

Programación

M. Jesús Marco Galindo
Jordi Àlvarez Canal
Josep Vilaplana Pastó (coordinadores)

PID_00149875

Material docente de la UOC



Universitat Oberta
de Catalunya

www.uoc.edu

**M. Jesús Marco Galindo**

Licenciada en Informática y máster en Dirección y Administración de empresas por la Universidad Politécnica de Cataluña (UPC). Diploma de Estudios Avanzados dentro del programa de doctorado en la Sociedad de la Información y el Conocimiento de la UOC. Ha ejercido como profesional de la informática dentro del área de banca y ha sido profesora del Departamento de Lenguajes y Sistemas Informáticos de la Universidad Politécnica de Cataluña. Actualmente es profesora de los Estudios de Informática y Multimedia de la UOC.

**Jordi Àlvarez Canal**

Licenciado en Informática por la Universidad Politécnica de Cataluña (UPC). Ha ejercido como profesor del Departamento de Lenguajes y Sistemas Informáticos de la UPC asignado a la Facultad de Informática de Barcelona, la Facultad de Matemáticas y Estadística, y la Escuela Técnica Superior de Ingenieros Industriales de Barcelona, impartiendo asignaturas de programación. Consultor de la Universitat Oberta de Catalunya. Actualmente trabaja en GFT-Iberia en el ámbito del desarrollo en software, concretamente utilizando tecnologías C y Java.

**Josep Vilaplana Pastó**

Ingeniero de Telecomunicaciones y doctor en Informática. Profesor del Departamento de Lenguajes Informáticos de la Universidad Politécnica de Cataluña (UPC). Adscrito a la Facultad de Informática de Barcelona, la Escuela Superior de Ingeniería Industrial de Barcelona y la Escuela Universitaria de Ingeniería Técnica Industrial de Terrassa, donde imparte asignaturas de programación, gráficos con ordenadores y programación concurrente.

**Xavier Burgués i Illa**

Licenciado en Informática, profesor del Departamento de Lenguajes y Sistemas Informáticos de la Universidad Politécnica de Cataluña (UPC). Adscrito a la Facultad de Informática de Barcelona, donde imparte asignaturas de programación. Consultor de la Universitat Oberta de Catalunya.

**Ramon Vilanova Arbós**

Doctor en Informática y profesor del Departamento de Informática de la Universidad Autónoma de Barcelona (UAB) y catedrático de Telecomunicación y Ingeniería de Sistemas en la Escuela Universitaria de Informática de Sabadell (UAB). Consultor de la Universitat Oberta de Catalunya.

**Raymond Lagonigro Bertran**

Licenciado en Informática por la Universidad Politécnica de Cataluña. Profesor del Departamento de Lenguajes y Sistemas Informáticos y del Departamento de Matemática e Informática de la Universidad de Vic. Director del Servicio de Informática de la Universidad de Vic. Consultor de la Universitat Oberta de Catalunya.

Primera edición: febrero 2010

© Jordi Àlvarez Canal, Xavier Burgués i Illa, Raymond Lagonigro Bertran, M. Jesús Marco Galindo, Ramon Vilanova Arbós, Josep Vilaplana Pastó.

Todos los derechos reservados

© de esta edición, FUOC, 2010

Av. Tibidabo, 39-43, 08035 Barcelona

Diseño: Manel Andreu

Realización editorial: Eureka Media, SL

Depósito legal: B-3.154-2010

ISBN: 978-84-692-9433-8

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Introducción

En el ámbito de las tecnologías de la información y la comunicación (TIC) una de las tareas más importantes es el desarrollo de programas de ordenador. Este curso os debe proporcionar las bases que os introducirán en este apasionante campo de la programación.

Como veréis más adelante, podemos definir el término algoritmo como un procedimiento de cálculo que consiste en seguir un conjunto ordenado y finito de instrucciones que conduce a la solución de un problema. Lo que este curso pretende es enseñaros a construir algoritmos para resolver problemas concretos.

Esto no es una tarea nada fácil. Requiere la asimilación de una serie de conceptos y técnicas que, a pesar de no ser demasiado numerosas, sí que necesitan un esfuerzo inicial de comprensión y una práctica constante.

