

Manual Imprescindible

# Curso de programación Java

Mariona Nadal Farré

**ANAYA**  
MULTIMEDIA

**N**

Manual Imprescindible

# Java

Curso de programación

Mariona Nadal Farré



# Manual Imprescindible

**Realización y adaptación de cubierta:** Celia Antón Santos

**Diseño de maqueta:** Laura Apolonio Guerra

**Revisión:** Gelsys M. García Lorenzo y Gustavo Pérez

**Maquetación:** Claudia Valdés-Miranda Cros

**Responsable editorial:** Eugenio Tuya Feijoó

Primera edición electrónica: 2021

Todos los nombres propios de programas, sistemas operativos, equipos hardware, etc., que aparecen en este libro son marcas registradas de sus respectivas compañías u 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 reprodujeran, 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.

Imágenes no aportadas por la autora: © 2021 iStockphoto LP/ Getty Images

Edición española:

© EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.), 2021  
Calle de Juan Ignacio Luca de Tena, 15, 28027 Madrid.

ISBN: 978-84-415-4425-3

Edición digital sobre la 1.<sup>a</sup> edición impresa

*A todos mis júniores, tanto a aquellos que disfrutaron con el «macetohuerto», como  
los que lo sufrieron (incluso a los que ni se enteraron de que lo hicimos). ;)  
¡Con vosotros empezó todo!*

# Mariona Nadal Farré

## SOBRE LA AUTORA



**Mariona Nadal Farré** (Barcelona, 1980) 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.

También es instructora de LinkedIn Learning, donde cuenta con un número creciente de cursos sobre Fundamentos de la Programación y Java.

Su manera de escribir 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/dsrschoool](http://bit.ly/dsrschoool)

### Cursos en LinkedIn Learning



[bit.ly/lilmnadal](http://bit.ly/lilmnadal)



## gracias

*Gracias a mi madre, Reyes, por regalarme, con gran esfuerzo, mi primer ordenador. A Fernando, por retarme a usarlo más y mejor.*

*Gracias a Julio, por empujarme a aprender Java fuera del temario oficial y por acompañarme en mi aprendizaje de la programación; y a Carmen, por convertirme en una programadora profesional.*

*Gracias a quien decidiera que yo me encargara durante un tiempo de las formaciones de los nuevos júniores, y a todos ellos por enseñarme mientras aprendían.*

*Gracias a Carlos por ficharme como instructora en LinkedIn Learning; y a Eugenio, por ofrecerme la oportunidad de escribir este libro.*

*Gracias a Gelsys y Claudia, y a Celia y el resto del equipo de Anaya Multimedia, por convertir unos textos en esta obra.*

*Gracias a mis hermanos, Guillem y Pol, por alejarme del ordenador regularmente.*

*Gracias también a Eduardo, Pablo, Mike, Ricardo, Xavier y a Quines Mones, a cada uno por sus distintas contribuciones y apoyos durante mis procesos creativos.*

*Gracias a todas aquellas personas que, aunque no mencione explícitamente, han hecho sus aportaciones, a lo largo de mi vida, que me han llevado a escribir este volumen. Y, por supuesto, también gracias a todo el personal involucrado en la impresión, distribución, exposición, reposición, venta, recomendación, reparto... sin su trabajo no sería posible que tengas hoy en tus manos este manual.*

# ndice de contenidos

<b>Cómo usar este libro</b>	<b>16</b>
<b>Introducción</b>	<b>20</b>

## **PARTE 1. Estructuras básicas** **23**

### **1. Mi primer programa** **25**

Introducción	26
Creando una clase en Java	26
El método main en Java	28
Ejecutar el método main de Java desde la línea de comandos	30
¿Qué puede salir mal con el método main de Java?	31
Mostrando mensajes al usuario en Java (System.out.println)	35
Test y ejercicios	36
Soluciones	37

### **2. Argumentos, variables, métodos y operadores** **41**

Introducción	42
Argumentos	42
Los argumentos del main y las secuencias	42
¿Cómo ejecutar el main con argumentos de texto?	44
Variables	46
Declaración y uso de variables	46
Tipos de las variables en Java	48
Métodos	49
Declaración de métodos	49
Uso de métodos	51



