

ESTRUCTURAS DE DATOS



Y

ALGORITMOS

GUÍA ILUSTRADA
PARA PROGRAMADORES

ANAYA
MULTIMEDIA

**MARIONA
NADAL FARRÉ**

ESTRUCTURAS DE DATOS Y ALGORITMOS

ESTRUCTURAS DE DATOS Y ALGORITMOS

GUÍA ILUSTRADA PARA PROGRAMADORES

MARIONA NADAL FARRÉ



Títulos Especiales

Diseño de cubierta: Celia Antón Santos

Ilustración de cubierta: Eduardo Rodríguez Meliá

Revisión: Gelsys M. García Lorenzo

Maquetación: Claudia Valdés-Miranda Cros

Responsable editorial: Eugenio Tuya Feijoó

1.^a edición digital: 2022

Todas las denominaciones de programas, sistemas operativos, equipos, etc., que aparecen en este libro son marcas registradas de sus respectivas compañías y organizaciones.

Reservados todos los derechos. El contenido de esta obra está protegido por la ley, que establece penas de prisión y/o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaran, distribuyeren o comunicaren públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.

Edición española:

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2022
Calle de Juan Ignacio Luca de Tena, 15
28027 Madrid
www.anayamultimedia.es

ISBN: 978-84-415-4639-4

Edición electrónica sobre la 1.^a edición impresa

A mi primera promoción¹ de GDAM en la escuela CES, juntos hemos probado en vivo el funcionamiento del libro de Java, mientras dabais vuestros primeros pasos en el mundo de la programación. También me habéis acompañado en el crecimiento de este libro, el de algoritmos, y me habéis inspirado para futuras obras.

¹ Grupo de primero, turno de tarde, curso 2021-2022.

Agradecimientos

A toda la gente mencionada en los agradecimientos del libro anterior y además a todos aquellos que me han ayudado a dar un paso más en mi carrera como profesora, dejando atrás mis tiempos como desarrolladora en la empresa privada:

- A Carlos, de LinkedIn Learning, por seguir confiando en mí para elaborar más y más videocursos.
- A Eugenio, de Anaya Multimedia, por ofrecerme hacer este segundo libro y los que vendrán.
- A Elena, Carles y Carlos, de la UOC, por abrirme la puerta a la docencia universitaria.
- A Sonia, de la escuela CES, por creer en mis posibilidades como profesora presencial y darme no solo una sino dos oportunidades (también en la escuela CEV) y así permitirme cambiar de vida.

Y específicamente sobre esta obra:

- A Pablo, por releer todos los capítulos, encontrar esos errores que a veces se escapan y escribir el prólogo.
- A Eduardo, por crear a Eday y por el resto de ilustraciones.
- A Gelsys, Claudia y Celia, por sus revisiones, maquetaciones y portada.
- A resto del equipo de Anaya Multimedia, y a todo el personal involucrado en la impresión, distribución, exposición, reposición, venta, recomendación, reparto... sin ellos no llegaríamos a las librerías ni a tus manos.

Sobre la autora

Mariona Nadal es ingeniera informática por la Universidad Politécnica de Madrid (2003).

Cuenta con más de 20 años de experiencia en la práctica del desarrollo de aplicaciones Java en entornos empresariales para grandes clientes internacionales, experiencia que le ha permitido ser formadora en J2EE de jóvenes ingenieros en su primer empleo como programadores.

Actualmente es profesora de programación en grados superiores de Formación Profesional. También es instructora de LinkedIn Learning, donde cuenta con un número creciente de cursos sobre Fundamentos de la Programación y Java. Su primer libro fue el *Manual Imprescindible. Curso de Programación Java*, publicado en junio de 2021.

Su metodología fresca, directa y realista hace sus cursos amenos, claros y útiles, llevándote a través de la práctica a un estilo de programación de fácil mantenimiento y alta empleabilidad.

