

# Contenido

<b>Prólogo</b> .....	XV	1.2.8. Caracteres especiales .....	13
		1.2.9. Estructuras de control .....	15
		1.2.10. Operadores aritméticos, relacionales y lógicos.....	17
<b>CAPÍTULO 1</b>		1.2.11. Funciones o métodos estáticos ....	18
<b>Introducción al lenguaje de Programación Java</b>		1.2.12. Inferencia de tipo de dato .....	20
.....	1	1.2.13. Clases y objetos.....	20
1.1 Introducción.....	1	1.3 Cadena de caracteres.....	22
1.1.1. Java en sus inicios .....	1	1.3.1. Tratamiento de cadenas de caracteres .....	22
1.1.2. JSE y JEE – Java Estándar y Java Enterprise Edition.....	2	1.3.1.1. Determinar la longitud de una cadena .....	22
1.1.3. Desarrollo de aplicaciones .....	2	1.3.1.2. Determinar si una cadena es la cadena vacía.....	23
1.1.4. Hola Mundo .....	2	1.3.1.3. Acceder a los caracteres de una cadena .....	23
1.1.5. Entorno Integrado de Desarrollo (IDE) .....	4	1.3.1.4. Comparar cadenas .....	23
1.1.6. Versiones Java .....	4	1.3.1.5. Determinar la posición que ocupa un carácter dentro de una cadena .....	24
1.2 Elementos del lenguaje de programación .....	5	1.3.1.6. Determinar la posición que ocupa una subcadena dentro de una cadena.....	24
1.2.1. Entrada y Salida Estándar .....	5	1.3.1.7. Convertir a mayúsculas y minúsculas.....	24
1.2.2. Identificadores y declaración de variables.....	7	1.3.1.8. Subcadenas.....	25
1.2.3. Comentarios en el código .....	8	1.3.1.9. Separar una cadena en varias subcadenas .....	25
1.2.4. Tipos de dato.....	8	1.3.1.10. Conversión entre cadenas y números.....	25
1.2.4.1. Tipos de dato primitivos .....	8	1.3.1.11. Concatenar cadenas .....	26
1.2.4.2. Wrappers: clases que representan a los tipos primitivos .....	9	1.3.2. Invarianza de las cadenas de caracteres .....	26
1.2.4.3. Autoboxing: conversión automática entre tipos primitivos y wrappers .....	10	1.3.3. Cadenas de caracteres dinámicas..	27
1.2.5. Constantes .....	10		
1.2.6. Valores literales.....	11		
1.2.7. Literales expresados en otros sistemas de numeración .....	12		
1.2.7.1. Números enteros expresados en binario.....	12		

1.3.3.1. Concatenar cadenas de caracteres ..... 27  
 1.3.3.2. Modificar los caracteres de una cadena..... 27  
 1.4. Autoevaluación y ejercicios..... 28  
 1.5 Resumen..... 28

CAPÍTULO 2

**Programación orientada a objetos** ..... 29  
 2.1 Introducción..... 29  
 2.2 Clases y objetos..... 30  
 2.2.1. Métodos..... 32  
 2.2.2. Herencia y sobrescritura de métodos ..... 34  
 2.2.3. Método toString..... 35  
 2.2.4. Método equals ..... 36  
 2.2.5. Annotations ..... 37  
 2.2.6. Declarar y crear objetos..... 39  
 2.2.7. Constructores ..... 41  
 2.2.8. Breve repaso de lo visto hasta aquí 43  
 2.2.9. Convenciones de nomenclatura ..... 46  
 2.2.9.1. Nombres de clases ..... 46  
 2.2.9.2. Nombres de métodos ..... 46  
 2.2.9.3. Nombres de atributos..... 46  
 2.2.9.4. Nombres de variables de instancia ..... 47  
 2.2.9.5. Nombres de constantes ..... 47  
 2.2.10. Sobrecarga ..... 47  
 2.2.11. Encapsulamiento ..... 51  
 2.2.12. Clase GregorianCalendar..... 55  
 2.2.13. Métodos obsoletos (deprecated) . 56  
 2.2.14. Visibilidad de métodos y atributos 57  
 2.2.15. Packages (paquetes) ..... 59  
 2.2.15.1. Nombre simple de una clase .... 59  
 2.2.15.2. Nombre completo de una clase. 59  
 2.2.16. Estructura de paquetes y classpath ..... 60  
 2.2.17. Aplicación Programming Interface (API) ..... 62  
 2.2.18. Representación gráfica UML ..... 62  
 2.2.19. Importar clases de otros paquetes ..... 64  
 2.3 Herencia y polimorfismo ..... 65  
 2.3.1. Polimorfismo ..... 69  
 2.3.2. Constructores de subclases ..... 71  
 2.3.3. Referencia super..... 72  
 2.3.4. Referencia this ..... 74  
 2.3.5. Clase abstracta ..... 75