Operadores .....	53
Instrucción de asignación .....	53
Operadores aritméticos .....	53
Operadores relacionales .....	55
Operadores lógicos .....	57
Operadores bit a bit .....	58
Test y ejercicios .....	59
Soluciones .....	63
<b>3. Estructuras condicionales</b>	<b>73</b>
Introducción .....	74
Una condición: si (if)... si no (else)...	74
Varias condiciones: si no si (else if)...	75
Constantes .....	78
Muchas condiciones: switch .....	80
Cuando no hace falta el if .....	81
El operador ternario .....	82
Test y ejercicios .....	83
Soluciones .....	86
<b>4. Estructuras iterativas</b>	<b>95</b>
Introducción .....	96
Bucle «para cada» elemento de una colección (for each) .....	96
Bucle «para» (for) .....	97
Bucle «mientras» se cumpla la condición (while) .....	98
Bucle «haz mientras» se cumpla la condición (do while) .....	100
La clase Scanner .....	102
Bucles anidados .....	104
Caracteres especiales .....	105
Test y ejercicios .....	107
Soluciones .....	110
<b>5. Proyecto «piedra, papel, tijeras»</b>	<b>121</b>
Introducción .....	122
Presentación del programa .....	123
Preparar el esqueleto y las constantes necesarias .....	123
Mostrar instrucciones al usuario .....	124
Generar jugada del ordenador .....	125
Recoger jugada del usuario .....	126
Interpretar jugada del usuario .....	127
Calcular el ganador de la jugada .....	130
Mostrar el resultado de la jugada .....	133
Hacer el juego repetitivo .....	134
Mejoras y evoluciones .....	137

<b>6. Diseño orientado a objetos</b>	<b>141</b>
Introducción .....	142
Requisitos del proyecto .....	142
Extracción de conceptos .....	143
Boceto del modelo conceptual .....	145
Representación del modelo conceptual en UML .....	147
Test .....	148
Soluciones .....	149
<b>7. Clases y objetos</b>	<b>151</b>
Introducción .....	152
Paquetes .....	152
Interfaces .....	155
Las interfaces de sOOPer .....	155
Clases abstractas .....	158
Las clases abstractas de sOOPer .....	158
Clases «normales» .....	162
Las clases de sOOPer .....	162
Enumerados .....	165
Los enumerados de sOOPer .....	165
Objetos .....	167
Los objetos de sOOPer .....	167
Los objetos en la máquina virtual de Java .....	170
Test .....	172
Soluciones .....	173
<b>8. Relaciones orientadas a objetos</b>	<b>175</b>
Introducción .....	176
Herencia .....	176
La herencia en sOOPer .....	177
Sobrecarga y sobrescritura .....	183
Sobrescritura de toString en el sOOPer .....	184
Algoritmo de distribución de productos en contenedores en sOOPer .....	188
Reparto de responsabilidades .....	190
Reparto de responsabilidades en sOOPer .....	191
Diagrama de secuencia .....	195
Diagrama de secuencia del método addProducto .....	195
Detección y corrección de problemas .....	196
Detección y corrección de problemas en sOOPer .....	197
Probado no exhaustivo de la implementación .....	203

Modificadores de Java .....	207
Ejemplo estático .....	209
Test .....	212
Soluciones .....	213
<b>9. Proyecto «macetohuerto» .....</b>	<b>215</b>
Introducción .....	216
Presentación del proyecto .....	216
Extracción de conceptos del macetohuerto .....	218
Estructura de paquetes del macetohuerto .....	219
Interfaces del macetohuerto .....	219
Interfaz IHuerto .....	219
Interfaz IMaceta .....	220
Enumerado FormaMaceta .....	220
Interfaz IPlanta .....	220
Enumerado Familia .....	221
Enumerado Especie .....	221
Clases de la estructura de macetas del macetohuerto .....	221
Clase abstracta Maceta .....	221
Clase MacetaRectangular .....	224
Clase MacetaTubular .....	225
Clases de la estructura de plantas del macetohuerto .....	226
Clase abstracta Planta .....	226
Clase abstracta PlantaAromatica .....	229
Clase abstracta PlantaFruto .....	229
Clase abstracta PlantaHoja .....	230
Clase abstracta PlantaRaiz .....	230
Clase Hinojo .....	230
Clase Lechuga .....	231
Clase Perejil .....	231
Clase Tomate .....	232
Clase TomateCherry .....	232
Nueva implementación de la clase Tomate .....	232
Clase Zanahoria .....	233
El huerto y sus procesos .....	233
Clase Huerto .....	233
Método plantar de la clase Maceta .....	235
Método esCompatible de la clase Planta .....	236
Método tengoEspacio de la clase Planta .....	236
Método tengoEspacio de la clase PlantaRaiz .....	237
Pruebas .....	238
Clase Sistema .....	238