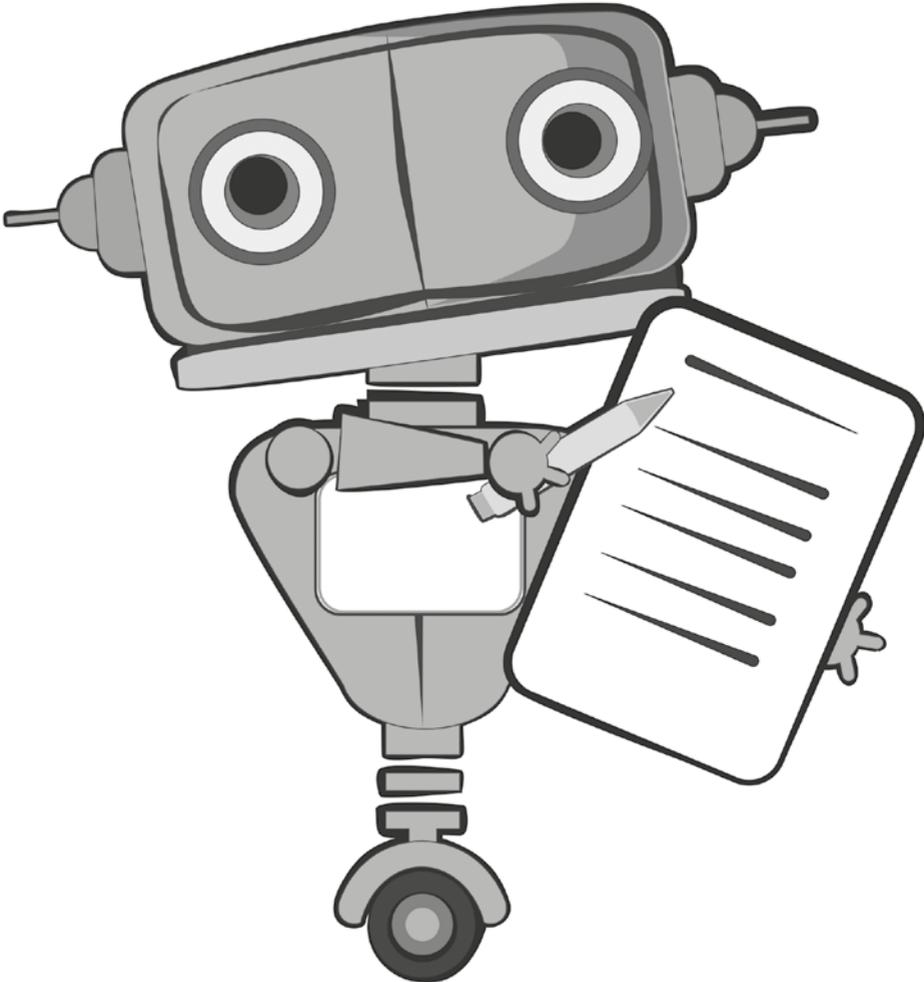
En <http://school.dsrroma.es> encontrarás todos sus cursos y más información.



Web DSR School
bit.ly/dsr-school



Cursos en LinkedIn Learning
bit.ly/lilmnadal



Índice de contenidos

	Cómo usar este libro	12
	Prólogo	15
	Parte 1. Estructuras de datos	
1	Introducción a las estructuras de datos	18
	Estructuras de datos y tipos abstractos de datos	19
	Introducción a la complejidad	20
	Análisis y elección de una estructura de datos	22
	Test y ejercicios	24
	Soluciones	25
2	Tipos abstractos de datos	26
	Tipos básicos	27
	Tuplas o registros	29
	Listas	31
	Conjuntos	33
	Pilas y colas	35
	Diccionarios	37
	Árboles	39
	Grafos	42
	Iteradores	45
	Test y ejercicios	46
	Soluciones	48
3	Estructuras de datos	50
	Introducción	51
	Listas estáticas	54
	Listas dinámicas	57
	Operaciones sobre listas	61
	Tablas hash	65
	Árboles	69
	Grafos	71
	Test y ejercicios	74
	Soluciones	76

4	Colecciones en Java	78
	Introducción	79
	ArrayList	80
	HashSet	90
	HashMap	99
	TreeSet	113
	Test y ejercicios	126
	Soluciones	126
	Parte 2. Algoritmos	
5	Introducción a los algoritmos	130
	Introducción	131
	Características de un algoritmo	132
	Metodología de creación de algoritmos	133
	Lenguajes para la definición de algoritmos	135
	Estructuras básicas en los algoritmos	136
	Test y ejercicios	138
	Soluciones	139
6	Caja de herramientas	141
	Introducción	142
	Variables y constantes	142
	Bifurcaciones	143
	Iteraciones	144
	Funciones y métodos	146
	Recursividad	148
	Complejidad	149
	Test y ejercicios	150
	Soluciones	151
7	Algoritmos muy simples	153
	Introducción	154
	Algoritmo: Mínimo de tres números	154
	Algoritmo: Sucesión de Fibonacci	162
	Test y ejercicios	166
	Soluciones	167

8	Algoritmos de ordenación	170
	Introducción	171
	Algoritmo: Ordenación por inserción (insertion sort)	171
	Algoritmo: Ordenación de burbuja (bubble sort)	175
	Algoritmo: Ordenación por mezcla (merge sort)	178
	Algoritmo: Ordenación rápida (quicksort)	189
	Test y ejercicios	197
	Soluciones	198
9	Algoritmos de búsqueda	199
	Introducción	200
	Algoritmo: Búsqueda lineal o secuencial	201
	Algoritmo: Búsqueda binaria o dicotómica	203
	Test y ejercicios	210
	Soluciones	210
10	Algoritmos de texto	212
	Introducción	213
	Algoritmo de cifrado César	213
	Algoritmo de búsqueda de cadenas Boyer-Moore	218
	Algoritmo de compresión Huffman	227
	Test y ejercicios	248
	Soluciones	249
11	Algoritmos matemáticos	252
	Introducción	253
	Algoritmo de Euclides (máximo común divisor)	253
	Criba de Eratóstenes (números primos)	257
	Algoritmo de Karatsuba (multiplicación de números grandes)	263
	Test y ejercicios	267
	Soluciones	268
	Índice alfabético	269

Cómo usar este libro

A quién va dirigido y qué es necesario para empezar

Este libro se dirige a estudiantes de programación y a curiosos de la informática que quieren afianzar sus conocimientos de programación y consolidar dos piezas básicas de la misma: las estructuras de datos y los algoritmos.

Poco importa qué lenguaje o lenguajes conozcan o quieran aprender, aunque los ejemplos de código incluidos en este libro son básicamente en Java, JavaScript o Python, o incluso alguno en Haskell, los conocimientos adquiridos son aplicables a cualquier otro lenguaje, siempre acompañados de pseudocódigo y diagrama de flujo independientes del lenguaje.

