

ÍNDICE

EL AUTOR	17
INTRODUCCIÓN	19
CAPÍTULO 1. INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN	
JAVA	21
1.1 COMENCEMOS A PROGRAMAR	22
1.1.1 El Entorno Integrado de Desarrollo (IDE).....	23
1.2 ESTRUCTURAS DE CONTROL	24
1.2.1 Entrada y salida de datos por consola	24
1.2.2 Definición de variables	25
1.2.3 Comentarios en el código.....	25
1.2.4 Estructuras de decisión.....	27
1.2.5 Estructuras iterativas	32
1.3 OTROS ELEMENTOS DEL LENGUAJE	35
1.3.1 Tipos de datos.....	35
1.3.2 Algunas similitudes y diferencias con C y C++.....	36
1.3.3 Definición de constantes	37
1.3.4 Arrays	39
1.3.5 Matrices.....	42
1.3.6 Literales de cadenas de caracteres.....	45
1.3.7 Caracteres especiales.....	48
1.3.8 Argumentos en línea de comandos	49
1.4 TRATAMIENTO DE CADENAS DE CARACTERES	50
1.4.1 Acceso a los caracteres de un String	50

1.4.2	Mayúsculas y minúsculas.....	52
1.4.3	Ocurrencias de caracteres.....	52
1.4.4	Subcadenas.....	53
1.4.5	Prefijos y sufijos.....	54
1.4.6	Posición de un substring dentro de la cadena.....	55
1.4.7	Concatenar cadenas.....	55
1.4.8	La clase StringBuffer.....	56
1.4.9	Conversión entre números y cadenas.....	58
1.4.10	Representación numérica en diferentes bases.....	60
1.4.11	La clase StringTokenizer.....	61
1.4.12	Usar expresiones regulares para particionar una cadena.....	63
1.4.13	Comparación de cadenas.....	64
1.5	OPERADORES.....	66
1.5.1	Operadores aritméticos.....	67
1.5.2	Operadores lógicos.....	67
1.5.3	Operadores relacionales.....	68
1.5.4	Operadores lógicos de bit.....	68
1.5.5	Operadores de desplazamiento de bit.....	68
1.6	LA MÁQUINA VIRTUAL Y EL JDK.....	69
1.6.1	El JDK (Java Development Kit).....	69
1.6.2	Versiones y evolución del lenguaje Java.....	70
1.7	QUÉ VIMOS Y QUÉ VIENE.....	70
CAPÍTULO 2. PROGRAMACIÓN ORIENTADA A OBJETOS.....		71
2.1	CLASES Y OBJETOS.....	72
2.1.1	Los métodos.....	73
2.1.2	Herencia y sobreescritura de métodos.....	76
2.1.3	El método toString.....	76
2.1.4	El método equals.....	78
2.1.5	Definir y “crear” objetos.....	79
2.1.6	El constructor.....	80
2.1.7	Un pequeño repaso de lo visto hasta aquí.....	82
2.1.8	Convenciones de nomenclatura.....	84
2.1.9	Sobrecarga.....	85
2.1.10	Encapsulamiento.....	89
2.1.11	Visibilidad de métodos y atributos.....	93
2.1.12	Packages (paquetes).....	95
2.1.13	La estructura de los paquetes y la variable CLASSPATH.....	95

