

YO PROGRAMO

{ en JAVA }

DANIEL E. AGUIL MALLEA



[códigos utilizados]

Introducción a la programación

utilizando JAVA como herramienta

Daniel E. Aguil Mallea

Este libro fue escrito para enseñar a programar.

Adopta el enfoque orientado a objetos y explica los conceptos básicos de la programación necesarios para resolver problemas.

Aguil Mallea, Daniel Eugenio

Yo programo : en JAVA / Daniel Eugenio Aguil Mallea ; Contribuciones de Germán Carlos Tejero ; Nadia Patricia Ramos ; Simon Paradiso ; Coordinación general de Daniel Eugenio Aguil Mallea ; Ilustrado por Ana Longo. - 1a edición para el alumno - Ushuaia : Daniel Eugenio Aguil Mallea, 2024.

296 p. : il. ; 25 x 20 cm.

ISBN 978-631-00-4641-9

1. Lenguaje de Programación. 2. Tecnología Informática. 3. Material Auxiliar para la Enseñanza. I. Tejero, Germán Carlos, colab. II. Ramos, Nadia Patricia, colab. III. Paradiso, Simon, colab. IV. Longo, Ana, illus. V. Título.

CDD 005.43

Colaboradores:

German Carlos Tejero
Nadia Patricia Ramos
Antonio Retamar
Simon Paradiso

Créditos:

Ilustraciones de capítulos por Ana Longo
<https://www.behance.net/anitalongobucco>

Imagen de tapa y contratapa por Freeepik
<https://www.freepik.es>

Retoques de diseño y estilo en imágenes por Adrian Aguil Mallea <https://www.behance.net/adrianaguilmallea>

Maquetación por Paula Castillo
<http://paucast.com.ar>



Contenidos

Prólogo | 14

Objetivos | 17

Audiencia | 18

Agradecimientos | 19

Capítulo 1 : Análisis y diseño de algoritmos

Objetivo | 22

Definiciones de conceptos básicos | 22

Modelización de problemas del mundo real | 24

Etapas en la resolución de problemas con una computadora | 26

[-] análisis | 26

[-] diseño de la solución | 28

[-] especificación de algoritmos | 29

[-] escritura de programas | 32

[-] verificación | 35

Algoritmos, programas y lenguajes de programación | 36

[-] lenguaje de programación | 39

[-] herramientas | 41

[-] mi primer programa | 42

[-] sintaxis | 44

[-] semántica | 44

Ejercicios | 46

Capítulo 2 : Representación de datos

Objetivo | 50

Dato | 50

Variable | 52

- [-] ámbito | 53
- # Tipo de dato | 53
 - [-] entero | 55
 - [-] decimal | 56
 - [-] lógico | 56
 - [-] caracter | 56
 - [-] texto | 57
- # Operaciones | 57
 - [-] operaciones de asignación | 58
 - [-] operaciones matemáticas | 58
 - [-] operaciones de comparación | 59
 - [-] operaciones lógicas | 60
 - [-] operaciones de concatenación | 61
- # Proposición | 61
 - [-] tablas de verdad | 64
- # Ejercicios | 68

Capítulo 3 : Entrada y salida de datos

- # Objetivo | 74
- # Variables, constantes y expresiones | 74
 - [-] constante | 75
 - [-] expresión | 76
- # La operación asignación | 77
- # Funciones internas | 78
- # Entrada y Salida | 82
 - [-] entrada | 84
 - [-] salida | 85
- # Ejercicios | 89

Capítulo 4 : Estructuras de control

- # Objetivo | 92
- # Flujo de ejecución | 92

- # Secuencia | 93
- # Selección | 93
- # Repetición | 98
 - [-] for | 98
 - [-] while y do-while | 99
- # Estructuras adicionales control | 103
 - [-] break | 103
 - [-] continue | 105
 - [-] switch | 106
 - [-] try-catch-finally | 108
 - [-] for-each | 109
- # Traza de ejecución | 110
- # Ejercicios | 113

Capítulo 5 : Estructuras de datos compuestas

- # Objetivo | 116
- # Datos compuestos | 116
- # Arreglos | 117
 - [-] vectores | 117
 - [-] matrices | 121
 - [-] arreglos n-dimensionales | 123
- # Ejercicios | 125
 - [-] vectores | 125
 - [-] matrices | 126

Capítulo 6 : Estructuras de programa

- # Objetivo | 132
- # Subproblemas | 132
- # Función | 133
 - [-] identificador | 134
 - [-] parámetros | 136
 - [-] variable local | 139

