

Programación orientada a objetos con Java y UML

MAURICIO SERGIO ORTIZ OCHOA

ANDREA MARICELA PLAZA CORDERO

CONTRAPORTADA

Uno de los estándares de programación es el paradigma de la Programación Orientada a Objetos que modela situaciones del mundo real en forma de objetos con características y funcionalidades. El éxito de este paradigma consiste en la aceptación de los lenguajes, tanto en diseño como en el soporte para su implementación.

En este libro, se aborda la Programación Orientada a Objetos amalgamando dos secciones importantes. Por un lado, se utiliza el Lenguaje Unificado de Modelado (UML) para describir objetos; y por otro, se presenta el lenguaje de programación denominado Java para la implementación de los modelos presentados.

Las diferentes concepciones teóricas son tratadas de forma práctica mediante ejercicios resueltos y modelos de objetos que permiten al lector ir creciendo en sus conocimientos de manera estructurada, ya que partiendo desde conceptos generales de algoritmos es posible tratar nociones abstractas de las propiedades inherentes a los objetos.

Finalmente, y no menos importante, es el hecho de que este libro fue concebido como una guía práctica para que las personas interesadas en la programación puedan determinar claramente cuáles son los principios fundamentales y, por ende, acceder a los conocimientos básicos sobre la Programación Orientada a Objetos.

PRESENTACIÓN

Bienvenidos, lectores, al libro *Programación orientada a objetos con Java y UML*, cuyo objetivo es plasmar la lógica de programación con su respectivo lenguaje, en nuestro caso, Java.

Los programadores de épocas pasadas aprendieron la programación estructurada basada en sentencias consecutivas, las mismas que no se alejan del todo al paradigma de la Programación Orientada a Objetos, puesto que algunos registros usan esta lógica. El paradigma de la Programación Orientada a Objetos es una práctica que permite la reutilización de código, además de flexibilidad, representatividad, entre otros aspectos. En este libro se presenta tanto la creación de objetos como su manipulación y correcta utilización.

Cada capítulo está dividido en secciones que ayudan a desarrollar de forma organizada el estudio de los temas. A continuación se detallan cada una de las secciones:

- **Introducción:** Se refiere a una explicación concreta y específica sobre el tema a tratarse en el capítulo.
- **Objetivos:** Se presentan los objetivos de aprendizaje de cada capítulo, teniendo como propósito medir la comprensión de los puntos más importantes.
- **Prerrequisitos:** A pesar de la organización secuencial, se ha visto la necesidad de añadir los prerrequisitos o temas de comprensión previa y necesaria antes de comenzar con el estudio de cada capítulo.
- Cada capítulo provee un conjunto de ejemplos desarrollados mediante el Kit de Desarrollo de Java Standard Edition (JDK) 6.
- Se presenta una gran cantidad de ilustraciones, cuadros, tablas, diagramas de flujo, entre otros.
- Los ejercicios servirán para verificar los conocimientos adquiridos.
- Para finalizar, se sistematizan los puntos clave a modo de resumen de los conceptos más importantes.

El libro tiene como objetivo enseñar y explicar la programación, usando ejemplos sencillos, los cuales se pueden realizar en el aula de clases. Los principales objetivos son:

- Presentar en cada capítulo la parte teórica y, posteriormente, ejemplos prácticos.
- Abordar a la programación desde su lógica, presentando la metodología y prácticas de aprendizaje.
- Se presenta un conjunto de ejemplos según diferentes grados de dificultad, tanto para ser presentados en el aula de clases, en las que el tiempo es reducido, como aquellos que puedan ser abordados con un tiempo de dedicación mayor.
- Los capítulos fueron cuidadosamente elaborados tratando de respetar su secuencia para que el lector no esté expuesto a temas que no se han tratado o explicado con anterioridad.
- Cada capítulo ofrece temas tratados con puntualidad y concreción para que, de esta manera, mantener el interés del lector y evitar confusiones.