2.1.14 Las APIs (“Application Programming Interface”).....	97
2.1.15 Representación gráfica UML	97
2.1.16 Importar clases de otros paquetes	99
2.2 HERENCIA Y POLIMORFISMO	99
2.2.1 Polimorfismo	103
2.2.2 Constructores de subclases.....	106
2.2.3 La referencia “super”	106
2.2.4 La referencia “this”	110
2.2.5 Clases abstractas.....	111
2.2.6 Constructores de clases abstractas	116
2.2.7 Instancias.....	120
2.2.8 Variables de instancia.....	122
2.2.9 Variables de la clase.....	124
2.2.10 El Garbage Collector (recolector de residuos).....	125
2.2.11 El método finalize	126
2.2.12 Constantes	127
2.2.13 Métodos de la clase	127
2.2.14 Clases utilitarias	130
2.2.15 Referencias estáticas	131
2.2.16 Colecciones (primera parte)	132
2.2.17 Clases genéricas	139
2.2.18 Implementación de una pila (estructura de datos).....	143
2.2.19 Implementación de una cola (estructura de datos).....	145
2.3 INTERFACES	146
2.3.1 Desacoplamiento de clases.....	149
2.3.2 El patrón de diseño factory method	152
2.3.3 Abstracción a través de interfaces.....	152
2.3.4 La interface comparable.....	153
2.3.5 Desacoplar aún más.....	159
2.4 COLECCIONES.....	163
2.4.1 Cambio de implementación.....	167
2.5 EXCEPCIONES	168
2.5.1 Excepciones declarativas y no declarativas	172
2.5.2 El bloque try-catch-finally.....	175
2.6 QUÉ VIMOS Y QUÉ VIENE	177
CAPÍTULO 3. ACCESO A BASES DE DATOS (JDBC).....	179
3.1 CONCEPTOS BÁSICOS SOBRE BASES DE DATOS RELACIONALES	179

3.1.1 Relaciones foráneas y consistencia de datos	181
3.1.2 Diagrama Entidad-Relación (DER)	182
3.1.3 SQL – Structured Query Language	183
3.1.4 Ejecutar sentencias query	183
3.1.5 Unir tablas (<i>join</i>)	184
3.1.6 Ejecutar sentencias update	185
3.2 CONECTAR PROGRAMAS JAVA CON BASES DE DATOS	187
3.2.1 Invocar un query con un join	193
3.2.2 Updates	194
3.2.3 Ejecutar un INSERT	194
3.2.4 Ejecutar un DELETE	196
3.2.5 Ejecutar un UPDATE	197
3.2.6 El patrón de diseño “Singleton” (Singleton Pattern)	197
3.2.7 Singleton Pattern para obtener la conexión	199
3.2.8 El shutdown hook	201
3.2.9 Inner classes (clases internas)	201
3.2.10 Manejo de transacciones	202
3.3 USO AVANZADO DE JDBC	205
3.3.1 Acceso a la metadata del resultSet (ResultSetMetaData)	205
3.3.2 Definir el “Query Fetch Size” para conjuntos de resultados grandes	208
3.3.3 Ejecutar batch updates (procesamiento por lotes)	208
3.4 QUÉ VIMOS Y QUÉ VIENE	210
CAPÍTULO 4. DISEÑO DE APLICACIONES JAVA (PARTE I)	211
4.1 ATRIBUTOS DE UNA APLICACIÓN	211
4.1.1 Casos de uso	212
4.2 DESARROLLO DE APLICACIONES EN CAPAS	213
4.2.1 Aplicación de estudio	215
4.2.2 Análisis de los objetos de acceso a datos (DAOs y DTOs)	216
4.2.3 Análisis del façade	221
4.2.4 Diagrama de secuencias de UML	224
4.3 PORTABILIDAD ENTRE DIFERENTES BASES DE DATOS	226
4.3.1 DAOs abstractos e implementaciones específicas para las diferentes bases de datos	228
4.3.2 Implementación de un factory method	232
4.3.3 Combinar el factory method con el singleton pattern	234
4.3.4 Mejorar el diseño de los DAOs abstractos	236
4.4 DISEÑO POR CONTRATOS	238

4.4.1 Coordinación de trabajo en equipo	239
4.5 QUÉ VIMOS Y QUÉ VIENE	242
CAPÍTULO 5. INTERFAZ GRÁFICA (GUI).....	243
5.1 COMPONENTES Y CONTENEDORES.....	244
5.1.1 Distribución de componentes (layouts).....	245
5.1.2 AWT y Swing.....	245
5.2 COMENZANDO A DESARROLLAR <i>GUI</i>	246
5.2.1 Distribuciones relativas	246
5.2.2 FlowLayout	247
5.2.3 BorderLayout	250
5.2.4 GridLayout	252
5.2.5 Combinación de layouts	253
5.3 CAPTURAR EVENTOS	261
5.3.1 Tipos de eventos.....	266
5.3.2 Eventos de acción.....	268
5.3.3 Eventos de teclado.....	271
5.4 SWING	274
5.4.1 Cambiar el look and feel	279
5.5 MODEL VIEW CONTROLLER (MVC).....	281
5.5.1 Ejemplo de uso: ListModel	281
5.5.2 Ejemplo de uso: TableModel	285
5.6 QUÉ VIMOS Y QUÉ VIENE	288
CAPÍTULO 6. MULTITHREADING (HILOS).....	289
6.1 IMPLEMENTAR THREADS EN JAVA	291
6.1.1 La interface Runnable	293
6.1.2 Esperar a que finalice un thread	294
6.1.3 Threads y la interfaz gráfica.....	295
6.1.4 Sistemas operativos multitarea.....	298
6.1.5 Ciclo de vida de un thread.....	299
6.1.6 Prioridad de ejecución.....	302
6.1.7 Sincronización de threads	303
6.1.8 Ejemplo del Productor/Consumidor.....	304
6.1.9 Qué vimos y qué viene	310
CAPÍTULO 7. NETWORKING.....	311
7.1 CONCEPTOS BÁSICOS DE NETWORKING	311
7.1.1 TCP – “Transmission Control Protocol”	312