Para seguir los ejemplos y los ejercicios del libro necesitarás preferiblemente un ordenador con:

- Un IDE (entorno de desarrollo integrado) adecuado para el lenguaje elegido, que es un editor de texto adaptado para programadores que integra muchas funcionalidades útiles, como la de compilar y ejecutar el código. No es imprescindible, pero, según van creciendo los proyectos, se agradece tenerlo. En este libro, para Java, he utilizado Eclipse IDE, pero, si prefieres otro, también te sirve. Puedes descargar Eclipse en eclipse.org.
- Acceso a internet y navegador web para utilizar compiladores online de los distintos lenguajes. Yo he utilizado:
 - Para Python: <https://www.programiz.com/python-programming/online-compiler/>.
 - Para JS: <https://js.do>.
 - Para Haskell: <https://paiza.io>.
 - Para Blockly: <https://developers.google.com/blockly>.

Estructura del libro

Esta obra se divide en dos partes: la primera dedicada a las estructuras de datos y la segunda a los algoritmos. Aunque puedes realizar una lectura lineal del libro, probablemente su uso más habitual sea ir directamente a la estructura de datos o al algoritmo que te interese en cada momento.

En la parte 1, empezamos en el primer capítulo con una introducción a las estructuras de datos, para profundizar en el capítulo 2 en los tipos abstractos de datos y en el 3 en las estructuras de datos que los implementan. El cuarto capítulo se dedica a las colecciones de Java, algunos conceptos serán aplicables a otros lenguajes y otros no.

La segunda parte también empieza con una introducción, en este caso a los algoritmos. El capítulo 6, la caja de herramientas, te permitirá conocer las bases para trabajar con algoritmos. Los capítulos siguientes están dedicados cada uno a un tipo de algoritmos, empezando por unos muy simples, luego los de ordenación, después los de búsqueda, para terminar con los algoritmos sobre textos y los matemáticos.

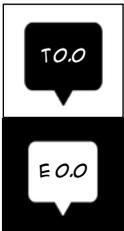
Convenios utilizados en este libro

Para facilitar la comprensión de este manual se han utilizado algunos formatos especiales:

- Los nombres de comandos, menús, opciones, cuadros de diálogo y otros elementos aparecen en letra «de palo seco» para distinguirlos fácilmente del resto del texto, por ejemplo, la instrucción `System.out.println`.
- Las combinaciones de teclas aparecen separadas por un guion y también en un tipo de letra diferente, por ejemplo, `Ctrl-Mayús-O`.
- Los fragmentos de código los verás como textos de ancho fijo, al igual que los resultados de la ejecución de los programas. El color del texto te dará la pista para saber de qué tipo se trata:

```
fragmento de código
resultados de ejecución
```

- Los test y ejercicios se encuentran al final del capítulo, pero en el momento en el que ya sabes lo imprescindible para resolverlos, encontrarás estos iconos que referencian el número de test (blanco sobre negro) o de ejercicio (negro sobre blanco) que ya puedes intentar.
- A lo largo del libro aparecen notas informativas separadas del texto principal que proporcionan información, tales como aclaraciones, advertencias, curiosidades o consejos:



Nota: Para facilitar o concretar información relacionada con el tema abordado, incluyen recomendaciones que conviene tener en cuenta.

Resumen

Breve resumen para afianzar el concepto recién explicado.

Información de soporte

Los ejemplos, ejercicios y proyectos desarrollados en este libro están disponibles en un repositorio git de acceso público: <https://github.com/DSRschool/Anaya-Eday>.



Repositorio en git

bit.ly/gitEday

Puedes descargarlo todo en un fichero zip, pero te recomiendo aprender las operaciones básicas de git para sacarle el máximo provecho.

Algunas nociones para empezar a utilizar el repositorio:

- La interfaz de gitHub puede cambiar sin previo aviso, pero además de ver el código en la propia web, mediante el botón Code es posible clonarlo para tener tu propia versión del repositorio o descargártelo como zip.
- Viendo un fichero en la web, el botón History te permitirá conocer todos los *commits* hechos sobre ese fichero. Cuando cojas una versión en concreto, verás con el fondo blanco las líneas de código sin cambios respecto a las anteriores, con fondo rojo y un signo menos, las líneas que estaban antes y ya no están, y con fondo verde y un signo más, las nuevas. Si una línea ha sido modificada, aparecerá la versión antigua en rojo y la nueva en verde.

Recomendaciones y buenas prácticas

Durante toda la obra te llevo de la mano para ir aprendiendo nuevos conceptos, te propongo ejemplos y ejercicios para afianzar esos conocimientos, pero ¡hay que ser valiente! ¡Suéltate de mi mano y avanza más! Si te muestro un ejemplo que calcula el máximo, intenta por tu cuenta calcular el mínimo. Si durante el ejercicio se te ocurren mejoras o retos, ¡hazlos! El espacio en el libro es limitado, tu capacidad de aprendizaje es infinita.

Puede parecer extraño, pero conviene que tengas papel y lápiz a mano, poder tomar apuntes, hacer esquemas y diagramas ayuda, y mucho, a la hora de programar. Y es una herramienta casi imprescindible para probar los algoritmos.