El texto ha sido revisado por distinguidos catedráticos universitarios que trabajaron conjuntamente con nosotros para que *Programación orientada a objetos con Java y UML* brinde un aprendizaje adecuado y pedagógico tanto a estudiantes como a profesionales que deseen no sólo aprender a programar usando un lenguaje orientado a objetos sino, además, aspiren a mejorar su lógica de programación.

El Lenguaje Unificado de Modelado (UML) es usado a nivel mundial para el diseño de software orientado a objetos. Por ser tan complejo el diseño se provee de trece diagramas y en el Capítulo 7 (“Modelación de Objetos con UML”) se accede a todo lo referente al lenguaje unificado modelado mediante un conjunto de ejercicios.

Esta investigación ha tomado como base el libro *Java (TM) how to program* para el capítulo 1 y 2 (“Lógica de programación”). Para los posteriores, nos basamos en *Object Oriented Programming, especialmente UML y patrones*, entre otras fuentes.

Índice general

1. Algoritmos	17
1.1. Definición	18
1.1.1. Características	20
1.2. Proceso	20
1.2.1. Análisis del Problema	21
1.2.2. Crear el Algoritmo	21
1.2.3. Prueba de Escritorio	21
1.3. Formas de representación	21
1.3.1. Gráficos	22
1.3.2. No Gráficos	23
1.4. Estructura secuencial	23
1.4.1. Representación de la Estructura	23
1.4.2. Funcionamiento de la Estructura	24
1.5. Estructura condicional	35
1.5.1. Estructura condicional Simple	36
1.5.2. Estructura condicional Doble	36
1.5.3. Estructura condicional Multialternativa	36
1.5.4. Representación de la Estructura	36
1.5.5. Funcionamiento de la Estructura	38
1.6. Estructura repetitiva	43
1.6.1. Mientras que	44
1.6.2. Repetir mientras	45
1.6.3. Desde hasta	46
2. Programación Orientada a Objetos (POO)	67
2.1. Introducción a la Programación Orientada a Objetos	68
2.1.1. Ventajas de la Programación Orientada a Objetos	69
2.1.2. Propiedades de la Programación Orientada a Objetos	70
2.1.3. Limitaciones de la Programación Orientada a Objetos	71
2.2. Clases y Objetos	72
2.3. Atributos	73
2.4. Métodos	74
3. Fundamentos de Programación con Java	77
3.1. Introducción al lenguaje	78
3.1.1. Historia de Java	78
3.1.2. Características del Lenguaje	80

3.2.	Estructura Léxica	81
3.2.1.	Comentarios	81
3.2.2.	Identificadores	81
3.2.3.	Palabras reservadas	82
3.2.4.	Separadores	82
3.3.	Operadores	83
3.3.1.	Aritméticos	83
3.3.2.	Asignación	83
3.3.3.	Condicionales	84
3.3.4.	Lógicos	84
3.4.	Variables y Tipos de datos	85
3.5.	Ejecución de un programa	86
3.5.1.	Descarga de fuentes e instalación	87
3.5.2.	Fases de un programa Java	88
3.6.	Estructuras condicionales	89
3.6.1.	Estructura if	90
3.6.2.	Estructura if ... else	90
3.6.3.	Estructura switch	91
3.7.	Estructuras repetitivas	92
3.7.1.	Estructura while	93
3.7.2.	Estructura do...while	94
3.7.3.	Estructura for	94
3.8.	Clases para el manejo de cadenas y números	95
3.8.1.	Clase Scanner	95
3.8.2.	Clase String	96
3.8.3.	Clase Math	97
4.	Creación de clases y objetos	103
4.1.	Clases	104
4.2.	Atributos	104
4.3.	Métodos	105
4.4.	Objetos	108
4.5.	Constructores	109
4.6.	Visibilidad	110
5.	Parametrización de métodos	115
5.1.	Parámetros de entrada	115
5.2.	Parámetros de salida	117
5.3.	Encapsulamiento	124
6.	Arreglos	127
6.1.	Arreglos unidimensionales	128
6.1.1.	Creación	128
6.1.2.	Inicialización	128
6.1.3.	Asignación de valores	129
6.1.4.	Recuperación	130
6.2.	Arreglos bidimensionales	134
6.2.1.	Creación	134
6.2.2.	Inicialización	135

