

INDICE

Prefacio	XIX
Prologo a la edición española	XXI
Parte I. el producto y el proceso	1
Capitulo 1. El producto	3
1.1. la evolución del software	
1.1.1. una perspectiva industrial	
1.1.2. 1.1.2. una fabrica de software que envejece	6
1.1.3. competitividad del software	
1.2. el software	7
1.2.1. características del software	
1.2.2. componentes del software	9
1.2.3. aplicaciones del software	10
1.3. Software: una crisis en el horizonte	
1.4. mitos del software	12
1.5. Resumen	
Referencias	14
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	15
Capitulo 2. El proceso	17
2.1. ingeniería del software: una tecnología estratificada	
2.1.1. proceso, métodos y herramientas	
2.1.1. una visión general de la ingeniería del software	18
2.2. el proceso del software	19
2.3. modelos de proceso del software	21
2.4. el modelo lineal secuencial	22
2.5. el modelo de construcción de prototipos	24
2.6. el modelo DRA	25
2.7. modelos evolutivos de proceso del software	26
2.7.1. el modelo incremental	
2.7.2. el modelo espiral	28
2.7.3. el modelo de ensamblaje de componentes	29
2.7.4. el modelo de desarrollo concurrente	30
2.8. el modelo de métodos formales	31
2.9. técnicas de cuarta generación	32
2.10. tecnologías de proceso	
2.11. producto y proceso	33
2.12. Resumen	
Referencias	34
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	35
Parte II. Gestión de proyectos de software	37
Capitulo 3. Conceptos sobre gestión de proyectos	
3.1. el espectro de la gestión	39
3.1.1. personal	
3.1.2. el problema	
3.1.3. el proceso	
3.2. personal	40

3.2.1. los participantes	41
3.2.2. los jefes de equipo	
3.2.3. el equipo de software	
3.2.4. aspectos sobre la coordinación y la comunicación	44
3.3. el problema	45
3.3.1. ámbito del software	
3.3.2. descomposición del problema	
3.4. el proceso	46
3.4.1. maduración del producto y del proceso	
3.4.2. descomposición del proceso	
3.5. el proyecto	48
3.6. Resumen	
Referencias	
Problemas y puntos a considerar	49
Otras lecturas y fuentes de información	50
Capítulo 4: Proceso de software y métricas de proyectos	51
4.1. medidas, métricas e indicadores	
4.2. métricas en el proceso y dominios del proyecto	
4.2.1. métricas en el proceso y mejoras en el proceso del software	52
4.2. métricas del proyecto	
4.3. mediciones del software	
4.3.1. métricas orientadas al tamaño	55
4.3.2. métricas orientadas a la función	56
4.3.3. métricas ampliadas de punto de función	57
4.4. reconciliación de los diferentes enfoques de métricas	58
4.5. métricas para la calidad del software	60
4.5.1. visión general de los factores que afectan a la calidad	
4.5.2. medida de la calidad	
4.5.3. eficacia de la eliminación de efectos	61
4.6. integración de las métricas dentro del proceso de ingeniería del software	62
4.7. Resumen	63
Referencias	64
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	
Capítulo 5. Planificación de proyectos de software	65
5.1. observaciones sobre la estimación	
5.2. objetivos de la planificación del proyecto	
5.3. ámbito del software	70
5.3.1. obtención de la información necesarias para el ámbito	
5.3.2. un ejemplo de ámbito	
5.4. recursos	71
5.4.1. recursos humanos	
5.4.2. recursos de software reutilizables	
5.4.3. recursos de entorno	73
5.5. estimación del proyecto de software	
5.6. técnicas de descomposición	
5.6.1. tamaño del software	74
5.6.2. estimación basad en el problema	75
	76

