lenguaje usado para publicar algoritmos en las revistas y libros. A pesar de sus fuertes mejoras sobre ALGOL, -especialmente en el área de entrada-salida, archivos, registros, gestión dinámica de memoria y estructuras de control- PASCAL también fue cuestionado por sus deficiencias, y por ello se propusieron sucesores importantes como algunos de los que describimos a continuación.

- **FORTRAN**. El miembro original de la familia, FORTRAN I, nació en 1.954, y fue implementado sobre la computadora IBM 704 en 1.956. Dos años más tarde, apareció FORTRAN II. Entre 1.958 y 1.963, FORTRAN se implementó sobre varias computadoras. FORTRAN III fue desarrollado durante este período, pero debido a que contenía demasiadas características dependientes de la máquina, nunca se implementó para uso público. En 1.962 FORTRAN IV fue desarrollado para las computadoras IBM 7090/7094. En 1.966 se estandarizaron las distintas versiones en dos únicas: la "Basic FORTRAN" y "FORTRAN". En 1.977 se añadieron extensiones y se revisaron ambas versiones, apareciendo así el FORTRAN 77, que mantenía la compatibilidad para los programas escritos antes con versiones anteriores al nuevo estándar.
- **COBOL**. (Common Business Oriented Language) A finales de 1.950, se sintió la necesidad de un lenguaje de procesamiento de datos. En mayo de 1.959, los representantes de los fabricantes de computadoras y de los usuarios de la industria y gobierno, se reunieron para formar el Comité CODASYL (Conference on Data Systems Languages), y se desarrolló una descripción de tal lenguaje, cuya revisión, en 1.960, empezó a conocerse como COBOL-60. Numerosas extensiones fueron formando nuevas versiones de este lenguaje COBOL inicial, hasta aprobarse un estándar en 1.974 por la ANS (American National Standard).
- **PL/I**. Es un lenguaje de propósito general, diseñado para soportar aplicaciones científicas, de procesamiento de datos, de procesamiento de textos y de programación de sistemas a muy alto nivel. La primera versión del PL/I fue presentada por el Comité del proyecto SHARE FORTRAN. Esta versión fue llamada NPL (New Programming Language). Fue implementado por primera vez

por IBM en 1.965 bajo el nombre de PL/I. Al principio, PL/I no fue aceptada por su ineficiencia, y no fue considerado una alternativa aceptable a los lenguajes establecidos por entonces, principalmente COBOL y FORTRAN, que hemos explicado anteriormente. A pesar de ello, en 1.975, después de haber mejorado la eficiencia de sus compiladores, fue publicado un borrador con una propuesta de un PL/I estándar, aceptado en 1.976. Alrededor de 1.981 apareció un refinamiento de PL/I estándar, llamado SUBSET/G, que preservaba la mayoría de las características útiles del PL/I, y suprimía las generalizaciones de poco uso y difíciles de implementar.

- **SNOBOL**. Fue diseñado en 1.962 por un grupo de investigación de los laboratorios Bell como una ayuda a sus propias aplicaciones sobre manipulación simbólica de fórmulas. Una versión mejorada y extendida, llamada SNOBOL3, apareció en 1.966 y fue implementada sobre distintas máquinas. Un potencial sucesor de SNOBOL4, que apareció en 1.969, se conoce como ICON. Aunque contiene muchas de las características de procesamiento de cadenas del SNOBOL, la sintaxis de ICON es mucho más parecida a la de los lenguajes convencionales como PASCAL o C, lenguaje de programación cuya evolución histórica explicaremos más adelante.
- **APL**. (A Programming Language) Fue el primero formalmente definido e introducido por Kenneth Iverson, en su libro “A Programming Language” en 1.962. Es marcadamente el producto de la genialidad de una persona con gran orientación matemática. IBM trató de desarrollar APL durante los siguientes siete años, pero nunca llegó a estandarizarse, ya que requería un conjunto único de caracteres con un hardware especial, distinto del que soporta el estándar ASCII. Como resultado, APL no ha sido implementado ampliamente sobre las distintas máquinas.
- **LISP**. (List Processor) El lenguaje LISP fue desarrollado por John McCarthy durante el periodo de 1.956 a 1.958, implementándose por primera vez entre 1.959 y 1.962. El trabajo de su autor consistía principalmente en representar información en un lenguaje formal y desarrollar un programa que razonara haciendo inferencias sobre ella. Las representaciones estaban en la forma de estructuras de listas y el lenguaje era una variante del lambda cálculo, tal y como describimos anteriormente con ayuda del diagrama de evolución. Desde 1.962 se han producido varios dialectos del LISP, incluyendo principalmente al MACLISP e INTERLISP.
- **PROLOG**. (Programming in Logic) Se diseñó principalmente para las aplicaciones de inteligencia artificial, definiendo objetos y relaciones de inferencia entre clases de objetos. Tiene unos fuertes fundamentos teóricos en el cálculo de proposiciones. Representaba una desviación tajante de las ideas tradicionales sobre comportamiento de programas, las cuales se basaban todas en las arquitecturas de máquina de von Neumann. PROLOG fue desarrollado a principios de los años 70 por Philippe Roussel. Su primer intérprete se implementó en 1.972. Desde entonces, PROLOG no cambió desde su concepción, ni se hizo ningún esfuerzo por estandarizarlo.