Librería | 141

Ejercicios | 143

Capítulo 7 : Recursión

Objetivo | 148

Recursión | 148

Ejercicios | 157

Capítulo 8 : Algoritmos fundamentales

Objetivo | 162

Algoritmos | 162

Búsqueda | 163

[-] lineal | 163

[-] binaria | 167

[-] métodos provisto por el lenguaje | 170

Ordenamiento | 171

[-] método de selección | 171

[-] método de burbuja | 174

[-] método de inserción | 176

[-] métodos previsto por el lenguaje | 179

Ejercicios | 180

Capítulo 9 : Paradigma orientado a objetos

Objetivo | 184

Paradigma | 184

Clase | 186

Objeto | 187

[-] constructor | 187

[-] el operador igualdad | 191

[-] el operador asignación | 193

[-] el valor nulo | 196

[-] contexto estático | 198

- [-] acceso | 200
- [-] mensajes | 202
- [-] this | 205
- [-] tipo de dato envoltorio | 207
- # Contenedores | 209
 - [-] lista | 211
- # Tipo abstracto de dato | 223
- # Ejercicios | 231

Capítulo 10 : Herencia y polimorfismo

- # Objetivo | 236
- # Herencia | 236
 - [-] subclases y superclases | 237
 - [-] sobrescritura de métodos | 245
- # Clase abstracta | 248
- # Interfaz | 253
 - [-] comparable | 257
 - [-] comparador | 263
- # Polimorfismo | 270
- # Genéricos | 279
- # Ejercicios | 283

Anexo | 287

- # Utilizar el entorno de desarrollo | 288

Bibliografía | 293

Índice de figuras

- Fig.01 - modelo. | 25
- Fig.02 - casos de usos. | 27
- Fig.03 - diagrama de clases. | 29
- Fig.04 - arquitectura de Von Neumann. | 37
- Fig.05 - proceso de compilación e interpretación. | 40
- Fig.06 - tabla de verdad - negación | 65
- Fig.07 - tabla de verdad - conjunción | 65
- Fig.08 - tabla de verdad - disyunción | 66
- Fig.09 - entrada y salida. | 83
- Fig. 10 - secuencia. | 93
- Fig.12 - estructura de selección con alternativa. | 96
- Fig.13 - estructura de repetición for. | 98
- Fig.14 - estructura de repetición while. | 100
- Fig.16 - for con break. | 104
- Fig.17 - for con continue. | 105
- Fig.18 - vector. | 117
- Fig.19 - matriz. | 122
- Fig.20 - partes de una función. | 134
- Fig.21 - invocación de una función. | 137
- Fig.22 - alcance. | 140
- Fig.23 - factorial recursivo. | 151
- Fig.24 - pila de llamadas. | 152
- Fig.25 - búsqueda lineal o secuencial. | 164
- Fig.26 - búsqueda lineal o secuencial ordenada. | 165
- Fig.27 - búsqueda binaria. | 168

Fig.28 - búsqueda binaria vs secuencial. | 170

Fig.29 - ordenamiento por selección. | 172

Fig.30 - ordenamiento por burbuja. | 174

Fig.31 - ordenamiento por inserción. | 177

Fig.32 - clase y objetos. | 185

Fig.33 - memoria - creación de un objeto. | 188

Fig.34 - memoria - creación de un objeto con parámetros. | 190

Fig.35 - memoria - operador de igualdad ==. | 192

Fig.36 - memoria - asignación de otro objeto. | 195

Fig.38 - lista. | 212

Fig.39 - lista de objetos. | 215

Fig.40 - mapa. | 218

Fig.41 - mapa de objetos. | 222

Fig.42 - jerarquía de clases. | 238

Fig.43 - sobrescritura de métodos. | 245

Fig.44 - clase abstracta | 249

Fig.45 - interfaz. | 254

Fig.46 - comparable. | 259

Fig.47 - comparator. | 264

Fig. Anexo 01 - Pantalla inicial. | 288

Fig. Anexo 02 - Proyecto nuevo. | 289

Fig. Anexo 03 - Clase nueva | 289

Fig. Anexo 05 - Salida de la ejecución | 293

Prólogo

¿Por qué aprender a programar?

La tecnología está en todas partes. Desde el momento que suena el despertador, cuando miramos una serie, escuchamos un podcast, vemos una transmisión en vivo, usamos el celular y tantas cosas más que se nos pueden ocurrir. Todos los dispositivos que usamos a diario funcionan con software.

Programar es el arte de crear software. Es escribir las instrucciones que le dicen a una máquina qué hacer.