En muchas ocasiones, la programación es considerada una especie de arte. De este modo, aprender a programar consiste en aprender la sintaxis y la semántica de un lenguaje de programación, y después, a partir de la experiencia, aprender a ir generando programas cada vez más complejos. En cambio, en este curso, igual que en otros, vemos la programación y la algorítmica como técnicas, y las intentamos sistematizar al máximo. Esto nos debe permitir incrementar nuestra productividad, y que seamos capaces de generar algoritmos de una cierta complejidad mucho antes de lo que podríamos sólo con nuestra experiencia.

Por otro lado, si en vuestra vida profesional tenéis que participar en el desarrollo de proyectos de grandes dimensiones, haréis vuestro trabajo de forma mucho más eficiente si habéis asimilado correctamente lo que se explica en este curso.

Por todo ello, y también porque en cualquier caso aprender algorítmica no es un trabajo fácil, es preciso que sigáis al pie de la letra la metodología presentada en este curso.

Durante el curso utilizaremos para describir algoritmos un lenguaje que denominaremos lenguaje algorítmico. A diferencia del lenguaje hablado, se trata de un lenguaje muy rígido con unas pocas construcciones que nos servirán para expresar lo que queremos que el ordenador haga, como si se tratase de una serie de instrucciones que le damos a alguien para que haga un trabajo.

Esta forma de programar (mediante la combinación de una serie de instrucciones) se denomina programación imperativa, en el sentido de que damos una serie de instrucciones al ordenador. Existen otros tipos de programación, como

por ejemplo la lógica y la funcional. Además, también hay diferentes metodologías que se pueden aplicar a algún tipo de programación, como por ejemplo la orientación a objetos. En este curso, sin embargo, nos dedicaremos sólo a estudiar los fundamentos de la programación imperativa con su metodología básica.

El lenguaje algorítmico no es entendido directamente por un ordenador. Antes de poder ejecutar nuestro algoritmo en un ordenador, tendremos que traducirlo a un lenguaje de programación imperativo: Pascal, C, Java o cualquier otro. Una vez disponemos del algoritmo, esta traducción es muy directa y no debe suponer ningún problema. Simplemente, se trata de conocer bien el lenguaje de programación que utilizaremos y el *software* informático, como por ejemplo el compilador, que nos servirá para generar el programa que el ordenador debe ejecutar.

Un compilador es la aplicación informática que traduce un fichero de texto escrito (con las normas del lenguaje de programación correspondiente) a un programa ejecutable que contiene una serie de instrucciones en el lenguaje propio del ordenador (denominado lenguaje máquina). Así pues, en primer lugar deberemos crear el fichero de texto con el programa en el lenguaje de programación elegido (codificación), y posteriormente, ejecutar el compilador para obtener el programa ejecutable. Todo este proceso recibe el nombre de implementación.

El uso del lenguaje algorítmico como paso previo al del lenguaje de programación puede parecer innecesario de entrada, pero tiene la importante ventaja de desvincular el desarrollo de un algoritmo del lenguaje de programación utilizado por su codificación en el ordenador. Esto nos permite utilizar cualquier lenguaje de programación imperativo sin demasiados problemas.

Por lo tanto, una vez diseñado el algoritmo, podríais traducirlo a cualquier otro lenguaje de programación imperativo con bastante facilidad. Esta parte de traducción al lenguaje de programación y generación del programa que podremos ejecutar en el ordenador no se encuentra en estos apuntes, sino que la encontraréis directamente en la web de la asignatura, en el campus virtual.

El curso está estructurado en cinco módulos, de forma que, a medida que avanza, podemos ir generando algoritmos de una mayor complejidad. El primer módulo contiene una visión general de lo que es la programación, ya que introduce una serie de conceptos que utilizaremos a lo largo del curso, y pone la mirada en lo que debe ser a la larga nuestro objetivo real: ser capaces de desarrollar aplicaciones informáticas de una medida considerable (a pesar de que en esta asignatura sólo aprenderemos a diseñar algoritmos de una complejidad baja o media).

El módulo “Introducción a la algorítmica” introduce el lenguaje que usaremos para describir algoritmos (denominado lenguaje algorítmico), así como la noción de especificación de un algoritmo. La especificación nos servirá para saber qué hace un algoritmo de una forma clara y precisa, y nos acompañará a lo largo de todo el curso.

En el módulo “Tratamiento secuencial” se introduce la técnica básica que nos permitirá sistematizar al máximo la elaboración de algoritmos. Esta técnica se fundamenta en la aplicación de unos esquemas (algoritmos muy generales, una especie de plantillas) que nos permitirán resolver un amplio abanico de posibilidades diferentes.