6.2.3. Asignación de valores	136
6.2.4. Recuperación	137
6.2.5. Limitaciones	141
7. Modelación de objetos con UML	145
7.1. La necesidad de diseñar de Software	146
7.2. Diagramas UML	146
7.2.1. Diagramas de Comportamiento	147
7.2.2. Diagramas de Interacción	147
7.2.3. Diagramas Estructurales	147
7.3. Diagrama de Casos de Uso	148
7.3.1. Componentes	148
7.3.2. Caso de Estudio	149
7.4. Diagrama de Clases	151
7.4.1. Clases	151
7.4.2. Relaciones	153
7.4.3. Relaciones de Conocimiento	155
7.5. Casos de Estudio	157
7.5.1. Tienda de Productos	157
7.5.2. Seguimiento de Tesis	159
8. Arreglos de objetos y colecciones	165
8.1. Arreglos de objetos	166
8.2. Colecciones y Diccionarios	168
8.2.1. Colecciones	168
8.2.2. Diccionarios	171
9. Paquetes y Documentación	177
9.1. Paquetes	178
9.1.1. Sentencia import	182
9.1.2. Ventajas del uso de paquetes:	185
9.2. Distribución de aplicaciones mediante de JARs	185
9.3. Documentar con JavaDoc	187
9.3.1. Sintaxis	187
9.3.2. Reglas	189
9.3.3. Elementos de Documentación	189
9.3.4. Doclet y Taglet API	190
9.3.5. Generación de javaDOC	190
10. Herencia	199
10.1. Concepto de herencia	199
10.2. Modificadores de acceso	202
10.3. Interfaces	204
10.4. Clases abstractas	206
10.5. Clases finales	208
10.5.1. Métodos Finales	209
10.6. Conversión de tipos de dato	210
10.6.1. Clases envoltorio:	210
10.6.2. Operador cast	211

11. Polimorfismo	215
11.1. Polimorfismo	216
11.1.1. Beneficios	216
11.1.2. Tipos de polimorfismo:	216
11.2. Sobrecarga	225
11.2.1. Sobrecarga de constructores	225
11.2.2. Sobrecarga de métodos	227
11.3. Redefinición y refinamiento	229
12. Manejo de Errores	235
12.1. Manejo de Excepciones en Java	236
12.2. Tipos de Excepciones	238
12.3. Lanzamiento de Excepciones	240
12.3.1. Personalización de errores	242
13. Elaboración de interfaces gráficas	249
13.1. Interfaces Gráficas de Usuario	250
13.2. Contenedores	250
13.2.1. JFrame	251
13.2.2. JPanel	252
13.2.3. JInternalFrame	255
13.3. Gestores de Distribución	258
13.3.1. FlowLayout	260
13.3.2. BorderLayout	262
13.3.3. GridLayout	265
13.3.4. BoxLayout	267
13.3.5. GridBagLayout	270
13.4. Componentes básicos	275
13.4.1. JLabel	275
13.4.2. JButton	278
13.4.3. JToggleButton	281
13.4.4. JComboBox	287
13.4.5. JList	290
13.4.6. JTextField	293
13.4.7. JTextArea	297
13.5. Otros componentes	301
13.5.1. Menús	301
13.5.2. JSlider	307
13.6. Eventos	309
13.6.1. Funcionamiento del Manejador de Eventos	311
13.6.2. Eventos con el Ratón	312
13.6.3. Eventos con el Teclado	317
14. Manejo de archivos	325
14.1. Conceptos generales sobre archivos	326
14.1.1. Archivo	326
14.1.2. Tipos de archivos en Java	327
14.1.3. Streams	328
14.1.4. Excepciones	329