Aprender a programar nos permite:

- Entender cómo funcionan los dispositivos que usamos.
- Crear nuestras propias aplicaciones y páginas web.
- Ser más creativos e innovadores.
- Tener mejores oportunidades laborales.
- Desarrollar habilidades de pensamiento crítico y habilidades para la resolución de problemas.
- Trabajar de forma lógica y metódica.
- Mejorar nuestra comunicación y la manera en que colaboramos.
- Ser más independientes y autosuficientes.

Más allá de si queremos convertirnos en un programador profesional -un desarrollador- o simplemente para aprender algo nuevo, aprender a programar es una excelente inversión.

Programar está de moda. Cada vez hay más demanda de programadores en el mercado laboral. ¿Por qué? La respuesta es sencilla: la tecnología está en todas partes. Como dijimos al principio, desde que nos despertamos hasta que nos dormimos usamos dispositivos que funcionan con software.

La digitalización ha transformado nuestra vida. Miles de aplicaciones y plataformas ne-

cesitan ser programadas para funcionar y posibilitar cada vez más funciones. Además, la automatización está presente en sectores como la industria, la robótica y la inteligencia artificial, donde los sistemas necesitan realizar tareas de forma autónoma.

Por suerte, aprender a programar es más fácil que nunca. Hay muchas plataformas de educación en línea y recursos gratuitos disponibles. Esto ha hecho que las habilidades en programación y tecnología sean más accesibles, con un impacto significativo en el mercado laboral.

Si juntamos la alta demanda de programadores, los avances tecnológicos y la tendencia a la interdisciplinariedad, encontraremos en el mundo de la programación términos que aparecen en primera plana como Machine learning, Big data, Automatización, IoT, DevOps, BlockChain. Aunque lo cierto es, como sucede con todas las modas, que dos de ellos están dando mucho que hablar últimamente: “programador fullstack” e “inteligencia artificial”.

Un programador fullstack es una persona que es capaz de trabajar en todas las capas del desarrollo de una aplicación, en general web. Esto incluye desde la parte visual -la llama-

mos front-end- hasta la lógica del negocio y sus servicios -le decimos back-end.

Si bien existen cursos y especializaciones en fullstack, para dominar cualquiera de las dos áreas es fundamental tener conocimientos sólidos de programación. Es muy difícil, por no decir imposible, que un desarrollador sea productivo sin dominar los conceptos básicos de la programación. Ser productivo implica hacer las cosas de la mejor manera posible, utilizando los recursos de manera eficiente y en un tiempo razonable.

En general un programador que no logra ser productivo en un tiempo razonable, difícilmente será tenido en cuenta para futuras ocasiones. Por lo tanto se recomienda, como se suele decir, comenzar por el principio.

La inteligencia artificial (IA) es una rama de la informática que busca crear máquinas que puedan imitar la inteligencia humana. Esto incluye el aprendizaje, el razonamiento y la resolución de problemas.

En los últimos años, ha crecido la idea de que la IA reemplazará a los programadores. Lo cierto es que no es probable que suceda en un futuro cercano o incluso a largo plazo.

La IA es una herramienta que puede ayudar a los programadores a automatizar tareas repetitivas y acelerar el proceso de desarrollo. Sin embargo, los programadores siguen siendo necesarios para:

- Diseñar y desarrollar aplicaciones de calidad.
- Comprender las necesidades de los usuarios.
- Tomar decisiones críticas durante el proceso de desarrollo.

La IA no es un sustituto completo de la creatividad, el juicio o el criterio humano. Estas habilidades, fundamentales en los programadores, se desarrollan en gran parte a través de la experiencia y no son fácilmente replicables por una máquina.

La programación es un campo en constante evolución. Los programadores necesitan estar al día con las últimas tecnologías y tendencias para seguir creciendo.

La IA puede ayudar a los programadores a mantenerse actualizados y a ser más productivos. Sin embargo, no reemplaza la necesidad de tener una comprensión profunda de los principios fundamentales de la programación. La experiencia sigue siendo esencial en la creación de software de alta calidad.

Objetivos

El principal objetivo del libro es enseñar a programar, lo que implica también proporcionar habilidades y conocimientos necesarios para crear, desarrollar y utilizar software.

Se pretende:

Fomentar el pensamiento computacional. La programación enseña a las personas a pensar de manera lógica y a descomponer problemas complejos en problemas pequeños y manejables. El pensamiento computacional es una habilidad fundamental de la disciplina y además ayuda a resolver problemas en otras áreas.