5.6.3. un ejemplo de estimación basada en LDC	77
5.6.4. un ejemplo de estimación basada en PF	78
5.6.5. estimación basada en el proceso	
5.6.6. un ejemplo de estimación basada en el proceso	79
5.7. modelos empíricos de estimación	
5.7.1. la estructura de los modelos de estimación	80
5.7.2. el modelo COCOMO	81
5.7.3. la ecuación del software	82
5.8. la decisión de desarrollo-comprar	83
5.8.1. creación de un árbol de decisiones	
5.8.2. subcontratación (outsourcing)	84
5.8. Herramientas automáticas de estimación	
5.10. Resumen	85
Referencias	
Problemas y puntos a considerar	86
Otras lecturas y fuentes de información	87
Capítulo 6. Gestión del riesgo	
6.1. Estrategias de riesgo proactivas vs. reactivas	89
6.2. riesgo del software	
6.3. identificación del riesgo	90
6.3.1 riesgo del tamaño del producto	
6.3.2. riesgo del impacto en el negocio	91
6.3.3. riesgos relacionados con el cliente	
6.3.4. riesgos del proceso	92
6.3.5. riesgos tecnológicos	
6.3.6. riesgo del entorno del desarrollo	93
6.3.7. riesgos asociados con el tamaño de la plantilla de personal y su experiencia	
6.3.8. componentes y controladores del riesgo	94
6.4. proyección del riesgo	
6.4.1. desarrollo de una tabla de riesgo	
6.4.2. evaluación del impacto del riesgo	
6.4.3. evaluación del riesgo	97
6.5. reducción, supervisión y gestión del riesgo	98
6.6. riesgos y peligros para la seguridad	
6.7. el plan RSGR	99
6.8. Resumen	
Referencias	100
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	101
Capítulo 7: Planificación temporal y seguimiento del proyecto	
7.1. conceptos básicos	103
7.1.1. comentarios sobre “los retrasos”	104
7.1.2. principio básicos	105
7.2. la relación entre las personas y el esfuerzo	
7.2.1. un ejemplo	106
7.2.2. una relación empírica	
7.2.3. distribución del esfuerzo	
7.3. Definición de un conjunto de tareas para el proyecto de software	107

7.3.1. grado de rigor	108
7.3.2. definir los criterios de adaptación	
7.3.3 calculo del valor selector del conjunto de tareas	
7.3.4. interpretar el valor SCT y seleccionar el conjunto de tareas	109
7.4. Selección de las tareas de ingeniería del software	110
7.5. refinamiento de las tareas principales	111
7.6. definir una red de tareas	113
7.7. la planificación temporal	114
7.7.1. gráficos de tiempo	
7.7.2. seguimiento de la planificación temporal	
7.8. El plan del proyecto	117
7.9. Resumen	
Referencias	118
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	119
Capítulo 8: Control de calidad del software	
8.1. conceptos de calidad	121
8.1.2. control de calidad	122
8.1.1. calidad	
8.1.3. garantía de calidad	
8.1.4. coste de calidad	123
8.2. la tendencia de la calidad	124
8.3. garantía aseguramiento de calidad del software	125
8.3.1. aspectos de segundo plano	
8.3.2. actividades de SQA	
8.4. Revisiones del software	126
8.4.1. impacto de los efectos del software sobre el coste	
8.4.2. amplificación eliminación de defectos	
8.5. Revisiones técnicas formales	128
8.5.1. la reunión de revisión	129
8.5.2. registros e informe de la revisión	
8.5.3. directrices para la revisión	
8.6. enfoques formales a la SQA	130
8.7. Garantía de la calidad estadística	131
8.8. fiabilidad del software	133
8.7.1. medidas de fiabilidad y de disponibilidad	
8.8.2. análisis de riesgos y seguridad del software	
8.9. El plan de SQA	134
8.10. los estándares de calidad ISO 9000	135
8.10.1. el enfoque ISO en sistemas de garantía de calidad	
8.10.2. El estándar de calidad ISO 90001	
8.11 Resumen	136
Referencias	137
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	138
Capítulo 9: Gestión de la configuración del software	
9.1. Gestión de la configuración del software	141
9.1.1. líneas base	142
9.1.2. elementos de configuraciones del software	143