14.2. Archivos de texto	329
14.2.1. Escritura de archivos de texto	330
14.2.2. Lectura de archivos de texto	331
14.3. Archivos binarios	332
14.3.1. Escritura de archivos binarios	333
14.3.2. Lectura de archivos binarios	334
14.3.3. Archivos de acceso aleatorio	335
14.4. Archivos de objetos	338
14.4.1. Serialización	338
14.4.2. Escritura de archivos de objetos	340
14.4.3. Lectura de archivos de objetos	341
15. Organización de archivos e índices	345
15.1. Índices primarios	346
15.1.1. Índices densos	346
15.1.2. Índices dispersos	348
15.1.3. Diferencias entre Índices densos y dispersos	350
15.2. Índices secundarios	351
15.3. Índices multinivel	351
15.4. Árboles	352
15.4.1. Árbol Binario (B) o de grado 2	358
15.4.2. Árboles Binarios de Búsqueda	360
15.4.3. Árbol Balanceado o B+	367
16. Programación específica	371
16.1. Introducción a los hilos	372
16.1.1. Ciclo de Vida de un Hilo	373
16.1.2. Métodos	373
16.1.3. Estados de los Hilos	374
16.1.4. Control de Hilos	375
16.1.5. Instancias para crear Hilos	376
16.1.6. Prioridades	381
16.1.7. Sincronización	382
16.1.8. ArrayBlockingQueue	387
16.2. Introducción al Timer	390
16.3. Manejo del Puerto USB	393
16.3.1. Introducción al JSR 80	394
16.3.2. jPicUSB	396
16.3.3. Usando jPicUSB	398
17. Programación genérica	403
17.1. Clases genéricas	404
17.1.1. Restricciones de Tipo	406
17.1.2. Constructores genéricos	407
17.1.3. Métodos genéricos	408
17.2. Colecciones genéricas	412
17.3. Internacionalización	415
Bibliografía	421

Índice de figuras

1.4.1.	Representación de la estructura secuencial	24
1.4.2.	Instrucción de asignación	24
1.4.3.	Diagrama de flujo de la instrucción de asignación	27
1.4.4.	Instrucción de entrada	27
1.4.5.	Diagrama de flujo de la instrucción de entrada	28
1.4.6.	Instrucción de salida	29
1.4.7.	Diagrama de flujo de la instrucción de salida	31
1.4.8.	Diagrama de flujo del ejemplo del perímetro del rectángulo	32
1.4.9.	Diagrama de flujo del ejemplo de la edad	34
1.5.1.	Estructura condicional simple	37
1.5.2.	Estructura condicional doble	38
1.5.3.	Diagrama de flujo del ejemplo si pasa o pierde el año	40
1.5.4.	Diagrama de flujo del ejemplo si es mayor, menor o igual	42
1.6.1.	Estructura repetitiva mientras-que	44
1.6.2.	Estructura repetitiva repetir-mientras	45
1.6.3.	Estructura repetitiva desde-hasta	46
1.6.4.	Diagrama de flujo del ejemplo de la sumatoria	48
1.6.5.	Diagrama de flujo del ejemplo de sumatoria de n números	52
1.6.6.	Diagrama de flujo del ejemplo sucesión de n números	56
1.6.7.	Diagrama de flujo del ejemplo del factorial	60
2.2.1.	Estructura de la clase	72
2.2.2.	Representación de una clase y objeto	73
2.3.1.	Representación de los atributos	74
2.4.1.	Representación de los métodos	75
4.3.1.	Clase TrianguloRectangulo	106
5.1.1.	Clase área (Parámetros de Entrada)	116
5.2.1.	Clase área (Parámetros de Salida)	118
5.2.2.	Clase Serie y Datos	119
5.2.3.	Clase Proceso	121
7.3.1.	Componentes de un diagrama de casos de uso	149
7.3.2.	Reserva de instalaciones	150
7.3.3.	Registro de socios	151
7.4.1.	Clase Persona	152
7.4.2.	Diagrama de clases	154