Estimular la creatividad y la innovación. La programación permite a las personas crear nuevas soluciones, desarrollar aplicaciones y diseñar sistemas que resuelvan problemas y satisfagan necesidades específicas. Aprender a programar fomenta la creatividad.

Mejorar la resolución de problemas. La programación implica enfrentarse a desafíos y buscar soluciones eficientes. Al aprender a programar, las personas adquieren habilidades para abordar problemas de manera sistemática, analizar situaciones y encontrar soluciones lógicas y estructuradas.

Impulsar el pensamiento analítico. La programación requiere un enfoque analítico para descomponer un problema en componentes más pequeños y comprender cómo interactúan entre sí. Esto ayuda a desarrollar habilidades de pensamiento crítico y analítico que son valiosas en muchas áreas profesionales.

Prepararse para la era digital. Vivimos en una sociedad cada vez más digitalizada, donde la tecnología desempeña un papel importante en casi todos los aspectos de nuestras vidas. Aprender a programar nos prepara para comprender y participar en esta era digital, y nos brinda una herramienta adicional para adaptarnos a los avances tecnológicos, que por cierto son cada vez más “vertiginosos”.

Fomentar habilidades de colaboración y trabajo en equipo. Programar no es solo un esfuerzo individual, sino que por lo general implica colaborar con otras personas, sobre todo en proyectos de mediana o alta complejidad. Aprender a programar promueve la colaboración, la comunicación y el trabajo en equipo, habilidades esenciales en el entorno laboral actual.

Audiencia

Este libro ha sido escrito para una audiencia diversa. Servirá de guía para la introducción al mundo de la programación, sin importar la formación o experiencia previa.

A los estudiantes de una carrera afín a la disciplina, les servirá para afianzar los fundamentos de la programación y desarrollar el pensamiento computacional. Este libro proporciona una base sólida para el futuro profesional en el campo de la tecnología.

A los estudiantes autodidactas, les servirá para aprender a su ritmo, sin necesidad de contar con un docente o instructor guía. El estudiante autodidacta podrá saltar de capítulo en capítulo volviendo o yendo a los lugares que quiera explorar, aunque se recomienda el recorrido en orden, dado que la complejidad de los temas que se desarrollaron se encuentran ordenados de menor a mayor.

A los profesionales de la disciplina, les servirá para actualizar sus conocimientos sobre el paradigma y quizá repasar algún detalle olvidado de los objetos. Por supuesto siempre utilizando JAVA como lenguaje.

A los profesionales de otras áreas o simplemente a quienes quieran aprender a programar por curiosidad, o para mejorar sus habilidades digitales, este libro les ofrece una introducción completa a los fundamentos de la programación, con explicaciones claras y concisas de los conceptos clave, ejemplos prácticos y ejercicios para poner en práctica los conocimientos.

Este libro te ofrece:

- Una introducción completa a los fundamentos de la programación.
- Explicaciones claras y concisas de los conceptos clave.
- Ejemplos prácticos y ejercicios para poner en práctica los conocimientos.
- Una guía completa para el aprendizaje autodidacta.
- Recursos adicionales para profundizar el aprendizaje.

Agradecimientos

Cuando decidí continuar lo que había empezado hace un tiempo y darle forma de libro, no sabía realmente el tiempo que me llevaría, por eso quiero comenzar agradeciendo a mi familia, que se bancaron todo el proceso. A mis hijos, aceptando que el tiempo dedicado al proyecto es también para ellos, aunque sé que en el fondo no terminan de comprender del todo, esta es una inversión y es para ellos. A mi “pareja” -le encanta que la presente de ese modo-, que con todo su amor siempre está acompañando el plan, poniendo el pecho a todas las circunstancias que se presentan. A mis viejitos, por su amor desde siempre y creer en lo que hago. A mis “brokis” por estar ahí para escucharme y animarme.

Un agradecimiento muy especial a mi mentora, mi profesora, mi colega, Beatriz Depetris. Fue la primera persona que confió en que me convertiría algún día en un “buen” profesional. Sus consejos, experiencia, paciencia y cariño han sido fundamentales para mi crecimiento profesional y personal. Agradezco profundamente su confianza en mí y por haberme brindado la oportunidad de aprender.

A mis amigos, colegas, profesores y a todas las personas que han estado presentes en este proceso. Agradezco su apoyo y motivación. A cada uno, sea a través de palabras cálidas, palabras que muchas veces me han movilizado y ayudado, o inclusive palabras de esas que son más difíciles de digerir pero que me han servido también, las palabras de crítica.