Los algoritmos que desarrollaremos utilizarán una serie de datos. El módulo “Introducción a la algorítmica” introduce el tipo básico de datos con los que podremos trabajar. Sin embargo, no tendremos suficiente con esto. Para algoritmos de una cierta complejidad nos puede resultar bastante útil combinar varios de estos datos en estructuras más complejas. Éste es el tema del módulo “Tipos estructurados de datos”.

Con estos cuatro módulos podemos resolver problemas bastante sencillos. En el módulo “Introducción a la metodología de diseño descendente” se introduce una técnica que nos debe permitir utilizar todo el contenido de los módulos anteriores para abordar problemas de una mayor complejidad; se trata de la técnica de diseño descendente.

Objetivos

Con los materiales asociados a esta asignatura, los estudiantes alcanzaréis los objetivos siguientes:

1. Diseñar algoritmos para resolver problemas de complejidad media.
2. Entender y saber especificar un problema correspondiente a un enunciado determinado.
3. Conocer bien el lenguaje algorítmico que se utiliza en este curso, como plataforma para expresar de una forma rigurosa los algoritmos.
4. Conocer y utilizar correctamente las técnicas de diseño algorítmicas presentadas en este curso: uso de esquemas de tratamiento secuencial y la metodología de diseño descendente.
5. Decidir qué estructuras de datos (de las presentadas en este curso) nos hacen falta para resolver un problema de forma eficiente, y saber cómo utilizarlas.
6. Implementar los algoritmos en un lenguaje imperativo para obtener un programa ejecutable en el ordenador.
7. Tener habilidad utilizando el *software* necesario (editor y compilador) para editar, compilar y ejecutar los programas.

Contenidos

Módulo didáctico 1

Introducción a la programación

Maria Jesús Marco Galindo, Josep Vilaplana i Pastó

1. Conceptos básicos de programación
2. La programación como disciplina de ingeniería
3. Objetivos de la asignatura

Módulo didáctico 2

Introducción a la algorítmica

Jordi Álvarez i Canal, Josep Vilaplana i Pastó

1. Objetos elementales del lenguaje algorítmico
2. Especificación de algoritmos
3. Estructuras algorítmicas
4. Acciones y funciones

Módulo didáctico 3

Tratamiento secuencial

Jordi Álvarez i Canal, Xavier Burgués i Illa

1. Algoritmos y secuencias
2. Esquema de recorrido de una secuencia
3. Esquema de búsqueda en una secuencia
4. Esquemas aplicados a secuencias de entrada/salida
5. Combinación de esquemas

Módulo didáctico 4

Tipos estructurados de datos

Raymond Lagonigro i Bertran, Ramon Vilanova i Arbós

1. Introducción y motivación. Estructuración de datos
2. Tablas
3. Tuplas

Módulo didáctico 5

Introducción a la metodología de diseño descendente

Josep Vilaplana i Pastó, Maria Jesús Marco Galindo

1. Diseño descendente

Bibliografía

Botella, P.; Bofill, M.; Burgués, X.; Franch, X.; Lagonigro, R; Vancells, J. (1998). *Fundamentos de programación I*. Barcelona: Ediuoc.

Burgués, X.; Franch, X. (1998). *Apunts de Fonaments de Programació (I), mòdul 2*. Barcelona: Universitat Oberta de Catalunya.

Brassard, G.; Bratley, P. *Algorítmica, concepción y análisis*. Manuales de informática Masson. Barcelona: Masson.

Castro, J.; Cucker, F.; Messeguer, X.; Rubio, A.; Solano, L.; Vallés, B. (1992). *Curs de programació*. Madrid, etc.: McGraw-Hill.

Peña Mari, R. (1998). *Diseño de programas. Formalismo y Abstracción* (2.^a edición). Prentice Hall.

Scholl, P.C.; Peurin, J.P. (1991). *Esquemas algorítmicos fundamentales. Secuencias e iteración*. Manuales de informática Masson. Barcelona: Masson.

Vancells, J.; López E. (1992). *Programació: introducció a l'algorísmica*. Vic: Eumo Editorial (Tecno-Ciència).

Wirth, N. (1986) *Algorithms and Data Structures*. Prentice Hall.