7.4.3.	Agregación	156
7.4.4.	Composición	156
7.5.1.	Venta de productos	157
7.5.2.	Registro de datos	158
7.5.3.	Tienda de productos	158
7.5.4.	Área de conocimiento	160
7.5.5.	Docentes	161
7.5.6.	Estudiantes	161
7.5.7.	Seguimiento de Tesis	162
8.2.1.	Diagrama de clases: registro de Auto	169
9.1.1.	Forma de representación de los paquetes	179
9.1.2.	Ejemplo del uso de paquetes	179
9.1.3.	Paquetes cíclicos	180
9.1.4.	Creación de un sistema haciendo uso de los paquetes	181
9.3.1.	JavaDoc de la clase inicio	192
9.3.2.	JavaDoc de la clase serie	194
9.3.3.	JavaDoc de la clase datos	195
10.1.1.	Jerarquía de herencia	200
10.2.1.	Herencia con modificadores protected	203
10.3.1.	Diagrama de clases con un interface	205
10.4.1.	Diagrama de clases con una clase abstracta	207
10.6.1.	Jerarquía de herencia	211
12.2.1.	Jerarquía de herencia de la clase Throwable	239
12.3.1.	Diagrama de clases de Calificacion	241
12.3.2.	Clase ExcepcionRango	243
13.2.1.	Contenedores	250
13.2.2.	Generación de una ventana	252
13.2.3.	Generación de paneles	254
13.2.4.	Generación de JInternalFrames	258
13.3.1.	Layout Manager de tipo NULL	259
13.3.2.	LayoutManager	260
13.3.3.	Alineación de FlowLayout	261
13.3.4.	FlowLayout	262
13.3.5.	Áreas del BorderLayout	263
13.3.6.	BorderLayout	265
13.3.7.	GridLayout	267
13.3.8.	BoxLayout	270
13.3.9.	Ubicación de componentes en GridBagLayout	270
13.3.10.	GridBagLayout	273
13.3.11.	GridBagLayout usando remainder y relative	275
13.4.1.	Componentes básicos	276
13.4.2.	Creación de JLabel	278
13.4.3.	Manejo y creación de JButton	281
13.4.4.	Manejo y creación de JCheckBox	284
13.4.5.	Manejo y creación de JRadioButton	287

13.4.6. Manejo y creación de JComboBox	290
13.4.7. Manejo y creación de JList Simple	293
13.4.8. Manejo y creación de JTextField	297
13.4.9. Manejo y creación de JTextArea	301
13.5.1. Manejo y creación de menús	306
13.5.2. Manejo y creación de JSlider	309
13.6.1. Eventos	310
13.6.2. Manejador de eventos o escucha	312
13.6.3. Eventos con el ratón	316
13.6.4. Eventos con el teclado	319
14.2.1. Jerarquía de clases para archivos de texto	330
14.3.1. Jerarquía de clases para archivos binarios	332
14.3.2. Jerarquía de clases para archivos aleatorios	335
15.1.1. Índice primario denso	347
15.1.2. Eliminar usando índice primario denso	347
15.1.3. Inserción usando índice primario denso	348
15.1.4. Índice primario disperso	349
15.1.5. Eliminar usando índice primario disperso	350
15.1.6. Inserción usando índice primario disperso	350
15.2.1. Índice secundario	351
15.4.1. Índice del capítulo 7	353
15.4.2. Formas de representación	354
15.4.3. Árbol propuesto	356
15.4.4. Formas de recorrer un árbol binario	357
15.4.5. Árbol binario	358
15.4.6. Árbol propuesto para ejercicio de recorrido	359
15.4.7. Árbol propuesto para ejercicio de eliminación	362
15.4.8. Árbol eliminando un nodo hoja	362
15.4.9. Árbol eliminando un nodo con 1 hijo	363
15.4.10.Árbol eliminando un nodo con 2 hijos	363
15.4.11.Nodo típico	368
15.4.12.Nodo hijo	368
17.1.1. Clase Genérica	405
17.1.2. Clase genérica Arreglo	408
17.2.1. Diagrama de clases: registro de autos	413