- C y C++**. La historia de C, y su versión ampliada, C++, la explicaremos más detenidamente, justo porque son los lenguajes de programación que nosotros utilizaremos durante el curso, además de su enorme importancia hoy en día en todos los ámbitos de la computación. C evolucionó a partir de dos lenguajes previos, BCPL y B. BCPL fue desarrollado en 1.967 por Martin Richards, como un lenguaje para escribir software y compiladores de sistemas operativos. En el lenguaje B, muchas características de BCPL fueron modeladas y se utilizó para crear versiones iniciales de lo que se llegó a denominar UNIX. Ambos lenguajes, BCPL y B eran lenguajes “sin tipo”. Cada elemento de datos ocupaba una palabra en memoria y quedaba a cargo del programador el tratar un elemento de datos como si se tratara de un número entero o de un número real. El lenguaje C fue derivado de B por Dennis Ritchie de los Laboratorios Bell, implantándose por primera vez en 1.972. C al inicio se popularizó como lenguaje de desarrollo del sistema operativo UNIX. Hoy día, virtualmente todos los sistemas están escritos en C y/o C++. C, independiente del hardware, se ha hecho disponible para la mayor parte de las computadoras a lo largo de las últimas décadas. Es posible incluso escribir programas en C que sean portátiles hacia la mayor parte de las computadoras. Hacia finales de los 70, C había evolucionado a lo que hoy se conoce como C “tradicional”. La expansión rápida de C sobre varios tipos de computadoras trajo consigo muchas variantes. Estas eran similares, pero no siempre compatibles. Esto provocaba un problema para los programadores, que necesitaban escribir códigos que pudieran funcionar con distintas máquinas. Así se vieron ante la necesidad de una versión estándar de C, que se aprobó en 1.989. El documento se conoce como ANSI / ISO 9899: 1.990. Más tarde, apareció C++, una mejoría sobre muchas de las características de C, y que proporciona capacidades de programación orientada a objetos que promete mucho para incrementar la productividad, calidad y reutilización del software. Los diseñadores de C y los responsables de sus primeras puestas en práctica nunca anticiparon que este lenguaje resultaría en un fenómeno como éste. Cuando un lenguaje de programación se torna tan arraigado como C, nuevas necesidades demandan que el lenguaje evolucione, en lugar de que sólo sea reemplazado por un nuevo lenguaje, como ya había ocurrido con muchos otros lenguajes, tal y como comentamos antes. C++ fue desarrollado por Bjarne Stroustrup en los Laboratorios Bell y originalmente fue llamado C “con clases”. El nombre C++ incluye el operador de incremento (++) de C, para indicar que C++ es una versión mejorada de C. C++ es un conjunto de C, por lo que, para compilar los programas existentes de C, los programadores pueden utilizar un compilador C++ y posteriormente modificar estos programas a C++. A mediados de los años noventa, la mayor parte de los entornos de programación C se convirtieron a C++.
- MODULA-2**. Podemos considerar, con toda exactitud, al lenguaje MODULA-2 como un descendiente directo y mejorado de su antecesor PASCAL. MODULA-“, al igual que su ancestro PASCAL, ha sido desarrollado por el profesor Niklaus Wirth, del Instituto Tecnológico de Zurich, como dijimos anteriormente. MODULA-2 nació, de una forma experimental, en 1.975 cuando se hizo necesario disponer de un lenguaje de programación de propósitos generales y que, además,



fuera lo suficientemente operativo y cualificado para abordar problemas de multiprogramación, diseño de sistemas, software de base, traductores y compiladores, etc. Como resultado, se realiza la primera implementación de MODULA-2 al proyecto Lilith del ETH de Zurich, con el objetivo de diseñar un sistema completo de hardware y software, y para el que era necesario disponer de un lenguaje de desarrollo sencillo y que pudiera abordar, conjuntamente, problemas resolubles con un lenguaje de alto nivel, y necesidades de programación a bajo nivel o cercana a la estructura de la máquina.