## 10. Manejo de excepciones 245

Introducción .....	246
¿Qué son las excepciones? .....	246
¿Qué es una excepción? .....	246
Tipos de excepciones .....	247
Naturaleza de las excepciones .....	253
API de las excepciones .....	254
Ejemplo práctico .....	256
¿Cómo manejar las excepciones? .....	259
¿De dónde vienen las excepciones? throw .....	259
¿Son contagiosas? throws .....	260
¿Pero no se pueden tratar? try / catch .....	261
¿Pero no se pueden tratar? (Más complicado) .....	262
¿Siempre hay que tratarlas? .....	265
¡Finalmente! finally .....	267
Con recursos (try) .....	268
Multicaptura (catch) .....	270
Ejemplo práctico .....	270
Malas prácticas .....	279
Comérsela con patatas .....	279
Comérsela con lechuga .....	281
Perder la memoria histórica .....	283
Generalizar .....	284
Conclusión .....	286
Test y ejercicios .....	286
Soluciones .....	291

## 11. Depuración 295

Introducción .....	296
Cómo depurar en Eclipse .....	296
Depurando Supermercado .....	297
Para qué sirve depurar .....	300
Test .....	301
Soluciones .....	302

## 12. Test unitarios 305

Introducción .....	306
Test Driven Development .....	306
JUnit .....	307
Esqueleto de un test unitario .....	308
Primeros test unitarios .....	310

Detección de regresiones .....	312
Elección de los test unitarios adecuados .....	314
Testando cosas que van mal .....	318
Objetos maquetados (mocks) .....	322
Interfaz SuperCalculadora .....	322
Clase SuperCalculadoraImpl .....	322
Clase de test SuperCalculadoraTest .....	323
Configuración para utilizar Mockito .....	323
Clase de test SuperCalculadoraTest utilizando Mockito .....	324
Test .....	325
Soluciones .....	326
<b>13. Trazas de ejecución</b> .....	<b>329</b>
Introducción .....	330
Niveles de prioridad .....	330
Configuración .....	331
Uso .....	332
Buenas prácticas .....	332
Ejemplo práctico .....	333
Dependencias .....	333
Configuración .....	334
Clase de ejemplo App.java .....	335
Salida por consola .....	336
Ficheros de logs .....	337
Test y ejercicios .....	338
Soluciones .....	339
<b>14. Proyecto «Gestión de récords»</b> .....	<b>341</b>
Introducción .....	342
Funcionalidad básica del programa .....	343
Clases RecordsManager .....	343
Validaciones .....	345
Clase RecordsManager .....	346
Clase PlayerNameTooShortException .....	348
Clase ScoreTooLowException .....	348
Control de errores .....	349
Clase RecordsManager .....	349
puntuaciones.txt .....	351
Salida por consola .....	352
Fichero de salida (records.txt) .....	352
Test unitarios .....	353
Clase RecordsManagerTest .....	353
Mejoras y evoluciones .....	355

**15. Estructuras de datos 359**

Introducción .....	360
Cadenas de texto .....	360
Clase String .....	361
Clase Character .....	362
Clases StringBuffer y StringBuilder .....	363
Internacionalización y localización .....	363
Clase MessageFormat .....	364
Números .....	366
Autoboxing y unboxing .....	366
Grandes números y alta precisión .....	367
Clase NumberFormat .....	368
Fechas .....	369
Fechas a la antigua .....	369
Fechas a partir de Java 8 (paquete java.time) .....	371
Colecciones .....	373
Tipos genéricos .....	374
Listas .....	375
Conjuntos .....	376
Funciones hash .....	377
Diccionarios .....	379
Test y ejercicios .....	382
Soluciones .....	384