9.2. El proceso de GCS	
9.3. Identificación de objetos en la configuración del software	144
9.4. Control de versiones	146
9.5. Control de cambios	148
9.6. Auditoría de la configuración	149
9.7. Informes de estado	
9.8. estándares de GCS	150
9.9. Resumen	
Referencias	151
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	152
Parte III. Métodos convencionales es para la ingeniería del software	155
Capítulo 10. Ingeniería de sistemas	
10.1. sistemas basados en computadora	157
10.2. la jerarquía de la ingeniería de sistemas	159
10.2.1. modelado del sistema	
10.2.2. ingeniería de la información: una visión general	160
10.2.3. ingeniería de productos: una visión general	162
10.3. ingeniería de la información:	
10.4. planificación de la estrategia de la información	163
10.4.1. modelado de la empresa	164
10.4.2. modelado de datos al nivel del negocio	165
10.5.1. identificación de requisitos	
10.5. análisis del área de negocio	
10.5.1. Modelado del proceso	166
10.5.2. modelado del flujo de información	167
10.6. ingeniería de productos	168
10.6.1. análisis del sistema	
10.6.2. identificación de necesidades	170
10.6.3. estudio de la viabilidad	171
10.6.4. análisis económico	172
10.6.5. análisis técnico	
10.7. modelado de la arquitectura del sistema	174
10.8. modelado y simulación del sistema	177
10.9. especificación del sistema	
10.10. Resumen	178
Referencias	
Problemas y puntos a considerar	179
Otras lecturas y fuentes de información	180
Capítulo 11. Conceptos y principios del análisis	
11.1. análisis de requisitos	183
11.2. técnicas de comunicación	
11.2.1. inicio del proceso	185
11.2.2. técnicas para facilitar las especificaciones de una aplicación	
11.2.3. despliegue de la función de calidad	187
11.3. Principios del análisis	188
11.3.1. el dominio de la información	189
11.3.2. modelado	
11.3.3. partición	190

11.3.4. visiones esenciales y de implementación	192
11.4. Creación de prototipos del software	
11.4.1. selección del enfoque de creación de prototipos	193
11.4.2. métodos y herramientas para el desarrollo de prototipos	
11.5. Especificación	194
11.5.1. principios de la especificación	
11.5.2. representación	195
11.5.3. la especificación de los requisitos del software	196
11.6. Revisión de la especificación	197
11.7. Resumen	
Referencias	198
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	199
Capítulo 12. Modelado del análisis	201
12.1. Breve historia	
12.2. elementos del modelo de análisis	202
12.3. modelado de datos	
12.3.1. objetos de datos, atributos y relaciones	203
12.3.2. cardinalidad y moralidad	
12.3.3. diagramas entidad-relación	205
12.4. Modelado funcional y flujo de información	207
12.4.1. diagramas de flujo de datos	208
12.4.2. ampliaciones para sistemas de tiempo real	209
12.4.3. ampliaciones de Ward y Mellor	210
12.4.4. ampliaciones de Hatley y Pirbhai	212
12.5. Modelado del comportamiento	213
12.6. Mecanismos del análisis estructurado	214
12.6.1. creación de un diagrama entidad-relación	215
12.6.2. creación de un modelo de flujo de datos	217
12.6.3. creación de un modelo de flujo de control	218
12.6.4. la especificación de control	219
12.6.5. especificación del proceso	220
12.7. El diccionario de datos	221
12.8. una visión general de otros métodos clásicos de análisis	
12.8.1. desarrollo de sistema estructurados de datos	223
12.8.2. desarrollo de sistemas Jackson	
12.8.3. SADT	
12.9. Resumen	224
Referencias	
Problemas y puntos a considerar	225
Otras lecturas y fuentes de información	226
Capítulo 13. Conceptos y principios de diseño	
13.1. diseño e ingeniería del software	229
13.2. el proceso de diseño	
13.2.1. diseño y calidad del software	230
13.2.2. la evolución del diseño del software	
13.3. Principios del diseño	231
13.4. Conceptos del diseño	
13.4.1. abstracción	232

13.4.2. refinamiento	
13.4.3. modularidad	233
13.4.4. arquitectura del software	235
13.4.5. jerarquía de control	
13.4.6. partición estructural	236
13.4.7. estructura de datos	237
13.4.8. procedimiento de software	
13.4.9. ocultación de información	238
13.5. Diseño modular efectivo	
13.5.1. independencia funcional	239
13.5.2. cohesión	
13.5.3. acoplamiento	240
13.6. Heurística de diseño para una modularidad efectiva	241
13.7. El modelo del diseño	
13.8. Documentación de el diseño	243
13.9. Resumen	244
Referencias	
Problemas y puntos a considerar	245
Otras lecturas y fuentes de información	246
Capítulo 14. Diseño arquitectónico	
14.1. Diseño de datos	249
14.2. diseño arquitectónico	250
14.2.1. contribuciones	
14.2.2. áreas de aplicación	251
14.3. el proceso del diseño arquitectónico	
14.3.1. flujo de transformación	
14.3.2. flujo de transacción	
14.4. Análisis de las transformaciones	252
14.4.1. un ejemplo	
14.4.2. pasos del diseño	253
14.5. Análisis de las transacciones	259
14.5.1. un ejemplo	
14.5.2. pasos del diseño	260
14.6. postproceso del diseño	262
14.7. optimización del diseño arquitectónico	
14.8. diseño de la interfaz	264
14.8.1. diseño de la interfaz interna y externa	
14.8.2. diseño de la interfaz de usuario	
14.9. diseño de la interfaz hombre- maquina	265
14.9.1. modelos de diseño de interfaz	
14.9.2. análisis y modelado de tareas	266
14.9.3. aspectos del diseño	267
14.9.4. herramientas de implementación	
14.9.5. evaluación del diseño	269
14.10. directrices para el diseño de interfaces	270
14.10.1. interacción general	
14.10.2. visualización de la información	271
14.10.3. entrada de datos	
14.11. diseño procedimental	272