Tienes todos los ficheros disponibles para descargar, pero si de verdad quieres aprender a programar, escríbelos tú. Como los cuadernos de caligrafía de la escuela, escribir letra a letra los programas te llevará a asimilar lo que estás escribiendo. Olvídate de copiar y pegar hasta que no sepas perfectamente qué estás copiando y pegando.

Para generar buenos programas, debemos ordenar nuestra mente, pero también nuestros ficheros. Clasificalos bien en carpetas, da nombres significativos tanto a las carpetas como a los ficheros, de forma que te resulte fácil recuperar lo que estés buscando.

Contacta conmigo

Tus sugerencias, comentarios, correcciones, propuestas... son una fuente de crecimiento para mí y mis contenidos. Puedes escribirme a school@dsrroma.es. Procuraré contestar tan pronto como las circunstancias tengan a bien. También puedes seguirme en Twitter (@MarionaJava) o en LinkedIn (marionanadal) para estar al día de mis nuevas publicaciones.

Prólogo

Quien se adentra en el desarrollo de software necesita un sólido conocimiento de los bloques básicos que sirven de fundamento a esta profesión. Estos bloques son las estructuras de datos que son objeto de manipulación en los programas que creamos y los algoritmos mediante los cuales se produce dicha manipulación. Un conocimiento profundo de ambos es más importante que las particularidades de uno u otro lenguaje de programación, puesto que estos son solo la representación de aquellos en una forma concreta.

En esta obra, Mariona presenta de una forma a la vez amena y rigurosa los principales algoritmos y estructuras de datos que son imprescindibles para un profesional de la informática o un estudiante que aspire a serlo algún día y que deben manejar con soltura. Mariona, gracias a su experiencia en el desarrollo de software, tiene una profunda comprensión de estas materias y, gracias a sus años de dedicación a la enseñanza, es capaz de transmitirla de una manera muy eficaz.

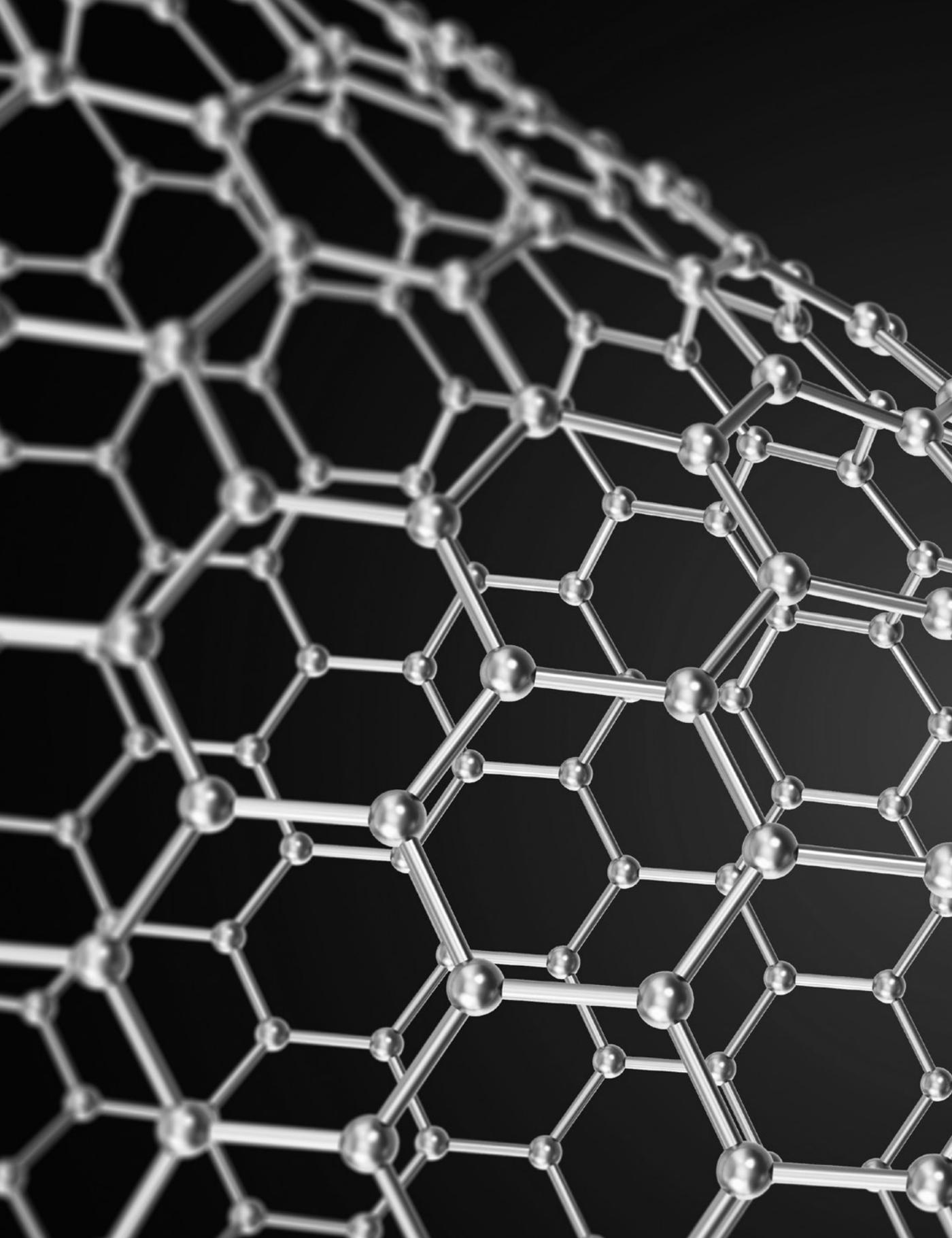
En el libro se asume que el lector tiene un conocimiento básico de los principios de la programación de ordenadores, pero no establece requisitos específicos de lenguajes o tecnología, lo cual lo hace apto para profesionales o estudiantes provenientes o enfocados en distintas áreas. Los ejemplos en el libro están dados en diferentes lenguajes de programación con lo que se pueden comparar distintas implementaciones de una misma solución y son fácilmente comprensibles incluso si no se es experto en dichos lenguajes.

