

## INDICE

Contenido	VII
Prologo	XIX
Prologo a la 2da edición	XXV
Sobre bibliografía, las fuentes de Internet y los ejercicios	XXVII
Prologo a la edición española	XXIX
Intentamos mejorar este libro	XXXVII
Parte A: Los problemas	1
<b>Capitulo 1. Calidad del software</b>	<b>3</b>
1.1. Factores externos e internos	1
1.2. Una revisión de los factores externos	4
1.3. Sobre el mantenimiento del software	16
1.4. Conceptos clave introducción en este capitulo	18
1.5. Notas bibliográficas	19
<b>Capitulo 2. Criterios de la orientación a objetos</b>	
2.1. Sobre los criterios	21
2.2. Método y lenguaje	22
2.3. Implementación y entorno	31
2.4. Bibliotecas	33
2.5. Para una visión previa mas extensa	
2.6. Notas bibliográficas y recursos de objetos	34
Parte B: El cambio a la orientación a objetos	37
<b>Capitulo 3. Modularidad</b>	<b>39</b>
3.1. Cinco criterios	40
3.2. Cinco reglas	46
3.3. Cinco principios	51
3.4. Conceptos clave presentados en este capitulo	
3.5. Notas bibliográficas	61
Ejercicios	62
<b>Capitulo 4. Enfoques para la reutilización</b>	
4.1. Las metas de la reutilización	65
4.2. ¿Qué es lo que se debería reutilizar?	68
4.3. La repetición en el desarrollo de software	71
4.4. Obstáculos no técnicos	72
4.5. El problema técnico	78
4.6. Cinco requisitos relativos a las estructuras de los módulos	80
4.7. Estructuras modulares tradicionales	85
4.8. Sobrecarga y genericidad	89
4.9. Conceptos clave presentados en este capitulo	94
4.10. Notas bibliográficas	95
<b>Capitulo 5. Hacia la tecnología de objeto</b>	
5.1. Los ingredientes de la computación	97
5.2. Descomposición funcional	99
5.3. Descomposición basad en objetos	109
5.4. Construcción de software orientado a objetos	110
5.5. Cuestiones	111
5.6. Conceptos clave presentados en este capitulo	
5.7. Notas bibliográficas	113

<b>Capítulo 6. Tipos abstractos de datos</b>	115
6.1. Criterios	
6.2. Variaciones de implementación	116
6.3. Hacia una visión abstracta de los objetos	119
6.4. Formalizar la especificación	123
6.5. De los tipos abstractos de datos a las clases	134
6.6. Mas allá del software	
6.7. Temas suplementarios	140
6.8. Conceptos clave presentados en este capítulo	150
6.9. Notas bibliográficas	151
Ejercicios	152
Parte C: Técnicas orientados a objetos	155
<b>Capítulo 7. La estructura estática: las clases</b>	
7.1. Los objetos no son el sujeto	157
7.2. Evitar la confusión estándar	158
7.3. El papel (rol) de las clases	161
7.4. Un sistema de tipos uniforme	162
7.5. Una clase sencilla	163
7.6. Convenciones básicas	168
7.7. El estilo orientado a objetos	171
7.8. Exportación selectiva y ocultación de información	181
7.9. Uniéndolo todo	183
7.10. Discusión	192
7.11. Conceptos clave introducción en este capítulo	202
7.12. Notas bibliográficas	203
Ejercicios	204
<b>Capítulo 8. La estructura de ejecución: los objetos</b>	
8.1. Objetos	205
8.2. Los objetos como herramienta de modelado	216
8.3. Manipulación de objetos y referencias	218
8.4. Procedimientos de creación	222
8.5. Mas pobre las referencias	226
8.6. Operaciones sobre referencias	228
8.7. Objetos compuestos y tipos expandidos	239
8.8. Conexión: semántica por referencia y por valor	246
8.9. Beneficios y peligros de trabajar con referencias	250
8.10. Discusión	255
8.11. Conceptos clave presentados en este capítulo	260
8.12. Notas bibliográficas	
Ejercicios	261
<b>Capítulo 9. Gestión de la memoria</b>	
9.1. Que les pasa a los objetos	263
9.2. El enfoque despreocupado	274
9.3. Recuperación de memoria: las cuestiones	276
9.4. Liberación de memoria controlada por el programador	277
9.5. El enfoque del nivel de componentes	279
9.6. Gestión automática de memoria	283
9.7. Recuento de referencias	284
9.8. Recolección de basura	286