14.11.1. programación estructurada	
14.11.2. notación grafica del diseño	273
14.11.3. notación tabular del diseño	274
14.11.4. lenguaje de diseño de programas	
14.11.5. un ejemplo de LDP	276
14.12. Resumen	278
Referencias	279
Problemas y puntos a considerar	280
Otras lecturas y fuentes de información	282
Capitulo 15. Diseño para sistemas de tiempo real	
15.1. consideraciones sobre los sistemas	285
15.2. sistemas de tiempo real	
15.2.1. aspectos de integración y rendimiento	286
15.2.2. manejo de interrupciones	287
15.2.3. bases de datos de tiempo real	
15.2.4. sistemas operativos de tiempo real	288
15.2.5. lenguajes de tiempo real	
15.2.6. sincronización y comunicación de tareas	289
15.3. análisis y simulaciones de tiempo	
15.3.1. herramientas matemáticas para el análisis de sistemas de tiempo real	290
15.3.2. técnicas de simulación y modelado	294
15.4. diseño en tiempo real	297
15.5. Resumen	
Referencias	298
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	
Capitulo 16. Métodos de prueba del software	
16.1. Fundamentos de las pruebas del software	301
17.1. objetivos de las pruebas	
16.1.2. principios de las pruebas	302
16.1.3. facilidad de prueba	303
16.2. Diseño de casos de prueba	304
16.3. Prueba de caja blanca	305
16.4. Prueba del camino básico	
16.4.1. notación de grafico de flujo	306
16.4.2. complejidad ciclomática	
16.4.3. obtención de casos de prueba	308
16.4.4. matrices de grafos	310
16.5. Prueba de la estructura de control	
16.5.1. prueba de condición	311
16.5.2. prueba del flujo de datos	313
16.5.3. pruebas de bucles	314
16.6. Prueba de caja negra	
16.6.1. métodos de prueba basados en grafos	315
16.6.2. partición equivalente	317
16.6.3. análisis de valores limite	
16.6.4. prueba de comparación	318
16.7. Prueba de entornos y aplicaciones especializadas	319

16.7.1. prueba de interfaces graficas de usuario	
16.7.2. prueba de arquitectura cliente/servidor	
16.7.3. prueba de la documentación y de ayuda	320
16.7.4. Prueba de sistemas de tiempo-real	
16.8. Resumen	
Referencias	322
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	323
Capitulo 17. Estrategias de pruebas de software	
17.1. Un enfoque estratégico para las pruebas del software	325
17.1.1. verificación y validación	326
17.1.2. organización para las pruebas del software	
17.1.3. una estrategia de prueba del software	327
17.1.4. criterios para completar la prueba	328
17.2. Aspectos estratégicos	329
17.3. Prueba de unidad	
17.3.1. consideraciones sobre la prueba de unidad	330
17.3.2. procedimientos de prueba de unidad	332
17.4. Prueba de integración	
17.4.1. integración descendente	333
17.4.2. integración ascendente	334
17.4.3. prueba de regresión	335
17.4.4. Comentarios sobre la prueba de integración	
17.4.5. documentación de la prueba de integración	336
17.5. Prueba de validación	
17.5.1. criterios de la prueba de validación	337
17.5.2. revisión de la configuración	338
17.5.3. pruebas ALFA y BETA	
17.6. Prueba del sistema	
17.6.1. prueba de recuperación	
17.6.2. prueba de seguridad	339
17.6.3. prueba de resistencia	
17.6.4. prueba de rendimiento	
17.7. El arte de la depuración	
17.7.1. el proceso de depuración	340
17.7.2. consideraciones psicológicas	
17.7.3. enfoques de la depuración	341
17.8. Resumen	342
Referencias	
Problemas y puntos a considerar	343
Otras lecturas y fuentes de información	344
Capitulo 18. Métricas técnicas del software	
18.1. Calidad del software	345
18.1.1. factores de calidad de McCaLL	346
18.1.2. FURPS	347
18.1.3. factores de calidad ISO 9126	348
18.2. Una estructura para las métricas técnicas del software	
18.2.1. el reto de las métricas técnicas	349
18.2.2. principios de medición	