La estructura de este libro comprende una parte inicial en la cual se tratan las estructuras de datos y una segunda dedicada a los algoritmos. En la primera parte, la autora explica en detalle las ideas que subyacen en las estructuras de datos que más utilizamos quienes nos dedicamos al desarrollo de software para hacer una representación de la realidad que nos permita atacar de una forma adecuada los problemas que debemos resolver en el desempeño de nuestra profesión. El conocimiento de las diferentes estructuras de datos y la comprensión de cuándo y por qué utilizar cada una de ellas es parte fundamental para la construcción de programas que sean a la vez eficientes, comprensibles y de fácil mantenimiento.

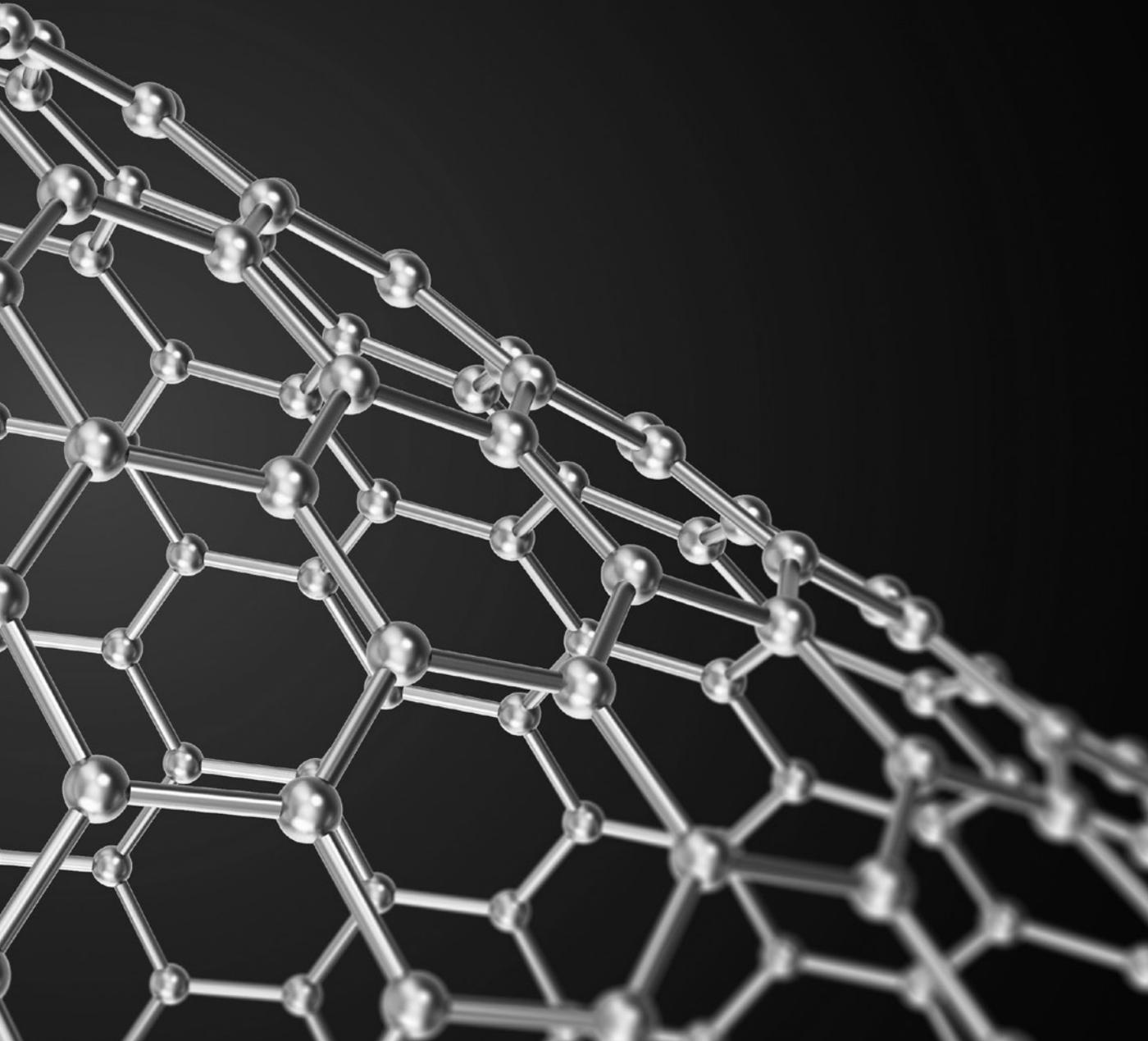
En la segunda parte del libro se tratan los algoritmos, partiendo de una presentación conceptual de los mismos, su definición y características generales, continuando con el estudio de diferentes algoritmos partiendo de los más simples y subiendo paulatinamente en complejidad. La función del aprendizaje de estos algoritmos no es necesariamente su utilización en la práctica, puesto que el desarrollador ya podrá encontrar bibliotecas que los implementan, sino ayudar a la comprensión de lo que sucede entre bambalinas cuando usamos estas bibliotecas y sobre todo estimular el hábito de atacar con método la resolución de problemas: lo importante no es el destino sino el camino.