Índice de cuadros

1.1. Representación de los Diagramas de Flujo	22
1.2. Prueba de escritorio de la instrucción de asignación	26
1.3. Prueba de escritorio de la instrucción de entrada	28
1.4. Prueba de escritorio de la instrucción de salida	30
1.5. Prueba de escritorio del ejemplo del perímetro del rectángulo	33
1.6. Prueba de escritorio del ejemplo de la edad	35
1.7. Prueba de escritorio del ejemplo si pasa o pierde el año	41
1.8. Prueba de escritorio del ejemplo si es mayor, menor o igual	43
1.9. Prueba de escritorio del ejemplo de la sumatoria	50
1.10. Prueba de escritorio del ejemplo de sumatoria de n números	54
1.11. Prueba de escritorio del ejemplo sucesión de n números	58
1.12. Prueba de escritorio del ejemplo del factorial	62
3.1. Palabras reservadas	82
3.2. Separadores	83
3.3. Operadores aritméticos	83
3.4. Operadores de asignación	84
3.5. Operadores condicionales	84
3.6. Operadores lógicos	85
3.7. Tipos comunes de datos	85
3.8. Funcionamiento de la estructura while	93
3.9. Funcionamiento de la estructura do...while	94
3.10. Funcionamiento de la estructura for	95
3.11. Resumen de métodos de la clase Scanner	96
3.12. Resumen de métodos de la clase String	97
3.13. Resumen de métodos de la clase Math	98
4.1. Diferencias entre atributos y variables	105
6.1. Arreglos unidimensionales	128
6.2. Inicialización de tres tipos de arreglos	129
6.3. Asignación de valores en el arreglo tabla	129
6.4. Asignación de valores en el arreglo nombre	130
6.5. Inicialización y creación de arreglos	130
6.6. Recuperación de un valor en el arreglo tabla	130
6.7. Recuperación de valores en el arreglo nombre	131
6.8. Prueba de escritorio de la generación de una tabla	133
6.9. Prueba de escritorio de la generación de días	134

6.10. Arreglo bidimensional	134
6.11. Inicialización de tres tipos de matrices	135
6.12. Inicialización de matriz de diferentes tamaños	136
6.13. Asignación de valores en la matriz tabla	136
6.14. Asignación de valores en la matriz nombre	137
6.15. Inicialización y creación de arreglos	137
6.16. Recuperación de un valor en la matriz tabla	138
6.17. Recuperación de valores en la matriz nombre	138
6.18. Prueba de escritorio de la creación de matriz	140
6.19. Prueba de escritorio visualizando el error NullPointerException	141
6.20. Prueba de escritorio visualizando el error ArrayIndexOutOfBoundsException	142
7.1. Tipos de multiplicidad	153
8.1. Diagrama de clase: datos de Persona	166
8.2. Clases concretas de implementación de colecciones	170
8.3. Clases concretas de implementación de diccionarios	173
9.1. Presentación de acceso con paquetes	182
9.2. Opciones para generación de JAR	187
9.3. Opciones para generación de la documentación javaDoc	191
10.1. Clases envoltorios	210
12.1. Resumen de métodos de la clase StackTraceElement	246
13.1. Interfaces de escucha de las clases adaptadoras del paquete java.awt.event	317
14.1. Resumen de métodos de la clase File	327
14.2. Resumen de métodos de la clase RandomAccessFile	336
15.1. Formas de recorrer un árbol binario	359
16.1. Métodos de inicialización de jPicUSB	396
16.2. Métodos generales de jPicUSB	397
16.3. Métodos rápidos de jPicUSB	398

Bibliografía

Javadoc 5.0 java 5.0 tool.

Prácticas de Lenguajes de Programación (Java). Universidad de Valencia.

(2008). *Documentando con Javadoc*. MADS Group - Departamento de Computacion.

Aguilar, L. J. (1996). *Programación Orientada a Objetos*. McGraw-Hill, 1.0 edition.

Association, E. C. M. et al. (2005). Standard ECMA-334: C# Language Specification.

Balaguer, F. (2011). Introducción a UML.

Barreda, G. G. and Alvarez, J. R. F. (2012). Capacidades concurrentes del lenguaje Io. *Editor: Prof. Ariel Ortiz Ramírez*, page 61.

Basurto, M. A. P. and Espín, J. M. C. (2010). *Introducción a la programación en C*, volume 42. Edicions Upc.