7.1.2 UDP – “User Datagram Protocol”	312
7.1.3 Puertos.....	312
7.1.4 Dirección IP.....	313
7.1.5 Aplicaciones cliente/servidor	313
7.2 TCP EN JAVA	313
7.2.1 El socket	313
7.2.2 Un simple cliente/servidor en Java	313
7.2.3 Serialización de objetos.....	318
7.2.4 Implementación de un servidor multithread	318
7.2.5 Enviar y recibir bytes	321
7.2.6 Enviar y recibir valores de tipos de datos primitivos.....	325
7.3 UDP EN JAVA	325
7.4 REMOTE METHOD INVOCATION (RMI)	328
7.4.1 Componentes de una aplicación RMI	329
7.4.2 Ejemplo de una aplicación que utiliza RMI.....	330
7.4.3 Compilar y ejecutar la aplicación RMI	333
7.4.4 RMI y serialización de objetos.....	334
7.5 QUÉ VIMOS Y QUÉ VIENE	334
CAPÍTULO 8. DISEÑO DE APLICACIONES (PARTE II).....	335
8.1 REPASO DE LA APLICACIÓN DE ESTUDIO.....	335
8.2 CAPAS LÓGICAS VS. CAPAS FÍSICAS	338
8.3 DESVENTAJAS DE UN MODELO BASADO EN 2 CAPAS FÍSICAS	339
8.4 DESARROLLO DE LA APLICACIÓN EN TRES CAPAS FÍSICAS	341
8.4.1 Desarrollo del servidor.....	341
8.4.2 Desarrollo de un cliente de prueba.....	347
8.4.3 El service locator (o ubicador de servicios)	349
8.4.4 Integración con la capa de presentación.....	356
8.5 IMPLEMENTACIÓN DEL SERVIDOR CON TECNOLOGÍA RMI.....	360
8.5.1 El servidor RMI.....	360
8.5.2 El ServiceLocator y los objetos distribuidos.....	362
8.5.3 Desarrollo de un cliente de prueba.....	363
8.5.4 Integración con la capa de presentación.....	364
8.5.5 El bussiness delegate.....	367
8.6 CONCURRENCIA Y ACCESO A LA BASE DE DATOS	368
8.6.1 El pool de conexiones	369
8.6.2 Implementación de un pool de conexiones	369
8.6.3 Integración con los servidores TCP y RMI.....	375