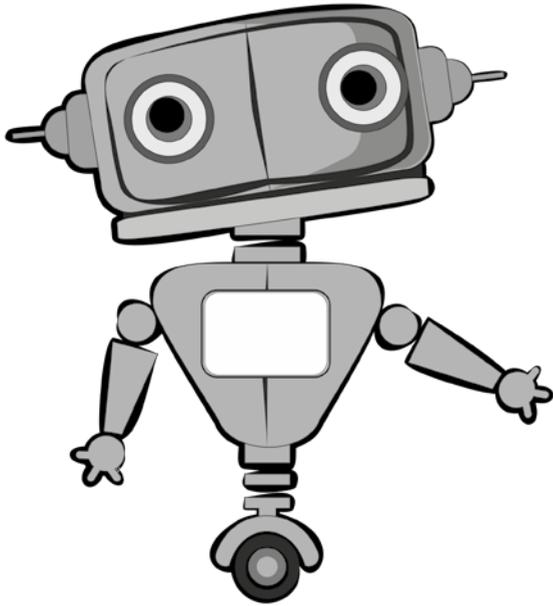
Cada capítulo tiene además una serie de ejercicios para ayudar al lector a fijar las ideas y evaluar su comprensión. En definitiva, un estudio cuidadoso de este libro ayudará en gran modo a los lectores a desarrollar las habilidades necesarias para resolver adecuadamente los problemas que se le presentarán en el desarrollo de esta profesión al tiempo que le ayudarán a comprender los retos que se plantean al resolver de forma eficiente un problema informático.

Pablo C. Lucien



Estructuras de datos





01

Introducción a las estructuras de datos

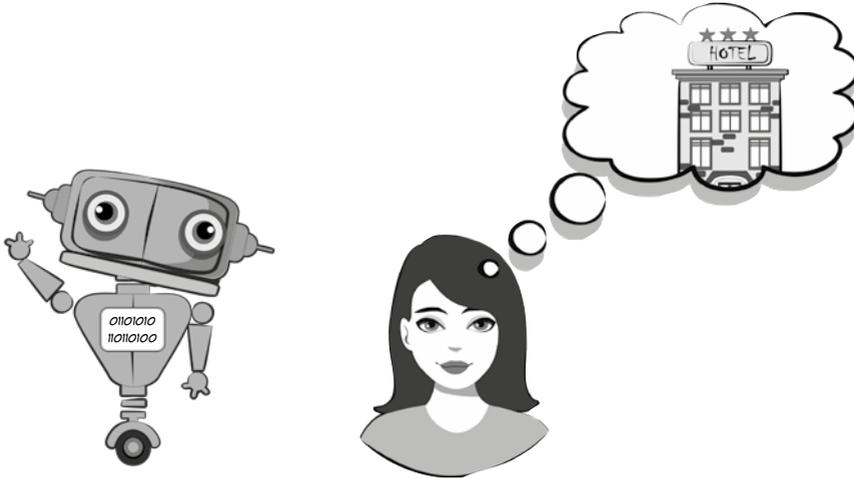
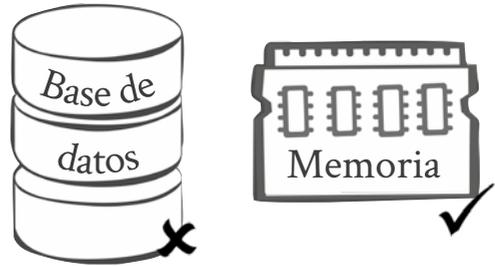
En este capítulo aprenderás a:

- Diferenciar entre estructuras de datos y tipos abstractos de datos.
- Identificar nociones de complejidad.
- Escoger tipos abstractos de datos y/o estructuras de datos.

Estructuras de datos y tipos abstractos de datos

En este libro conoceremos varias formas de organizar los datos que maneja nuestro programa. No estamos hablando de los datos que persistirán en base de datos, es decir, que seguirán ahí almacenados esté ejecutándose o no nuestro programa, sino de los datos que un programa necesita tener en memoria durante su ejecución.

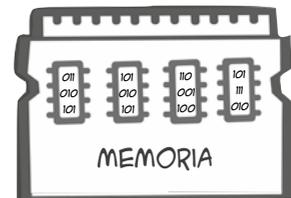
Aunque no quiero adentrarme en detalles demasiado técnicos, cada programa en memoria dispone de una zona de memoria para almacenar sus datos de trabajo. Una zona de memoria no es más que un espacio en el que se pueden almacenar ceros y unos porque, en esto de la informática y la programación, todo son ceros y unos. Los ordenadores funcionan con ceros y unos, pero los humanos necesitamos representar y organizar los datos de forma distinta.