Bell, D. and Parr, M. (2010). *Java for Students*. Prentice Hall Press Upper Saddle River, sexta edition.

Blanco, L. M. (2002). Programación en Visual Basic. Net-Grupo EIDOS, Madrid (España). Technical report, ISBN 84-88457-53-7.

Booch, G., Jacobson, I., and Rumbaugh, J. (1999). The unified modeling language reference manual.

Chan, P., Lee, R., and Kramer, D. (1999). *The Java Class Libraries: Supplement for the Java 2 platform standard edition, v. 1.2*, volume 1. Addison-Wesley Professional.

Corvi, J. (2012). Componente Genérico de Auditoría para Monitorear Cambios en el Modelo de Objetos.

de Silva Garza, A. G. and de Jesús Ania Briseño, I. (2008). *Introducción a la Computación*. Cengage Learning Editores.

Deitel, H. M., Deitel, P. J., and Santry, S. E. (2001). *Advanced Java 2 platform how to program*. Prentice Hall PTR.

Deitel, P. and Deitel, H. (2006). *Java(TM) how to program*. Prentice Hall Press.

Dr. Sodiya, A. A. (2008). *Programming and Algorithms*. National Open University of Nigeria.

Durán, F., Gutiérrez, F., and Pimentel, E. (2007). *Programación orientada a objetos con Java*. Editorial Paraninfo.

- ESPAÑOLA, R. (2005). Diccionario de la real academia española. madrid, 2005.
- Farrell, J. (2012). *An object-oriented approach to programming logic and design*. CengageBrain.com.
- Farrell, J. (2013). *Java Programming*. Cengage Learning.
- Flanagan, D. (2013). *Java in a Nutshell*. o'reilly.
- Fowler, M. and Scott, K. (2000). *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley Longman Publishing Co., Inc.
- Freeman, E., Freeman, E., Sierra, K., and Bates, B. (2004). *Head first design patterns*. O'Reilly Media.
- Friesen, J. and Friesen, G. (2001). *Java 2 by example*. Que Publishing.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.
- García, J., Rodríguez, J. I., Mingo, I., Imaz, A., Brazález, A., Larzabal, A., Calleja, J., and García, J. (2000). *Aprenda Java como si estuviera en primero*.
- García, J. C. L. (2009). *Algoritmos y Programación*, segunda edition.
- Gironés, J. T. (2012). La herencia en Java.
- Gosling, J., Joy, B., Steele, G., and Bracha, G. (2005). *Java (TM) Language Specification, The (Java (Addison-Wesley))*. Addison-Wesley Professional, tercera edition.
- Groussard, T. (2010). *Recursos Informáticos Java Enterprise Edition - Desarrollo de aplicaciones web con JEE 6*. Ediciones ENI.
- Groussard, T. (2012). *JAVA 7 - Los fundamentos del lenguaje Java*. Ediciones ENI.
- Harold, E. (2010). *Java I/O*. O'Reilly Media.
- Hekmat, S. (2005). *C++ Essentials*. PragSoft Corporation.
- Hernández, G. and María, J. (2012). Visualización de esquemas UML con HTML5.
- Horstmann, C. S. and Cornell, G. (2001). Core Java(TM) 2: Volume II–Recursos Avanzados. *Sao Paulo: Makrom Book*.
- Horstmann, C. S. and Cornell, G. (2005). Core Java(TM) 2: Volume I–Fundamentals. *PTR Prentice-Hall, Englewood Cliffs, NJ, 7632*.
- Horton, I. (2011). *Ivor Horton's Beginning Java*. Wiley. com.
- J. Antonio Lemos B, E. V. Z. (2004). *Algoritmos con JAVA*. 1.0 edition.
- Jr, F. P. B. (1987). No silver bullet essence and accidents of software engineering. *Computer*, 20(4):10 – 19.
- Lafore, R. (2002). *Data Structures and Algorithms in Java*. Sams Publishing, segunda edition.
- Larman, C. (1999). *UML y patrones*. Pearson.