- **DELPHI**. En el año 1995 se crea el nuevo sucesor de Pascal, al que se llamó Delphi, siendo la primera herramienta con un entorno de desarrollo visual construida por Borland. Esta caracterizado por ser un lenguaje orientado a eventos, es decir, que la ejecución del programa no es secuencial, sino que depende de los eventos que suceden durante la ejecución de la aplicación. Delphi es una herramienta de Desarrollo Rápido de Aplicaciones (RAD). Los componentes que incorpora facilitan el acceso a bases de datos, comunicación a través de Internet, calidad en impresiones, desarrollo de aplicaciones multimedia, enlaces DDE, componentes OLE y VBX, etc. Borland ha introducido al mercado varias versiones de Delphi, aportando mejoras notables, entre las que cabe destacar el CodeInsight, un asistente que muestra automáticamente las listas de parámetros de procedimientos, métodos y eventos. En el año 2001 Borland lanzo al mercado la versión de Delphi 6.0 que funciona bajo Windows y es compatible con todas las versiones anteriores. Junto con esta versión se introdujo en el mercado la primera versión Kylix, una versión de Delphi que funciona bajo Linux. La ultima versión disponible en el mercado es Delphi 7.0. Entre las nuevas características se incluye un nuevo compilador que permite construir aplicaciones basadas en la plataforma .NET
- **JAVA**. En 1991 un grupo de ingenieros de Sun Microsystems liderados por Patrick Naughton y James Gosling comienza el desarrollo de un lenguaje destinado a generar programas independientes de la plataforma en la que se ejecutan. Su objetivo inicial nada tiene que ver con lo que hoy en día es Java, sus creadores buscaban un lenguaje para programar los controladores utilizados en la electrónica de consumo. Existen infinidad de tipos de CPU distintas, y generar código para cada una de ellas requiere un compilador especial y el desarrollo de compiladores sabemos que es caro. Después de dieciocho meses de desarrollo aparece la primera versión de un lenguaje llamado OAK que más tarde cambiaría de nombre para convertirse en Java. La versión de 1992 está ampliada, cambiada y madurada, y a principios de 1996 sale a la luz la primera versión de Java. Los inicios son difíciles, no se encuentran los apoyos necesarios en Sun y el primer producto que sale del proyecto, un mando a distancia muy poderoso y avanzado, no encuentra comprador. Pero el rumbo de Java cambiaría debido a una tecnología completamente ajena a los controladores de electrodomésticos: Internet.

Mientras Java se estaba desarrollando, el mundo de las comunicaciones crecía a una velocidad de vértigo, Internet y principalmente el mundo World Wide Web dejaban los laboratorios de las universidades y llegaban a todos los rincones del planeta. Se iniciaba una nueva era y Java tuvo la suerte de estar allí y aprovechar la oportunidad. En 1993 con el fenómeno Internet en marcha, los desarrolladores de Java dan un giro en su desarrollo al darse cuenta de que el problema de la portabilidad de código de los controladores es el mismo que se produce en Internet, una red heterogénea y que crece sin parar, y dirigen sus esfuerzos hacia allí. En 1995 se libera una versión de HotJava, un navegador escrito totalmente en Java y es en ese mismo año cuando se produce el anuncio por parte de Netscape de que su navegador sería compatible con Java. Desde ahí otras grandes empresas se unen y Java se expande rápidamente. No obstante, las primeras versiones de Java fueron incompletas, lentas y con errores. Han tenido que pasar varios años de desarrollo y trabajo para que Java sea un lenguaje perfectamente asentado y lleno de posibilidades. Actualmente es ampliamente utilizado en entornos tanto relacionados con Internet como completamente ajenos a la Red. El mundo Java está en constante desarrollo, las nuevas tecnologías surgen y se desarrollan a gran velocidad haciendo de Java un lenguaje cada día mejor y que cubre prácticamente todas las áreas de la computación y comunicaciones, desde teléfonos móviles hasta servidores de aplicaciones.

## Bibliografía

- “Lenguajes de programación”  
Allen B. Tucker,  
Editorial McGraw Hill
  
- “Modulo 2. Desarrollo de software”  
Carlos Galan,  
Editorial Paraninfo.
  
- “Cómo programar en C y C++”  
H. M. Deitel y P. J. Deitel  
Editorial Prentice Hall
  
- “Programación, algoritmos y ejercicios resueltos en JAVA”  
David Camacho y José María Valls  
Editorial Prentice Hall