9.9. Aspectos prácticos de la recolección de basura	291
9.10. Un entorno con gestión de memoria	293
9.11. Conceptos clave presentados en este capítulo	296
9.12. Notas bibliográficas	
Ejercicio	297
<b>Capítulo 10. Genericidad</b>	
10.1. Generalización de tipos horizontal y vertical	299
10.2. Necesidad de la parametrización de tipos	300
10.3. Clases genéricas	302
10.4. Arrays	307
10.5. El coste de la genericidad	
10.6. Discusión: no se hará todavía	310
10.7. Conceptos clave presentados en este capítulo	
10.8. Notas bibliográficas	311
Ejercicios	
<b>Capítulo 11. Diseño por contrato: construcción de software fiable</b>	313
11.1. Mecanismos básicos de fiabilidad	
11.2. Sobre la correlación del software	314
11.3. Expresar una especificación	316
11.4. Introducir aserciones en los textos de software	
11.5. Precondiciones y postcondiciones	319
11.6. Contratación para la fiabilidad del software	323
11.7. Trabajo con aserciones	329
11.8. Invariantes de clase	343
11.9. ¿Cuándo es correcta una clase?	349
11.10. La conexión TAD	352
11.11. Una instrucción de aserción	357
11.12. Invariantes y variantes de bucle	359
11.13. Utilización de aserciones	366
11.14. Discusión	375
11.15. Conceptos clave presentados en este capítulo	
11.16. Notas bibliográficas	383
Ejercicios	384
Apéndice: El desastre del Ariane 5	386
<b>Capítulo 12. Cuando se rompe el contrato: tratamiento de excepciones</b>	387
12.1. Conceptos básicos del tratamiento de excepciones	
12.2. Tratamiento de excepciones	390
12.3. Un mecanismo de excepciones	394
12.4. Ejemplos de tratamiento de excepciones	396
12.5. La tarea de una cláusula de rescate	402
12.6. Tratamiento avanzado de excepciones	405
12.7. Discusión	410
12.8. Conceptos clave presentados en este capítulo	411
12.9. Notas bibliográficas	
Ejercicios	412
<b>Capítulo 13. Mecanismos de soporte</b>	
13.1. Interfaz con el software NO O-O	413
13.2. Pago de argumentos	418

13.3. Instrucciones	420
13.4. Expresiones	425
13.5. Cadenas de caracteres	429
13.6. Entrada y salida	
13.7. Convenciones lexicográficas	430
13.8. Conceptos clave introducidos en este capítulo	
Ejercicios	431
<b>Capítulo 14. Introducción a la herencia</b>	
14.1. Polígonos y rectángulos	433
14.2. Polimorfismo	441
14.3. Tipos y herencia	445
14.4. Ligadura dinámica	452
14.5. Clases y características diferidas	454
14.6. Técnicas de redeclaración	463
14.7. El significado de la herencia	466
14.8. El papel de las clases diferidas	471
14.9. Discusión	478
14.10. Conceptos clave introducidos en este capítulo	486
14.11. Notas bibliográficas	
Ejercicios	487
<b>Capítulo 15. Herencia múltiples</b>	
15.1. Ejemplos de herencia múltiple	489
15.2. Renombrar características	504
15.3. Aplanar la estructura	510
15.4. Herencia repetida	512
15.5. Discusión	531
15.6. Conceptos clave introducidos en este capítulo	533
15.7. Notas bibliográficas	
Ejercicios	534
<b>Capítulo 16. Técnicas de herencia</b>	
16.1. Herencia y aserciones	537
16.2. La estructura global de la herencia	547
16.3. Características congeladas	550
16.4. Genericidad restringida	552
16.5. Intento de asignación	557
16.6. Tipos y redeclaración	561
16.7. Declaración con ancla	564
16.8. Herencia y ocultación de información	570
16.9. Conceptos clave introducidos en este capítulo	
16.10. Notas bibliográficas	575
Ejercicios	576
<b>Capítulo 17. Comprobación estricta de tipos</b>	
17.1 El problema de la comprobación estricta de tipos	577
17.2. Comprobación estática de tipos: como y por que	581
17.3. Covarianza y ocultación de información en el descendente	587
17.4. Primeras aproximaciones a la validación de sistemas	594
17.5. Tomando como base los tipos con ancla	595
17.6. Análisis global	599
17.7. ¡Cuidado con los CATCALL polimorfos!	601