- Larman, C. (2012). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, 3/e. Pearson Education India.
- lga Pons Capote (2008). *Introducción a los sistemas de bases de datos*. Editorial Paraninfo.
- Lindholm, T., Yellin, F., Bracha, G., and Buckley, A. (2013). *The Java virtual machine specification*. Addison-Wesley.
- Linington, P. F., Rutt, T., Miyazaki, H., and Vallecillo, A. (2013). Representing ODP enterprise language deontic tokens in UML. In *Proceedings of the 1st Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling*, page 7. ACM.
- Loucopoulos, P. and Karakostas, V. (1995). *System requirements engineering*. McGraw-Hill, Inc.
- Loy, M., Eckstein, R., Wood, D., Elliott, J., and Cole, B. (2012). *Java swing*. O'Reilly, segunda edición.
- Mariano, T. and Tomás, V. (2012). Encriptación y compresión de archivos en un modelo cliente-servidor.
- Maza, M. Á. S. (2012). *JavaScript*. INNOVACIÓN Y CUALIFICACIÓN.
- Morero, F. (2000). *Object Oriented Programming*. Eidos, 1.0 edition.
- Müller, K. (2012). How to make unified modeling language diagrams accessible for blind students. In *Computers Helping People with Special Needs*, pages 186 – 190. Springer.
- Northrup, T. (2009). *MCTS Self-Paced Training Kit (Exam 70-536)*. Microsoft Press.
- Paul Dietel, H. D. (2008). *Java Como Programar*. Pearson Education, séptima edición.
- Pérez, G. G. (2008). *Aprendiendo Java y Programación Orientada a Objetos*.
- Poyla, G. (1988). *How to solve it*. Princeton University Press, segunda edición.
- Quintero, J. B., de Páez, R. A., Marín, J. C., and López, A. B. (2012). Un estudio comparativo de herramientas para el modelado con UML. *revista universidad eafit*, 41(137):60 – 76.
- Regino, E. M. O. (2005). *Lógica de programación*. ECOE Ediciones, segunda edición.
- Rivera, F. L. O. (2008). *Lógica y programación orientada a los objetos: un inicio al desarrollo de software*. ITM.
- Roldán, R. C. Programación.
- Schildt, H. (2002). *Java (TM) 2: The Complete Reference*. McGraw-Hill/Osborne Media, quinta edición.
- Schmuller, J. (2000). *Aprendiendo UML en 24 horas*. Pearson educación.
- Serrano, J. F. V., Abril, A. P., Bellas, P. G., and Calle, Á. S. (2010). *Diseñar y programar, todo es empezar.: Una introducción a la programación orientada a objetos usando UML y Java*. Librería-Editorial Dykinson.
- Sierra, K. and Bates, B. (2008). *SCJP Sun Certified Programmer for Java 5: Study Guide*. McGraw-Hill.

- Tello, J. C. Diagramas de Caso de Uso.
- Tremblay, J. P., Garcia-Bermejo, R., and Grassmann, W. K. (1996). *Matemática discreta y lógica*. Prentice Hall.
- Valbuena, S. J., Torres, S. A. C., and Ramirez, M. L. V. (2008). *Técnicas de Diseño de Algoritmos en Java*. Ediciones Elizcom, primera edition.
- Valbuena, S. J., Torres, S. A. C., and Rodríguez, L. A. H. (2010). *Programación Orientada a Objetos*. Ediciones Elizcom.
- Villarreal, A., Rómulo, D., Carrera, V., and Fernando, A. (2012). *Análisis, diseño y desarrollo de un curso de Java a nivel básico (Netbeans y Eclipse) para el personal técnico del ejército utilizando la plataforma de teleformación MOODLE y herramientas Web 2.0*. PhD thesis, SANGOLQUÍ/ESPE/2012.
- Xavier Ferré Grau, M. I. S. S. Desarrollo Orientado a Objetos con UML.
- Yu, L., France, R. B., Ray, I., and Sun, W. (2012). Systematic scenario-based analysis of uml design class models. In *Engineering of Complex Computer Systems (ICECCS), 2012 17th International Conference on*, pages 86 – 95. IEEE.