2.3.6. Constructores de clases abstractas 80  
 2.3.7. Instancias..... 84  
 2.3.8. Variables de instancia ..... 86  
 2.3.9. Variables de clase..... 89  
 2.3.10. Gestión dinámica de memoria y garbage collector ..... 89  
 2.3.11. Método finalize ..... 91  
 2.3.12. Constantes..... 92  
 2.3.13. Métodos de clase ..... 93  
 2.3.14. Clases utilitarias ..... 96  
 2.3.15. Referencias estáticas..... 96  
 2.3.16. Bloques estáticos ..... 98  
 2.3.17. Colecciones (Introducción) ..... 99  
 2.3.18. Clases genéricas ..... 105  
 2.4 Interfaces y factorías de objetos ..... 108  
 2.4.1. Desacoplamiento de clases ..... 111  
 2.4.2. Factory method ..... 114  
 2.4.3. Abstracción a través de interfaces.. 114  
 2.4.4. Interface comparable ..... 115  
 2.4.5. Desacoplar todavía más..... 122  
 2.4.6. Expresiones lambda ..... 126  
 2.4.7. Interface comparator ..... 128  
 2.5 El framework de colecciones de Java (JCF) ..... 130  
 2.5.1. Listas y colecciones ..... 130  
 2.5.2. For-each ..... 133  
 2.5.3. Cambio de implementación ..... 133  
 2.5.4. Mapas o diccionarios de datos ..... 134  
 2.5.5. Las clases collections y arrays ..... 135  
 2.5.5.1. Convertir arrays en list y viceversa ..... 135  
 2.5.5.2. Ordenar listas y arrays ..... 136  
 2.5.6. Properties..... 137  
 2.5.6.1. Grabar properties en un archivo de texto ..... 138  
 2.5.6.2. Leer properties almacenadas en un archivo de texto ..... 138  
 2.5.6.3. Archivo de propiedades ubicado dentro del classpath..... 139  
 2.5.6.4. Simplificando el modo en que accedemos a un archivo de propiedades.. 140  
 2.6. Excepciones..... 143  
 2.6.1. Introducción ..... 143  
 2.6.2. Ejemplo de uso ..... 147  
 2.6.3. Excepciones declarativas y no declarativas..... 151  
 2.6.4. Stack trace ..... 151  
 2.6.5. Excepciones propi..... 152  
 2.6.6. Bloque try-catch-finally ..... 154

2.6.7. Bloque try con recurso .....	158
2.7. Autoevaluación y ejercicios .....	159
2.8 Resumen .....	159

### CAPÍTULO 3

#### Lambda, streams y

<b>programación funcional</b> .....	161
3.1 Introducción .....	161
3.2. Interfaces funcionales .....	162
3.2.1. Prototipo de un método o función ...	163
3.2.2. Referencias a métodos .....	163
3.3. Expresiones lambda .....	164
3.4. Funciones, predicados y	
consumidores .....	166
3.4.1. La interface Predicate .....	166
3.4.2. La interface BiPredicate .....	169
3.4.3. La interface Consumer .....	171
3.4.4. La interface BiConsumer .....	172
3.4.5. La interface Function .....	173
3.4.6. La interface BiFunction .....	174
3.5. Streams .....	176
3.5.1. Introducción .....	176
3.5.2. Fundamentos .....	176
3.5.3. Operaciones sobre streams .....	178
3.5.3.1. Operaciones intermedias .....	178
3.5.3.2. Operaciones finales .....	178
3.5.4. Ejemplos .....	178
3.5.4.1. Filtrar elementos de una	
colección .....	178
3.5.4.2. Buscar un elemento dentro de	
una colección .....	179
3.5.4.3. Iterar y mostrar los elementos	
de la colección .....	180
3.5.4.4. Procesar los elementos de la	
colección .....	180
3.6. Autoevaluación y ejercicios .....	181
3.7 Resumen .....	182