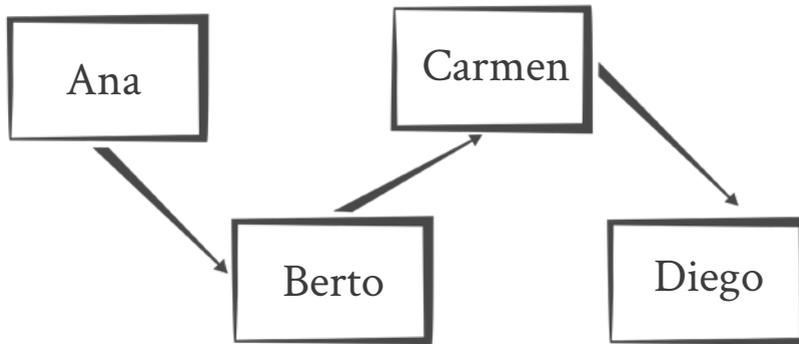
Cuando tenemos muy pocos datos que manejar, podemos usar los tipos básicos, o quizá registros o tuplas, pero para grandes cantidades de datos, hay que manejar estructuras de datos más complejas, como listas, conjuntos, diccionarios...

Hablamos de estructuras de datos cuando nos referimos a la organización, digamos, física, de los datos en memoria.

Cuando abordemos cada **estructura de datos**, estudiaremos cómo sería su representación en memoria, así como las operaciones que podemos hacer sobre ella y qué costes tienen.



Pero a un nivel más alto, antes debemos aprender algunos **tipos abstractos de datos**. Los tipos abstractos de datos son un conjunto de datos a los cuales se les asocian unas operaciones. Nos ofrecen una interfaz que define cómo acceder a esos datos o cómo manipularlos.



Si las estructuras de datos hemos dicho que eran la organización física de los datos, los tipos abstractos de datos nos mostrarán la organización lógica de estos. Un mismo tipo abstracto puede ser implementado usando distintas estructuras de datos. Del mismo modo, algunas estructuras podrían usarse para varios tipos abstractos. No te preocupes, vamos a verlos por separado, pero, a la vez, estudiaremos sus vínculos.

Introducción a la complejidad

El cálculo de la complejidad de los algoritmos y operaciones de un programa es toda una ciencia. Así que mejor no nos meternos en exceso en detalle, porque no quiero que cierres ya el libro. Pero es preciso que introduzcamos un poco la idea.

Tenemos varios tipos abstractos de datos con ciertas operaciones para acceder y manejar a los datos, y distintas estructuras de datos para implementarlos. Y en función de cómo implementemos, por ejemplo, una lista, el coste de cada operación que podemos hacer sobre una lista varía.

Siguiendo con el ejemplo de la lista, operaciones típicas sobre ella son añadir o encontrar un elemento. Contaremos con implementaciones en las que quizá añadir un elemento tenga una solución muy directa, pero encontrarlo requiera una búsqueda elemento a elemento. También podemos tener una implementación en la que el coste de añadir un elemento sea elevado, pero encontrarlo sea muy fácil.

Índice alfabético

A

aleatorio/a, 95, 189
 algoritmo, 20, 22, 24, 45, 64-65, 73, 105, 118, 129-139, 141-147, 149-155, 157-159, 161-163, 165-168, 170-178, 182-183, 185-187, 189-192, 195, 197-205, 208, 210-215, 217-218, 222, 225, 227-228, 235, 244, 248-249, 251-254, 256-257, 259, 262-263, 265-268
 árbol, 39-42, 44, 46-49, 69-70, 74, 76, 104, 115-116, 118-119, 121-126, 200-201, 228, 231-232, 237-238, 242, 244, 248, 251
 asignación, 137, 158-159

B

binaria, 21, 63-64, 200, 203, 205, 207, 209-211
 Blockly, 158-159
 Boyer-Moore, 213, 218-219, 221, 223, 225, 248-249
 buenas prácticas, 103, 142
 búsqueda, 20-21, 45, 63-64, 67, 70, 73, 112, 125, 199-205, 207, 209-211, 218-219, 221, 223, 225, 248, 262

C

C, 53
 César, 213-215, 217, 248-249
 cifrado, 65, 213-217
 clave, 23, 37-38, 66, 98, 100-101, 103-104, 106-110, 112, 118, 122-123, 136, 192, 235, 237
 cola, 31, 35-37, 46, 120-121, 228-230, 237-239, 242-245
 cola con prioridad, 228, 237, 242
 colección, 34-35, 46, 91, 101
 complejidad, 18, 20-23, 25, 62-64, 67, 70, 81, 83-84, 91, 115, 119, 125-127, 141, 149-152, 157, 165, 173, 177, 185, 188-189, 192, 202, 208, 225, 256, 259-260
 compresión, 65, 213, 227, 229, 231, 233, 235, 237-239, 241, 243-245
 conjunto, 20, 33-34, 38, 40, 42, 44-48, 91-96, 99, 101-102, 114, 116, 131, 135, 147, 244
 constante, 21-25, 62, 64, 67, 81, 83, 91-93, 100, 137-139, 143-144, 150-151, 157, 206, 208, 256
 cuadrática, 22, 24-25, 173, 177, 192, 265