8.7 QUÉ VIMOS Y QUÉ VIENE	377
CAPÍTULO 9. ESTRUCTURAS DE DATOS DINÁMICAS.....	379
9.1 ESTRUCTURAS DINÁMICAS.....	380
9.1.1 El nodo	380
9.1.2 Lista enlazada (LinkedList).....	381
9.1.3 Pilas (Stack).....	387
9.1.4 Colas (Queue).....	388
9.1.5 Implementación de una cola sobre una lista circular	389
9.1.6 Clases Java que implementan estructuras de datos	392
9.1.7 Tablas de dispersión (Hashtable)	393
9.1.8 Estructuras de datos combinadas	396
9.1.9 Árboles	399
9.1.10 Árbol binario de búsqueda	400
9.1.11 La clase TreeSet	401
9.2 QUÉ VIMOS Y QUÉ VIENE	402
CAPÍTULO 10. PARAMETRIZACIÓN MEDIANTE ARCHIVOS XML.....	403
10.1 XML – “EXTENSIBLE MARKUP LANGUAGE”	404
10.2 ESTRUCTURAR Y DEFINIR PARÁMETROS EN UN ARCHIVO XML	406
10.2.1 Definición de la estructura de parámetros.....	406
10.2.2 Leer y parsear el contenido de un archivo XML.....	409
10.2.3 Acceder a la información contenida en el archivo XML	412
10.3 QUÉ VIMOS Y QUÉ VIENE	423
CAPÍTULO 11. INTROSPECCIÓN DE CLASES Y OBJETOS	425
11.1 IDENTIFICAR MÉTODOS Y CONSTRUCTORES.....	427
11.2 ACCESO AL PROTOTIPO DE UN MÉTODO	429
11.3 QUÉ VIMOS Y QUÉ VIENE	433
CAPÍTULO 12. GENERALIZACIONES Y DESARROLLOS DE FRAMEWORKS.....	435
12.1 ¿QUÉ ES UN FRAMEWORK?.....	436
12.1.1 ¿Frameworks propios o frameworks de terceros?.....	437
12.1.2 Reinventar la rueda.....	437
12.2 UN FRAMEWORK PARA ACCEDER A ARCHIVOS XML	439
12.2.1 Diseño de la API del framework	441
12.2.2 Análisis del elemento a generalizar.....	443
12.2.3 Parsear el archivo XML y cargar la estructura de datos	444
12.3 UN FRAMEWORK PARA ACCEDER A BASES DE DATOS.....	452

12.3.1 Identificación de la tarea repetitiva	454
12.3.2 Diseño de la API del framework	455
12.3.3 Java Beans	457
12.3.4 Transacciones	470
12.4 EL BEAN FACTORY	476
12.5 INTEGRACIÓN	478
12.5.1 Los objetos de acceso a datos	479
12.5.2 El facade	481
12.5.3 El archivo de configuración	482
12.5.4 El cliente	483
12.6 QUÉ VIMOS Y QUÉ VIENE	484
CAPÍTULO 13. ENTRADA/SALIDA	485
13.1 I/O STREAMS (FLUJOS DE ENTRADA Y SALIDA)	485
13.1.1 Entrada y salida estándar	485
13.1.2 Redireccionar la entrada y salidas estándar	487
13.1.3 Cerrar correctamente los streams	489
13.1.4 Streams de bytes (InputStream y OutputStream)	490
13.1.5 Streams de caracteres (readers y writers)	491
13.1.6 Streams bufferizados	492
13.1.7 Streams de datos (DataInputStream y DataOutputStream)	493
13.1.8 Streams de objetos (ObjectInputStream y ObjectOutputStream)	496
13.2 QUÉ VIMOS Y QUÉ VIENE	498
CAPÍTULO 14. CONSIDERACIONES FINALES	499
14.1 CONSIDERACIONES SOBRE MULTITHREADING Y CONCURRENCIA	499
14.1.1 Clases con o sin métodos sincronizados	499
14.1.2 El singleton pattern en contextos multithreaded	500
14.2 CONSIDERACIONES SOBRE CLASES “LEGACY”	502
14.2.1 La clase StringTokenizer y el método split	502
APÉNDICE A. NOCIONES BÁSICAS DE PROGRAMACIÓN	503
A.1 EL LENGUAJE DE PROGRAMACIÓN	503
A.2 EL COMPILADOR	504
A.3 LOS INTÉRPRETES	504
A.4 LAS MÁQUINAS VIRTUALES	504
A.5 JAVA Y SU MÁQUINA VIRTUAL	505
A.6 RECURSOS DE PROGRAMACIÓN	505
A.6.1 Las variables	505

A.6.2 Tipos de datos.....	506
A.6.3 Operadores aritméticos.....	507
A.6.4 Estructuras de decisión.....	507
A.6.5 Estructuras de repetición	508
APÉNDICE B. APPLETS	511
B.1 COMENZANDO CON APPLETS	512
B.1.1 El ciclo de vida de un applet.....	514
B.1.2 El contexto del applet	515
B.1.3 Pasarle parámetros a un applet	516
MATERIAL ADICIONAL	517