### CAPÍTULO 4

<b>Acceso a bases de datos (JDBC)</b> .....	183
4.1 Introducción .....	183
4.1.1. Driver y la cadena de conexión URL	
183 .....	183
4.1.2. Queries y updates .....	184
4.1.3. Estructura de un programa que usa	
JDBC .....	184
4.1.4. Conexión cerrada vs. Conexión nula	
187 .....	187
4.2 Ejecutar sentencias SQL .....	188
4.2.1. Queries .....	189

4.2.1.1. Obtener múltiples filas .....	189
4.2.1.2. Obtener 1 o ninguna fila .....	190
4.2.1.3. Queries con JOIN .....	191
4.2.1.4. Queries de funciones de la base de	
datos .....	192
4.2.1.5. Queries parametrizados .....	192
4.2.1.6. Queries con fechas .....	193
4.2.1.7. Funciones propietarias de la base de	
datos .....	194
4.2.2. Updates .....	195
4.2.2.1. Insertar una fila .....	195
4.2.2.2. Valores autoincrementales .....	196
4.2.2.3. Modificar una fila .....	197
4.2.2.4. Borrar una fila .....	198
4.2.2.5. Transacciones .....	198
4.3 Administrar la conexión JDBC .....	201
4.3.1. Factoría para instanciar la conexión	
201 .....	201
4.3.2. Singleton para garantizar una única	
instancia de la conexión .....	203
4.3.3. Properties para parametrizar los	
atributos de la conexión .....	204
4.3.4. Bloque estático para registrar el	
driver una única vez .....	205
4.3.5. Shutdownhook para cerrar la	
conexión .....	206
4.4 Encapsular el acceso a los datos .....	208
4.4.1. Modelo de objetos o modelo de	
dominio .....	208
4.4.2. Data Access Object (DAO) .....	210
4.4.2.1. Buscar una fila (método buscar	
o find) .....	210
4.4.2.2. Acceso a los datos foráneos .....	212
4.4.2.3. Representación orientada a	
objetos (Composición) .....	213
4.4.2.4. Recuperar todas las filas (método	
buscarTodo o findAll) .....	218
4.4.2.5. Recuperar sólo un conjunto de filas	
(métodos buscarPor o findBy) .....	219
4.4.2.6. Insertar una fila .....	221
4.4.2.7. Update .....	223
4.4.2.8. Delete .....	224
4.4.3. Encapsular la transacción .....	225
4.5 Separar el acceso a los datos en API	
e implementación .....	227
4.5.1. Escribir los DAO como interfaces ...	
227 .....	227
4.5.2. Implementar las interfaces según	
una determinada tecnología .....	228
4.5.3. Factory method para abstraernos	
de la implementación .....	230

## XII | Contenido

4.5.4. Factory method dinámico .....	231
4.6 Poniendo todo junto a trabajar .....	233
4.6.1. Unificar las clases utilitarias de acceso a datos .....	233
4.6.2. Ejemplo completo .....	233
4.7. Autoevaluación y ejercicios .....	234
4.8 Resumen.....	234

### CAPÍTULO 5

<b>Diseño de aplicaciones Java</b> .....	237
5.1 Introducción.....	237
5.2 Arquitectura de una aplicación Java....	238
5.2.1. Atributos no funcionales.....	238
5.2.2. Backend y frontend.....	239
5.2.3. Capas lógicas y físicas de la aplicación .....	240
5.3 Análisis y desarrollo de una aplicación completa.....	242
5.3.1. Façade y DAO .....	242
5.3.2. Cliente.....	244
5.4. Autoevaluación y ejercicios .....	246
5.5 Resumen.....	246

### CAPÍTULO 6

<b>Introspección de clases y objetos</b> .....	247
6.1 Introducción.....	247
6.2 Comenzando a introspección .....	249
6.2.1. Clase Class.....	249
6.2.1.1. Acceder al nombre de la clase ....	250
6.2.1.2. Acceder a los campos de la clase (variables de instancia y variables de clase) .....	251
6.2.1.3. Acceder a los métodos .....	251
6.2.1.4. Acceder a los constructores .....	254
6.2.2. Crear objetos dinámicamente	255
6.2.3. Invocar métodos dinámicamente....	256
6.3 JavaBeans.....	256
6.3.1. Qué son los bean .....	256
6.3.2. Instrospectando beans .....	257
6.3.2.1. Conocer los atributos de un vean	257
6.3.2.2. Invocar dinámicamente a los métodos de acceso.....	259
6.4 Annotations.....	261
6.5. Autoevaluación y ejercicios .....	264
6.6 Resumen.....	264