**16. Bases de datos: Mapeo Objeto-Relacional 393**

Introducción .....	394
Herramientas necesarias .....	395
Configuración del proyecto .....	396
Entidades .....	399
Entidad Pedido .....	401
Objetos de Acceso a Datos (DAO) .....	401
Interfaz Dao .....	402
Gestor de entidades .....	403
El DAO abstracto .....	405
Los DAO del gestor .....	407
Consultas simples .....	409
Consulta del pedido más reciente .....	410
Consulta de pedidos de la semana pasada .....	413
Relaciones 1:N .....	415
Entidad Albaran .....	415
El DAO de Albaran .....	418

Relaciones 1:N .....	418
Entidad Factura .....	418
El DAO de Factura .....	420
Relaciones M:N .....	420
Entidad Producto .....	421
Gestor de Pedidos .....	422
Relaciones M:N bidireccionales .....	423
Consultas con criterios de búsqueda .....	425
Pros y contras de criterios y queries .....	425
Recuperando datos sin consultas. ....	426
Clase de prueba SinQueries .....	426
Las queries que no vemos .....	427
Test .....	428
Soluciones .....	430
<b>17. Expresiones lambda y Streams</b> .....	<b>433</b>
Introducción .....	434
Expresiones lambda .....	434
Calculadora lambda .....	434
Streams .....	436
Cuentas varias .....	436
Test y ejercicios. ....	437
Soluciones .....	438
<b>18. Proyecto «otra reunión más»</b> .....	<b>441</b>
Introducción .....	442
Código de base. ....	442
Cartel de sala .....	442
Clase ReunionDao. ....	443
Clase CartelSala .....	443
Informe de reunión .....	445
Clase InformeReunion .....	445
Informes de todas las reuniones. ....	447
Clase InformeReuniones .....	447
Mejoras y evoluciones .....	451
Esto es solo el principio .....	451
<b>Índice alfabético</b> .....	<b>452</b>

# Cómo usar este libro

## A quién va dirigido y qué es necesario para empezar

Este libro va dirigido principalmente a programadores noveles de Java, que comienzan a dar sus primeros pasos en el mundo de la programación. Empezamos desde cero para, paso a paso, lograr el desarrollo de pequeños programas.

Los programadores con cierta experiencia también pueden usar este libro para afianzar algunos conceptos más complejos, yendo directamente al capítulo que necesiten o, para practicar test y ejercicios, yendo al final de cada capítulo o a los capítulos finales de cada parte.

Hablo de programadores noveles y de programadores con cierta experiencia, pero ¿verdad que no he dicho programadores jóvenes o mayores? Porque poco importa la edad para empezar a programar. Este libro está pensado para ayudar a aprender a programar a:

- Niños (y niñas, evidentemente) mayores y adolescentes, que como actividad extraescolar o curricular quieren sacarle el máximo partido a su razonamiento lógico y prepararse para un mundo que nos exige ser capaces de afrontar y resolver problemas.
- Jóvenes estudiantes, ya estén estudiando programación ya sean de otras ramas del conocimiento: para casi cualquier profesión del futuro saber programar será un plus muy valorado.
- Adultos, profesionales o aficionados, programadores de otros lenguajes o con otras profesiones, reciclarse y seguir aprendiendo forma parte de nuestro día a día.
- Mayores, con más tiempo disponible y aún mucha curiosidad vital. Nunca es tarde para lanzarse a programar.

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

- JDK (kit de desarrollo Java), disponible en la web [oracle.com](http://oracle.com). No es suficiente con la máquina virtual de Java (JVM) que suele estar ya instalada. Entre otras utilidades, incluye los ejecutables `javac` y `java` para compilar y ejecutar el código sin utilizar otras herramientas.
- Un IDE (entorno de desarrollo integrado), 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 he utilizado Eclipse IDE, pero, si prefieres otro, también te sirve. Puedes descargar Eclipse en [eclipse.org](http://eclipse.org).

Si no puedes instalar estas herramientas, pero tienes acceso a internet, busca «java compiler online» y encontrarás páginas web que te permitirán probar tu código online, desde una tablet o un móvil. Si bien no son los dispositivos más adecuados para programar, sí te servirán para seguir aprendiendo.