18.2.3. características fundamentales de las métricas del software	350
18.3. Métricas del modelo de análisis	
18.3.1. métricas basadas en la función	351
18.3.2. la métrica BANG	352
18.3.3. métricas de localidad de la especificación	353
18.4. Métricas del modelo de diseño	
18.4.1. métricas del diseño de alto nivel	354
18.4.2. métricas de diseño a nivel de componentes	356
18.4.3. métricas de diseño de interfaz	
18.5. métricas del código fuente	358
18.6. métricas para pruebas	360
18.7. métricas de mantenimiento	
18.8. Resumen	361
Referencias	362
Problemas y puntos a considerar	363
Parte IV. Ingeniería del software orientada a objetos	365
Capítulo 19. Conceptos y principios orientados a objetos	
19.1. El paradigma orientado a objetos	367
19.2. Conceptos de orientación a objetos	368
19.2.1. clases y objetos	369
19.2.2. atributos	
19.2.3. operaciones, metodos y servicios	371
19.2.4. mensajes	
19.2.5. encapsulamiento, herencia y polimorfismo	373
19.3. Identificación de los elementos de un modelo de objetos	
19.3.1. Identificación de clases y objetos	375
19.3.2. Especificación de atributos	377
19.3.3. definición de operaciones	378
19.3.4. fin de la definición del objeto	
19.4. Gestión de proyectos de software orientado a objetos	379
19.4.1. el marco de proceso común para OO	
19.4.2. métricas y estimación en proyectos orientados a objetos	380
19.4.3. un enfoque OO para estimaciones y planificación	
19.4.4. seguimiento del progreso en un proyecto orientado a objetos	382
19.5. Resumen	
Referencias	383
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	384
Capítulo 20. Análisis orientado a objetos	
20.1. Análisis orientado a objetos	387
20.1.1. enfoques convencionales y enfoques OO	
20.1.2. el panorama del AOO	388
20.2. Análisis del dominio	
20.2.1. Análisis de reusabilidad y del dominio	390
20.2.2. el proceso de análisis del dominio	391
20.3. Componentes genéricos del modelo de análisis orientado a objetos OO	392
20.4. El proceso de AOO	
20.4.1. casos de uso o de utilización	393

20.4.2. modelado de clases-responsabilidades-colaboradores	394
20.4.3. definición de estructuras y jerarquías	
20.4.4. definición de subsistemas	398
20.5. El modelo objeto-relación	400
20.6. El modelo objeto-comportamiento	
20.6.1. identificación de sucesos con casos de uso	401
20.6.2. representación de estados	402
20.7. Resumen	
Referencias	404
Problemas y puntos a considerar	405
Otras lecturas y fuentes de información	406
Capítulo 21. Diseño orientado a objetos	
21.1. diseño para sistemas orientados a objetos	407
21.1.1. enfoque convencional y el enfoque Vs. OO	408
21.1.2. asuntos del diseño	409
21.1.3. la visión del DOO	410
21.2. Los componentes genéricos del modelo de diseño orientado a objetos	412
21.3. El proceso de diseño de sistema	
21.3.1. participación el modelo de análisis	413
21.3.2. concurrencia y asignación subsistemas	
21.3.3. el componente para la gestión de tareas	414
21.3.4. el componente para la gestión de datos	
21.3.5. el componente para la gestión de recursos	415
21.3.6. el componente para la interfaz hombre-maquina	
21.3.7. comunicación entre subsistemas	
21.4. el proceso de diseño de objeto	416
21.4.1. descripciones de objetos	
21.4.2. diseño de algoritmos y estructuras de datos	417
21.4.3. componentes de programas e interfaces	418
21.5. Patrones de diseño	
21.5.1. descripción de un patrón de diseño	420
21.5.2. uso de patrones en el diseño	
21.6. Programas orientada a objetos	421
21.7. Resumen	
Referencias	
Problemas y puntos a considerar	422
Otras lecturas y fuentes de información	423
Capítulo 22: Pruebas orientadas a objetos	425
22.1. Ampliando la visión de las pruebas	
22.2. modelos de AOO y DOO	426
22.2.1. concreción de los modelos de AOO y DOO	
22.2.2. consistencia de los modelos de AOO y DOO	
22.3. estrategias de pruebas a objetos	427
22.3.1. pruebas de unidad en el contexto de OO	
22.3.2. pruebas de integración en el contexto OO	428
22.3.3. pruebas de validación en un contexto OO	
22.4. Diseño de casos de prueba para software OO	
22.4.1. implicaciones de los conceptos de OO para el diseño de casos	429