### CAPÍTULO 7

<b>Generalizaciones y desarrollo de frameworks</b> .....	265
7.1 Introducción .....	265
7.1.1. ¿Qué es un framework? .....	265
7.1.2. ¿Frameworks propios o frameworks de terceros? .....	266
7.1.3. Sobre este capítulo.....	268
7.2 Framework de persistencia basado en ORM .....	269
7.2.1. Representación del modelo de datos relacional mediante un modelo de objetos	269
7.2.2. Funcionalidad y modo de uso .....	271
7.2.3. Desarrollando el método find.....	272
7.2.4. Desarrollando el método findAll.....	278
7.2.5. Otros métodos de la clase MyHibernate .....	279
7.2.5.1. Método findBy.....	279
7.2.5.2. Método insert.....	281
7.2.5.3. Método update .....	281
7.2.5.4. Método delete.....	282
7.3 Framework de inyección de dependencias .....	282
7.3.1. Funcionalidad y modo de uso .....	283
7.3.2. Desarrollo del framework .....	286
7.4 Poniendo a trabajar todos juntos.....	287
7.5. Autoevaluación y ejercicios.....	290
7.6 Resumen .....	264

### CAPÍTULO 8

<b>Streams: flujos de entrada y salida de datos</b> .....	291
8.1 Introducción .....	291
8.2 Entrada y salida estándar .....	292
8.2.1. System.out, System.in y System.err	292
8.2.2. Redireccionar la entrada, la salida y la salida de error estándar .....	292
8.3 Archivos .....	294
8.3.1. Escribir un archivo .....	295
8.3.2. Leer un archivo .....	295
8.3.3. Archivos de acceso aleatorio.....	296
8.3.4. Las clases java.io.File y java.nio.Files .....	297
8.4 Serialización de objetos.....	298
8.4.1. Escribir objetos en un archivo .....	299
8.4.2. Leer objetos desde un archivo .....	300
8.5 Readers y writers .....	300
8.6. Autoevaluación y ejercicios.....	301
8.7 Resumen .....	301

CAPÍTULO 9	
<b>Networking</b> .....	303
9.1 Introducción .....	303
9.2 Conceptos básicos de networking .....	303
9.2.1. TCP - "Transmission Control Protocol" .....	304
9.2.2. UDP - "User Datagram Protocol" .....	304
9.2.3. Puertos .....	305
9.2.4. Dirección IP .....	305
9.2.5. Aplicaciones cliente/servidor .....	305
9.3. TCP .....	306
9.3.1. Socket .....	306
9.3.2. Un simple cliente/servidor en Java .	306
9.3.2.1. El server .....	306
9.3.2.2. El cliente .....	309
9.4. UDP .....	310
9.5. Autoevaluación y ejercicios .....	312
9.6 Resumen .....	313
CAPÍTULO 10	
<b>Threads: Multiprogramación</b> .....	315
10.1 Introducción .....	315
10.2 programar con threa .....	316
10.2.1. La interface runnable .....	318
10.2.2. Esperar hasta que finalice un thread .....	319
10.2.3. Ciclo de vida de un thread .....	320
10.2.4. prioridad de ejecución .....	322
10.3 Sincronización de threads .....	323
10.3.1. Monitores y sección crítica .....	323
10.3.2. Productor / consumidor .....	324
10.4 Autoevaluación y ejercicios .....	328
10.5 Resumen .....	329
CAPÍTULO 11	
<b>Poniendo todo junto a trabaja</b> .....	331
11.1 Introducción .....	331
11.2 MySpringBoot .....	332
11.2.1. Server .....	333
11.2.1.1. Componentes y servicios .....	333
11.2.1.2. Ejecutar el servidor .....	336
11.2.2. Cliente .....	337
11.2.3. Request y Response .....	339
11.3 Exponer los servicios del backend de una aplicación empresarial .....	340
11.3.1. Server .....	341
11.3.2. Cliente .....	343
11.3.2.1. Business Delegate .....	344
11.3.2.2. La aplicación .....	345
11.4 Desarrollo de MySpringBoot .....	347
11.4.1. Server .....	347
11.4.1.1. registerComponent .....	348
11.4.1.2. runServer .....	351
11.4.2. Cliente .....	354
11.5. Autoevaluación y ejercicios .....	357
11.5 Resumen .....	357