## Estructura del libro

Esta obra se divide en cuatro partes. Cada una de ellas trabaja tres o cuatro temas relacionados. El último capítulo es un proyecto práctico con el que afianzar los conocimientos adquiridos.

La primera parte nos introduce al mundo de la programación en general mediante la **programación estructurada**, es decir, los `if`, `for`, métodos y variables... ese tipo de cosas básicas. Si es la primera vez que vas a programar es imprescindible que empieces por aquí. Si ya sabes programar en algún otro lenguaje, te convendría al menos una lectura rápida para asociar tus conocimientos previos con sus equivalentes en Java. Si ya has hecho tus pinitos en Java, tampoco está de más que le eches un ojo, aunque quizá no sea lo más prioritario.

La segunda parte está dedicada a la **orientación a objetos** que es el paradigma de programación que realmente se utiliza en Java. Aprenderás desde cómo extraer los conceptos de un problema hasta convertirlos en una jerarquía de clases, aprendiendo a repartir funcionalidades, la reutilización de código mediante herencia, la sobrescritura y la sobrecarga de código.

En la tercera parte veremos **buenas prácticas** que quizá no son necesarias en prácticas académicas o proyectos personales, pero sí son muy importantes en proyectos profesionales. Por favor, dedícale especial energía al tema de las excepciones. Era el tópico en el que mis alumnos siempre querían profundizar más porque por lo general se trata muy superficialmente. Y sin entender bien las excepciones, no es posible manejarlas bien. Depuración, test unitarios y trazas de ejecución contribuyen a hacer el código mantenible.

Terminamos con la cuarta parte dedicada a los **datos**. En ella se explican tanto las estructuras de datos que se utilizan durante la ejecución de los programas (textos, números, colecciones...) como el acceso a las bases de datos que almacenan de forma persistente la información, sin olvidar las expresiones lambda y los *streams*.

Dentro de los capítulos «de teoría», además de los ejemplos, encontrarás test para afianzar conocimientos y ejercicios para ponerlos en práctica. Intenta hacerlos antes de mirar las respuestas, pero no te saltes la lectura de la solución propuesta porque puede que te explique algo nuevo ahí mismo. Sin embargo, el capítulo final de cada parte es un proyecto en sí, a desarrollar tú y yo juntos. Esta vez, sin test ni ejercicios adicionales.

## 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`.
- Para indicar la secuencia para ejecutar una opción determinada, se ha decidido abreviar su escritura presentando la secuencia de menús u opciones en el orden el que deben

seleccionarse, separados por el signo «mayor que» (>). Por ejemplo, en lugar de indicar que seleccionemos la opción Perspectiva del menú Ventana y luego la opción Abrir Perspectiva, indicaremos directamente Ventana>Perspectiva>Abrir Perspectiva.

- 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 referenciando 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.*

#### ADVERTENCIA

*Para evitar posibles errores como consecuencia de una operación mal realizada.*

#### TRUCO:

*Consejos y artimañas para facilitar el trabajo o conseguir mejores resultados.*

## Información de soporte

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



Repositorio en git  
[bit.ly/gitMIJava](https://bit.ly/gitMIJava)

Puedes descargar todo en un fichero zip, pero te recomiendo aprender las operaciones básicas de git para poder sacarle el máximo provecho, pues algunos de los ficheros pueden tener varias versiones, en

función de cómo vamos trabajando sobre ellos. En esta figura se aprecian las dos versiones

History for [Anaya-MI-Java](#) / cap01 / [Ejemplo01\\_01a.java](#)

Commits on Mar 29, 2021

3. Qué puede salir mal con el método main de Java - SOLUCIÓN

MarionaDSR committed 28 days ago

3. Qué puede salir mal con el método main de Java

MarionaDSR committed 28 days ago

**Figura C.1.** Histórico de un fichero.

del primer ejemplo. Coge la primera (normalmente la de más abajo) y trabaja sobre ella para lograr llegar a la de más arriba, que ya contiene la solución.

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 descargarlo 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.
- El título de cada *commit* te dará una pista de en qué sección del libro se modifica ese fragmento de código.