D

descompresión, 234, 238, 241
 diagrama de flujo, 135-136, 154, 156, 164, 173, 183, 186, 194, 225, 235, 255, 257, 259, 265
 diccionario, 37-38, 108, 148, 203, 234-235, 237-239, 241, 243-245, 248-249
 dicotómica, 200, 203-205, 207, 209
 Dijkstra, 200
 dinámica, 57, 60, 165, 188, 261
 divide y vencerás, 149, 183, 189, 265

E

elemento, 20-21, 31-32, 34-38, 40, 42, 45-46, 48, 54, 56-57, 59-64, 66, 68-70, 82-84, 87-89, 92, 95-99, 104, 106, 109-110, 112-113, 116, 118-123, 125-127, 135, 145, 148-149, 163, 171-172, 178, 180, 188-191, 194, 196, 198, 200-205, 208-211, 235, 237, 243
 EntrySet, 101-102, 107, 245
 Eratóstenes, 132, 257, 259, 261, 267-268
 estática, 54, 56, 61, 66, 68, 72, 188
 Euclides, 253-255
 existe, 52, 64, 73, 81-82, 103, 132, 145, 150-152, 187, 201

F

Fibonacci, 153-154, 162-169
 FIFO, 35-37
 funcional, 192, 194, 196
 física, 19-20, 23, 31, 68

G

grafo, 42-46, 48-49, 71-73

H

hash, 65-68, 75, 91, 96, 98, 100, 102-106, 109, 111-113
 HashMap, 91-93, 95, 99-103, 105-111, 244-245
 HashSet, 90-93, 95, 97, 100, 126
 Haskell, 196
 heurística, 218
 Huffman, 213, 227-228, 244-246

I

imperativo, 192
 iteración, 21, 91, 100, 102, 119, 137, 139, 150-151,
 221, 230, 256, 268

J

Java, 28, 53, 78-81, 84, 90-91, 93, 99-100, 106, 113,
 115-116, 120, 123, 126, 158, 161, 165, 168, 174-175,
 177, 186-188, 195, 208-211, 217, 225, 244, 261-262
 JavaScript, 158-159, 161, 166, 169, 174-175, 203, 256,
 261-262

K

Karatsuba, 253, 263-265, 267-268

L

lenguaje, 27-28, 32, 34, 51, 53-54, 64, 79, 132-136,
 138-139, 142, 145, 149, 158, 161-162, 187, 196, 227
 LIFO, 35, 37, 48
 lineal, 21-25, 40, 61-64, 67, 81, 83-84, 115, 127, 151-
 152, 185, 192, 200-202, 208, 210-211, 225, 256, 260
 lista, 20-21, 24, 30-33, 35, 37-38, 46-48, 54, 56-57,
 59-64, 66, 68-70, 72-73, 75, 77, 80-82, 84-91, 98,
 101, 109, 111-112, 116, 127, 131, 137, 145, 148-149,
 162-163, 165, 171-174, 176-198, 201-211, 222, 228,
 243, 259, 261
 logarítmica, 21, 24-25, 64, 67, 70, 115, 119, 125, 151,
 165, 185, 208, 256, 260
 lógica, 20, 88

M

máximo común divisor, 253-255
 minimax, 201
 mod, 254-256

N

números grandes, 28, 253, 263

P

para todo, 145, 150-152, 163, 240
 patrón, 45, 218, 221-222
 pila, 35-37, 46-48
 posición, 21, 24, 28, 31-32, 34-37, 42, 46, 48, 51-52,
 54, 56, 59, 61-62, 66-68, 70-71, 74, 76, 82-89, 97-
 100, 103, 110-113, 126-127, 163, 171-172, 175, 177,
 180-181, 185, 189, 197-198, 201-205, 207-210, 215,
 218-225, 250, 259, 263

primos, 132-133, 253, 257-259, 261-262, 267-268
 prioridad, 37, 228, 237, 242
 proceso, 133-135, 180, 219, 222, 234, 244, 248, 257
 pseudocódigo, 130, 134-136, 139, 151, 154-155,
 164-165, 173, 185-186, 194, 207, 216, 225, 235, 255,
 259, 265
 puntero, 28, 51-52, 57, 59, 61-62, 64, 66, 76
 Python, 165-166, 177-178, 186-188, 203, 209-211,
 217, 226, 242, 256

R

recursividad, 141, 148-149, 183, 205

S

secuencia, 28, 32, 41, 54, 132, 137, 162-163, 165, 172,
 218, 222
 secuencial, 132, 137-139, 200-201
 selección, 137
 Set, 33, 90-93, 106-107, 114-116
 SortedSet, 114-115
 stack, 35

T

tabla, 27-29, 65-68, 75, 77, 91, 96-101, 103, 106, 109,
 111-113, 126, 214, 221-225, 228, 232-235, 238-239,
 242, 248, 251, 257, 267
 TreeSet, 113, 115-117, 119-121, 123, 125-126

V

valor, 23, 27-29, 32, 36-38, 49, 51-53, 55-57, 59,
 65-67, 69, 71, 74, 76, 89, 92, 95, 100-101, 103-104,
 106-110, 112, 117, 137, 142, 144, 147, 149-152,
 157-159, 172, 177, 185, 189, 192, 196, 204-207, 209,
 211, 220, 235
 variable, 27, 51-53, 83, 89, 95, 137-139, 142, 144, 148,
 150-152, 158, 178, 228
 voraz, 201