de prueba	
22.4.2. aplicabilidad de los métodos convencionales de diseño de casos de prueba	
22.4.3. pruebas basadas en fallas	
22.4.4. el impacto de la programación OO en la realización de pruebas	430
22.4.5. casos de prueba y jerarquía de clases	
22.4.6. diseño de pruebas basadas en el escenarios	431
22.4.7. probando la estructuras superficial profunda	432
22.5. Métodos de prueba aplicables al nivel de clase	
22.5.1. pruebas aleatorias para clases OO	433
22.5.2. prueba de partición al nivel de clase	
22.6. Diseño de casos de prueba interclases	
22.6.1. prueba de múltiples clases	434
22.6.2. pruebas derivadas de modelo de comportamiento	435
22.7. Resumen	
Referencias	436
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	437
Capítulo 23. Métricas técnicas para sistemas orientadas a objetos	
23.1. objetivo de las métricas orientadas a objetos	439
23.2. características distintivas	
23.2.1. Localización	
23.2.2. encapsulamiento	
23.2.3. ocultamiento de información	440
23.2.4. herencia	
23.2.5. abstracción	
23.3. Métricas para el modelo de diseño orientados a objeto	
23.4. Métricas orientadas a clases	441
23.4.1. el conjunto de métricas CK	
23.4.2. Métricas propuestas por Lorenz y Kidd	443
23.5. Métricas orientadas a operaciones	
23.6. Métricas para pruebas orientadas a objetos	444
23.7. Métricas para proyectos orientados a objetos	445
23.8. Resumen	
Referencias	446
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	447
Parte V. Temas avanzados en ingeniería del software	449
Capítulo 24. Métodos formales	
24.1. Conceptos básicos	451
24.1.1. deficiencias de los enfoques menos formales	452
24.1.2. las matemáticas en el desarrollo del software	
24.1.3. conceptos de los métodos formales	453
24.2. Preliminares matemáticos	
24.2.1. conjuntos y especificación constructiva	457
24.2.2. operadores de conjuntos	458
24.2.3. operadores lógicos	
24.2.4. sucesiones	459
24.3. aplicación de la notación matemática para la especificación formal	460

24.4. Lenguajes formales de especificación	461
24.5. utilización de Z para representar un componente de software	462
24.6. los diez mandamientos de los métodos formales	463
24.7. Métodos formales: el futuro	464
24.4. Especificación algebraica	
24.8. Resumen	
Referencias	465
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	466
Capítulo 25. Ingeniería del software de sala limpia	
25.1. el enfoque de sala limpia	469
25.1.1. la estrategia de sala limpia	470
25.1.2. ¿Qué hace diferente la sala limpia?	472
25.2. Especificación funcional	473
25.2.1. especificación de caja negra	
25.2.2. especificación de caja de estados	474
25.2.3. especificación de caja transparente	
25.3. Refinamiento y verificación del diseño	
25.3.1. refinamiento y verificación del diseño	475
25.3.2. ventajas de la verificación del diseño	477
25.4. comprobación de sala limpia	478
25.4.1. comprobación estadística de casos prácticos	
25.4.2. certificación	479
Resumen	480
Referencias	
Problemas y puntos a considerar	481
Otras lecturas y fuentes de información	482
Capítulo 26. Reutilización del software	
26.1. asuntos de gestión	485
26.1.1. dificultades para la reutilización	
26.1.2. una analogía con el hardware	486
26.1.3. algunas sugerencias para establecer un enfoque de la reutilización	487
26.2. el proceso de reutilización	
26.2.1. artefactos reutilizables	488
26.2.2. un modelo de procesos	
26.3. ingeniería de dominios	489
26.3.1. el proceso de análisis del dominios	490
26.3.2. funciones de caracterización	491
26.3.3. modelado estructural y puntos de estructura	492
26.4. construcción de componentes reutilizables	
26.4.1. análisis y diseño para la reutilización	493
26.4.2. métodos de construcción	
26.4.3. desarrollo basado en componentes	494
26.5. clasificación y recuperación de componentes	
26.5.1. descripción de componentes reutilizables	495
26.5.2. el entorno de reutilización	497
26.6. economía de la reutilización del software	
26.6.1. impacto sobre la calidad, la productividad y el coste	498