## 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.

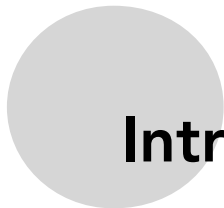
Tienes todos los ficheros disponibles para descargar, pero si de verdad quieres aprender a programar, escríbelos tú. Como los cuadernos de caligrafía de la escuela, escribir letra a letra los programas te llevará a asimilar lo que estás escribiendo. Olvídate de copiar y pegar hasta que no sepas perfectamente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

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

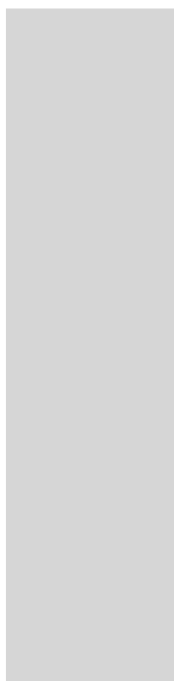
## Contacta conmigo

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

También puedes seguirme en Twitter (@MarionaJava) o en LinkedIn (marionanadal) para estar al día de mis nuevas publicaciones.



# Introducción



## L

l10n, 364  
lambda, 17, 320, 406, 433-435, 437-438, 441, 448  
length, 43, 48, 65, 116, 361  
llaves, 27-30, 51-53, 76-78, 81, 91, 156, 160, 209, 262, 365, 434  
Locale, 255, 362, 364-365, 368, 370, 390  
localización, 363-365  
log4j, 329-331, 333-335, 343, 346, 349  
logger, 331-339, 343-344, 346, 349-351  
logs, 184, 253, 255, 282-283, 330, 334, 337, 451  
long, 48, 367, 369  
línea de comandos, 30-31, 35, 38-39, 44, 66

## M

main, 25-26, 28-39, 42-44, 46, 50-52, 63, 70, 79, 103, 114, 123, 134, 142, 168, 275, 277, 306, 412  
maven, 323, 333-334, 395-397, 407  
memoria, 56, 97, 170-171, 182, 190, 248, 369, 394  
Mockito, 305, 322-326  
mocks, 322-323, 325  
modificador, 50, 175, 207-208, 210, 213  
mySQL, 395-396, 398-399, 404  
    mySQL Workbench, 397, 409, 416  
método, 25-26, 28-31, 35-39, 41-43, 46, 48-53, 62, 70-71, 74, 80, 84, 86-87, 89, 104, 142-143, 147, 152, 155-169, 176-177, 183-184, 188, 195-196, 207-209, 246, 253-256, 258-265, 267-268, 284-285, 296, 298-299, 306-313, 322, 330, 332, 337, 353, 355, 361-363, 366, 368, 374, 376, 400, 402, 405-406, 412, 426, 434, 436, 437

## N

new, 56, 103, 167, 169-171, 182, 185, 226, 259, 261  
NullPointerException, 187, 251, 253, 255, 279-280, 282-285, 289, 292, 413  
números mágicos, 78, 89, 114, 448

## O

objeto, 26-27, 29, 56, 79, 116, 142, 146, 148, 151-152, 157, 163, 167, 169-171, 176, 182, 184-185, 190-191, 195-196, 201, 207-208, 218, 222, 224, 226, 251, 253, 269, 296, 360-362, 373-374, 402, 406, 408, 418-419, 426  
objetos inmutables, 363  
objetos maquetados, 322

operaciones CRUD, 402  
operador, 41-42, 53-59, 83, 368, 434  
    operador ternario, 82-83  
Optional, 403, 405, 444-446