17.8. Valoración	604
17.9. El ajuste perfecto	605
17.10. Conceptos clave estudiados en este capítulo	
17.11. Notas bibliográficas	606
<b>Capítulo 18. Objetos globales y constantes</b>	
18.1. Constantes de os tipos básicos	609
18.2. Uso de las constantes	611
18.3. Constantes de tipo clase	612
18.4. Aplicaciones de las rutinas ONCE	614
18.5. Constantes de tipo STRING	
18.6. Valores UNIQUE	619
18.7. Discusión	621
18.8. Conceptos clave introducidos en este capítulo	624
18.9. Notas bibliográficas	
Ejercicios	625
<b>Parte D: Metodología orientada a objetos: aplicar bien el método</b>	627
<b>Capítulo 19. Sobre la metodología</b>	
19.1. Metodología del software: que y por que	629
19.2. Construcción de buenas reglas: asesorar a los asesores	630
19.3. Sobre la utilización de metáforas	637
19.4. La importancia de ser humilde	639
19.5. Notas bibliográficas	
Ejercicios	640
<b>Capítulo 20. Patrón de diseño: sistemas interactivos Multi-Panel</b>	
20.1. Sistemas Multi-Panel	641
20.2. Un intento intuitivo	643
20.3. Una solución funcional descendente	644
20.4. Una critica de la solución	647
20.5. Una arquitectura orientada a objetos	649
20.6. Discusión	657
20.7. Notas bibliográficas	658
<b>Capítulo 21. Herencia, un caso de estudio: “undo” en su sistema interactivo</b>	659
21.1. Diabolicum perseverare	
21.2. Encontrar las abstracciones	662
21.3. Deshacer-rehacer de múltiples niveles	667
21.4. Aspectos de implementación	670
21.5. Una interfaz de usuario para deshacer y rehacer	674
21.6. Discusión	675
21.7. Notas bibliográficas	677
Ejercicios	678
<b>Capítulo 22. Como encontrar las clases</b>	
22.1. Estudio de un documento de requisitos	681
22.2. Señales de peligro	687
22.3. Heurística generales para encontrar las clases	692
22.4. Otras fuentes de clases	696
22.5. Reutilización	701
22.6. El método para obtener las clases	702
22.7. Conceptos clave presentados en este capítulo	703

22.8. Notas bibliográficas	
Ejercicios	704
<b>Capítulo 23. Principios de diseño de clases</b>	707
23.1. Efectos laterales de las funciones	708
23.2. ¿Cuántos argumentos debe tener una característica?	723
23.3. Tamaño de clase: el enfoque de lista de la compra	729
23.4. Estructura activas de datos	733
23.5. Exportaciones selectivas	753
23.6. Abordaciones los casos excepcionales	754
23.7. Evolución de clases. La cláusula obsoleto	758
23.8. Documentación de una clase y de un sistema	760
23.9. Conceptos presentados en este capitulo	762
23.10. Notas bibliográficas	763
Ejercicios	763
<b>Capítulo 24. Utilizando bien de la herencia</b>	
24.1. Como hay que utilizar la herencia	765
24.2. ¿Prefiere usted comprar o heredar?	768
24.3. Una aplicación. La técnica Handle	773
24.4. Taxonomía	775
24.5. Utilización de la herencia: una taxomonía de la taxonomía	777
24.6. ¿Un mecanismo o mas?	788
24.7. Herencia de subtipos y ocultación en descendentes	790
24.8. Herencia de implementación	797
24.9. Herencia de facilidades	800
24.10. Criterios múltiples y herencia de vistas	804
24.11. Como desarrollar estructuras de herencia	810
24.12. Visión resumida: utilizando bien la herencia	814
24.13. Conceptos clave presentados en este capitulo	
24.14. Notas bibliográficas	815
24.15. Apéndice: una historia de la taxonomía	816
Ejercicios	821
<b>Capítulo 25. Técnicas útiles</b>	
25.1. Filosofía de diseño	823
25.2. Clases	824
25.3. Técnicas de herencia	825
<b>Capítulo 26. Un sentido del estilo</b>	
26.1. ¡Asuntos cosméticos!	827
26.2. Selección de nombres correctos	830
26.3. Utilización de constantes	836
26.4. Comentarios de encabezado y cláusulas de indexación	837
26.5. Disposición y presentación del código	843
26.6. Tipos de letra	851
26.7. Notas bibliográficas	852
Ejercicios	853
<b>Capítulo 27. Análisis orientado a objetos</b>	
27.1. Objetivos del análisis	855
27.2. La naturaleza cambiante del análisis	
27.3. La contribución de la tecnología de objetos	858
27.4. Programación de una emisora de televisión	859