26.6.2. análisis de coste empleando puntos de estructura	
26.6.3. métricas de reutilización	499
26.7. Resumen	
Referencias	500
Problemas y puntos a considerar	501
Otras lecturas y fuentes de información	502
Capítulo 27. Reingeniería	
27.1. Reingeniería de procesos de negocio	505
27.1.1. procesos de negocios	506
27.1.2. principios de reingeniería de procesos de negocio	507
27.1.3. un modelo de BPR	508
27.1.4. advertencias	
27.2. Reingeniería del software	509
27.2.1. mantenimiento del software	
27.2.2. un modelo de procesos de reingeniería del software	510
27.3. Ingeniería inversa	512
27.3.1. ingeniería inversa para comprender el procesamiento	513
27.3.2. ingeniería inversa para comprender los datos	514
27.3.3. ingeniería inversa de interfaces de usuario	515
27.4. Reestructuración	
27.4.1. Reestructuración del código	516
27.4.2. Reestructuración de los datos	
27.5. Ingeniería progresiva (directa)	517
27.5.1. Ingeniería directa para arquitecturas cliente/servidor	518
27.5.2. Ingeniería directa para arquitectura orientadas a objetos	
27.5.3. ingeniería progresiva para interfaces de usuario	519
27.6. economía de la reingeniería	520
27.7. Resumen	
Referencias	521
Problemas y puntos a considerar	522
Otras lecturas y fuentes de información	523
Capítulo 28. Ingeniería del software cliente/servidor	
28.1. Estructura de los sistemas cliente/servidor	525
28.1.1. componentes de software para sistemas C/S	526
28.1.2. Distribución de componentes de software	527
28.1.3. líneas generales para distribuir componentes de aplicaciones	
28.1.4. enlazado de componentes de software	528
28.1.5. software intermedio (middleware) y arquitectura de redistribución de objetos	529
28.2. ingeniería de software para sistemas C/S	
28.3. problemas de modelado de análisis	530
28.4. diseño para sistemas C/S	
28.4.1. enfoques de diseño convencionales	
28.4.2. diseño de bases de datos	531
28.4.3. visión general de un enfoque de diseño	533
28.4.4. iteración del diseño de procesos	
28.5. problemas de prueba	534
28.5.1. estrategia general de pruebas C/S	535
28.5.2. tácticas de pruebas C/S	536

28.6. Resumen	
Referencias	537
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	538
Capítulo 29. Ingeniería del software asistida por computadora	
29.1. ¿Qué significa case?	541
29.2. bloques básicos de CASE	542
29.3. Una taxonomía de herramientas CASE	543
29.4. entornos CASE integrados	546
29.5. la arquitectura de integración	547
29.6. El repositorio CASE	548
29.6.1. el papel del deposito en I-CASE	
29.6.2. características y contenidos	549
29.7. Resumen	552
Referencias	
Problemas y puntos a considerar	553
Otras lecturas y fuentes de información	554
Capítulo 30. El futuro	
30.1. importancia del software, segunda arte	555
30.2. el ámbito del cambio	556
30.3. las personas y la forma en que constituyen sistemas	557
30.4. el nuevo proceso del software	558
30.5. nuevos modos para representar la información	559
30.6. la tecnología como impulsor	560
30.7. comentario final	
Referencias	562
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	563
Apéndice. Siglas Castellano/Ingles-Ingles/Castellano	565
Índice	571