## P

paquete, 103, 152-154, 158, 165-166, 172-173, 177, 185, 207, 212-213, 217, 219-220, 308, 313, 327, 331, 368, 400  
parámetro, 35, 50-51, 53, 63-64, 83, 102, 155, 160, 163, 183-184, 208, 213, 226, 246, 299, 306, 413, 429-430, 434, 436, 438  
paréntesis, 29-30, 35, 51, 55, 65, 81, 87, 97, 99, 108, 110, 112, 115-116, 246, 269, 279, 374, 434, 438  
patrones de diseño, 402  
persistencia, 394, 397-398, 400, 403-404, 427-428  
pom.xml, 323, 334, 396, 442  
precedencia, 55  
preincremento, 54-55, 388  
principio KISS, 307  
prioridad, 330  
private, 50, 79, 207, 226, 308, 327  
producción, 241, 253, 255, 269, 279-280, 282, 330, 332, 395, 399, 404  
programación, 16-17, 21, 26, 42, 46, 66, 199, 369, 376-377, 426, 451  
    programación concurrente, 209  
    programación estructurada, 17, 26, 123, 137, 155, 176, 192, 207, 434  
    programación funcional, 433-434, 436  
    programación imperativa, 434  
    programación orientada a objetos, 137, 142, 155, 184, 196, 394, 399, 434  
    programación profesional, 82, 108  
protected, 79, 207, 212, 226-227, 308, 327, 353, 410  
public, 27, 29, 79, 160, 207  
puntero, 56, 170-171, 182-183, 251  
punto de ruptura, 296-298, 302-303  
punto y coma, 35-36, 51, 110, 156, 160, 166, 209, 246  
puntos suspensivos, 362, 436

## Q

query, 252, 376, 394, 405-406, 408, 410-414, 425-430, 443

## R

relación,  
    relación 1:1, 416, 419, 429, 431  
    relación 1:N, 415-419, 429, 431  
    relación M:N, 416, 419-421, 423, 429  
responsabilidades, 71, 176, 190-191, 193, 196,  
    235-236, 269  
return, 50-53, 57, 82, 291, 434  
rollback, 406  
runtime exception, 250-252, 256-257, 260, 262, 266,  
    272-274, 277-278, 280, 285, 319, 343, 413

## S

Scanner, 95, 102-103, 117, 126, 131, 343, 349-350, 445  
servidor, 26, 253, 322, 331, 395-396, 398-399, 428, 430, 451  
setter, 227, 400, 415, 418, 420-421  
short, 48, 366  
sobrecarga, 17, 175, 183, 212-213, 381  
sobrescritura, 17, 175-176, 183-184, 198-199, 201, 208,  
    212-213, 255, 258  
static, 28-30, 36-37, 50, 79, 208-213, 362  
strings, 28-29, 42-43, 46-49, 56, 63, 66-67, 103, 105,  
    108, 116, 119, 154, 160, 183, 185-186, 223, 251-252,  
    261, 278, 280-281, 285, 300, 302, 325-326, 333,  
    360-363, 370, 388, 448  
super, 163-164, 179-181, 186, 198-199, 222-225,  
    229-233, 237, 257-258, 321, 348  
switch, 73, 80-82, 90-91, 96, 133, 135, 275-277, 279, 293

## T

TDD, 305-306, 325-326  
test unitario, 17, 203, 206-207, 241, 290, 305-308, 310,  
    312-320, 322-324, 326, 341-342, 353, 451  
this, 163-164, 189-191, 227, 232, 235-236, 258, 422, 424  
throw, 246, 259-261, 263, 265-267, 278-279, 283-284,  
    286-293  
Throwable, 247, 254-255, 257-258, 285-286, 291,  
    320-321  
throws, 246, 260-261, 263, 265-267, 273-275, 278-279,  
    286-288, 290-293, 320  
TODO, 46, 47, 128-129, 132, 136, 160-161, 177, 179,  
    186, 188, 191-192, 202-203, 223, 308  
trace, 331-332, 335-336, 338  
trazas, 17, 133, 184, 235, 238, 241, 255, 271, 275-277,  
    285, 296, 312-313, 329-339, 341-342  
try, 246, 261-262, 264-270, 275, 277-280, 282-286, 288,  
    290, 292, 320, 345, 349, 412

## U

UML, 141, 146-149, 175, 195-196, 217  
unboxing, 366-367

## V

variable, 17, 27, 41-42, 46-54, 56, 58-59, 63, 67, 70, 74,  
    79-82, 84, 86-87, 96-97, 99-100, 102-104, 109, 114-115,  
    190, 208-210, 226, 246, 266, 297-300, 311, 332  
void, 28-30, 33, 37, 50, 264, 353

## W

warn, 330, 334-339, 346, 350, 352, 404  
while, 98-102, 107-110, 114-117, 134, 344, 346, 350,  
    385-386  
wrapping classes, 366

## X

XML, 331, 398-399