27.5. Expresión del análisis: puntos de vista múltiples	865
27.6. Métodos de análisis	869
27.7. La notación de objetos en negocios	870
27.8. Bibliografía	873
<b>Capítulo 28. El proceso de construcción del software</b>	
28.1. Clusters	875
28.2. Ingeniería concurrente	876
28.3. Pasos y tareas	877
28.4. El modelo de clusters del ciclo de vida del software	878
28.5. Generalización	880
28.6. Ausencia de discontinuidades y reversibilidad	882
28.7. Con nosotros, todo es la cara	884
28.8. Conceptos clave abarcados en este capítulo	
28.9. Notas bibliográficas	885
<b>Capítulo 29. Enseñanza del método</b>	
29.1. Formación industrial	887
29.2. Cursos introductorios	889
29.3. Otros cursos	892
29.4. Hacia una nueva pedagogía del software	894
29.5. Un plan orientado a objetos	897
22.6. Conceptos claves estudiados en este capítulo	
22.7. Notas bibliográficas	899
Parte E: Temas avanzados	901
<b>Capítulo 30. Concurrencia, distribución, cliente-servidor e Internet</b>	903
30.1. Una presentación previa algo chivata	
30.2. El crecimiento de la importancia de la concurrencia	905
30.3. De procesos a objetos	908
30.4. Presentación de la ejecución concurrente	915
30.5. Problemas de sincronización	927
30.6. Acceso a objetos separados	932
30.7. Condiciones de espera	941
30.8. Solicitud de servicios especiales	948
30.9. Ejemplos	952
30.10. Hacia una regla de demostración	968
30.11. Un resumen del mecanismo	970
30.12. Discusión	973
30.13. Conceptos clave presentados en este capítulo	976
30.14. Notas bibliográficas	977
Ejercicios	979
<b>Capítulo 31. Persistencia de objetos y bases de datos</b>	
31.1 Persistencia a partir del lenguaje	983
31.2. Mas allá del cierre de persistencia	985
31.3. Evolución de esquemas	986
31.4. De la persistencia a las bases de datos	992
31.5. interoperabilidad objeto-relacional	993
31.6. Fundamentos de las bases de datos orientadas a objetos	996
31.7. sistemas de bases de datos O-O: Ejemplos	1001
31.8. Discusión: mas allá de las bases de datos O-O	1003

31.9. Conceptos clave estudiados en este capítulos	1006
31.10. Notas bibliográficas	1007
Ejercicios	1007
<b>Capítulo 32. Técnicas O-O- para aplicaciones graficas interactivas</b>	1009
32.1. Herramientas necesarias	1010
32.2. Portabilidad y adaptación a una plataforma	1012
32.3. Abstracciones graficas	1014
32.4. Mecanismos de interacción	1017
32.5. Gestionando los eventos	1018
32.6. Un modelo matemático	
32.7. Notas bibliográficas	1022
Parte F: Aplicación del método a diferentes lenguajes y entornos	1023
<b>Capítulo 33. Ada y la programación orientada a objetos</b>	
33.1. Un poco descontento	1025
33.2. Paquetes	
33.3. Implementación de una pila	1027
33.4. Ocultación de la presentación: la historia de la cláusula private	1031
33.5. Excepciones	1033
33.6. Tareas	1036
33.7. de Ada a Ada 95	1037
33.8. Conceptos clave del capítulo	
33.9. Notas bibliográficas	1041
Ejercicios	1042
<b>Capítulo 34. Emulación de la tecnología de objetos en entornos no O-O</b>	1045
34.1. Niveles de soporte de los lenguajes	
34.2. ¿Programación orientada a objetos en pascal?	1046
34.3. Fortran	1047
34.4. C y la programación orientada a objetos	1052
34.5. Notas bibliográficas	
Ejercicios	1957
Capítulo 35. De simula a Java y mas alla: los principales entornos y lenguajes O-O	1059
35.1. SIMULA	
35.2. SMALLTALK	1079
35.3. EXTESNIONES DE LISP	
35.4. EXTESNIONES DE C	1075
35.5. JAVA	1080
35.6. Otros lenguajes sacados a objeto	
35.7. Notas bibliográficas	1081
Ejercicios	1083
Parte G. Hacerlo bien	1085
<b>Capítulo 36. Un entorno orientado a objetos</b>	
36.1. COMPONENTES	1087
36.2. El lenguaje	
36.3. La tecnología de compilación	1088
36.4. HERRAMEINTAS	1092
36.5. BIBLIOTECAS	1094



36.6. MECANISMO DE INTERFAZ	1096
36.7. Notas bibliográficas	1103
Epilogo, exponiendo con total franqueza el lenguaje	1105
Parte H: Apéndices	1107
Apéndice A. Extractos de las bibliotecas base	1109
Apéndice B. Genericidad frente a herencia	
B.1. GENERALIDAD	1111
B.2. HERENCIA	1116
B.3. EMULACION DE LA HERENCIA MEDIANTE GENERICIDAD	1118
B.4. EMULACION DE LA GENERICIDAD MEDIANTE HERENCIA	1119
B.5. COMBINACION DE LA GENERICIDAD Y DE LA HERENCIA	1126
B.6. CONCEPTOS CLAVE PRESTADOS EN ESTE APENDICE	
B.7. Notas bibliográficas	1130
Ejercicios	
<b>Apéndice C. Principios, reglas, preceptos y definiciones</b>	1133
<b>Apéndice D. Glosario de tecnología de objetos</b>	1137
<b>Apéndice E. Bibliografía</b>	
E.1. Trabajos de otros autores	1149
E.2. Trabajos realizados por el autor del presente libro	1165
<b>Indice</b>	1171