

## INDICE

<b>Acerca del autor</b>	XXIII
<b>Prefacio</b>	XXV
<b>Prologo a la cuarta edición en español</b>	XXIX
<b>Prologo a la quinta edición en español</b>	XXXIII
<b>Utilización del libro</b>	XXXVII
<b>Parte Primera: el producto y el proceso</b>	
<b>Capítulo 1: El producto</b>	3
1.1. la evolución del software	4
1.2. el software	
1.2.1. características del software	5
1.2.2. aplicaciones del software	6
1.3. Software: ¿una crisis en el horizonte?	
1.4. mitos del software	8
Resumen	
Referencias	10
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	11
<b>Capítulo 2: El proceso</b>	13
2.1. ingeniería del software: una tecnología estratificada	
2.1.1. proceso, métodos y herramientas	14
2.2. el proceso del software	16
2.3. modelos de proceso del software	19
2.4. el modelo lineal secuencial	20
2.5. el modelo de construcción de prototipos	21
2.6. el modelo DRA	22
2.7. modelos evolutivos de proceso del software	
2.7.1. el modelo incremental	23
2.7.2. el modelo espiral	24
2.7.3. el modelo espiral WINWIN (Victoria & Victoria)	26
2.7.4. el modelo de desarrollo concurrente	27
2.8. desarrollo basado en componentes	28
2.9. el modelo de métodos formales	
2.10. técnicas de cuarta generación	29
2.11. tecnologías de proceso	30
2.12. producto y proceso	
Resumen	31
Referencias	
Problemas y puntos a considerar	32
Otras lecturas y fuentes de información	33
<b>Parte segunda: Gestión de proyectos de software</b>	
<b>Capítulo 3: Conceptos sobre gestión de proyectos</b>	37
3.1. el espectro de la gestión	
3.1.1. personal	38
3.1.2. producto	
3.1.3. proceso	
3.1.4. proyecto	
3.2. personal	39

3.2.1. los participantes	
3.2.2. los jefes de equipo	
3.2.3. el equipo de software	40
3.2.4. aspectos sobre la coordinación y la comunicación	43
3.3. producto	
3.3.1. ámbito del software	44
3.3.2. descomposición del problema	
3.4. proceso	45
3.4.1. maduración del producto y del proceso	46
3.4.2. descomposición del proceso	47
3.5. proyecto	48
3.6. el principio W5HH	
3.7. prácticas críticas	49
Resumen	
Referencias	50
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	51
<b>Capítulo 4: Proceso de software y métricas de proyectos</b>	<b>53</b>
4.1. medidas, métricas e indicadores	54
4.2. métricas en el proceso y dominios del proyecto	
4.2.1. métricas en el proceso y mejoras en el proceso del software	55
4.2. métricas del proyecto	
4.3. mediciones del software	58
4.3.1. métricas orientadas al tamaño	59
4.3.2. métricas orientadas a la función	
4.3.3. métricas ampliadas de punto de función	61
4.4. reconciliación de los diferentes enfoques de métricas	62
4.5. métricas para la calidad del software	
4.5.1. visión general de los factores que afectan a la calidad	63
4.5.2. medida de la calidad	
4.5.3. eficacia de la eliminación de efectos	64
4.6. integración de las métricas dentro del proceso de ingeniería del software	65
4.6.1. argumentos para las métricas del software	
4.6.2. establecimiento de una línea base	
4.6.3. colección de métricas, computo y evaluación	66
4.7. el desarrollo de la métrica y de la OPM (objetivo, pregunta, métrica)	67
4.8. variación de la gestión: control de procesos estadísticos	69
4.9. métrica organizaciones pequeñas	71
4.10. establecimiento de un programa de métricas de software	72
Resumen	73
Referencias	74
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	75
<b>Capítulo 5: Planificación sobre la estimación</b>	<b>78</b>
5.2. objetivos de la planificación del proyectos	
5.3. ámbito del software	79
5.3.1. obtención de la información necesarias para el ámbito	

5.3.2. viabilidad	
5.3.3. un ejemplo de ámbito	80
5.4. recursos	
5.4.1. recursos humanos	82
5.4.2. recursos de software reutilizables	
5.4.3. recursos de entorno	83
5.5. estimación del proyecto de software	84
5.6. técnicas de descomposición	
5.6.1. tamaño del software	85
5.6.2. estimación basad en el problema	86
5.6.3. un ejemplo de estimación basada en LDC	87
5.6.4. un ejemplo de estimación basada en PF	88
5.6.5. estimación basada en el proceso	
5.6.6. un ejemplo de estimación basada en el proceso	89
5.7. modelos empíricos de estimación	
5.7.1. la estructura de los modelos de estimación	90
5.7.2. el modelo COCOMO	91
5.7.3. la ecuación del software	
5.8. la decisión de desarrollo-comprar	92
5.8.1. creación de un árbol de decisiones	93
5.8.2. subcontratación (outsourcing)	
5.9. Herramientas automáticas de estimación	94
Resumen	
Referencias	95
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	96
<b>Capítulo 6. Análisis y gestión del riesgo</b>	97
6.1. Estrategias de riesgo proactivas vs. reactivas	
6.2. riesgo del software	98
6.3. identificación del riesgo	99
6.3.1. evaluación global del riesgo del proyecto	100
6.3.2. componentes y controladores del riesgo	
6.4. proyección del riesgo	101
6.4.1. desarrollo de una tabla de riesgo	
6.4.2. evaluación del imanto del riesgo	
6.4.3. evaluación del riesgo	103
6.5. refinamiento del riesgo	104
6.6. reducción, supervisión y gestión del riesgo	105
6.7. riesgos y peligros para la seguridad	106
6.8. el plan RSGR	
Resumen	107
Referencias	
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	108
<b>Capítulo 7: Planificación temporal y seguimiento del proyecto</b>	111
7.1. conceptos básicos	
7.1.1. comentarios sobre “los retrasos”	112
7.1.2. principio básicos	113
7.2. la relación entre las personas y el esfuerzo	114

7.2.1. un ejemplo	115
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	116
7.3.1. grado de rigor	117
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	119
7.4. Selección de las tareas de ingeniería del software	
7.5. refinamiento de las tareas principales	120
7.6. definir una red de tareas	121
7.7. la planificación temporal	122
7.7.1. gráficos de tiempo	123
7.7.2. seguimiento de la planificación temporal	124
7.8. Análisis de valor ganado	125
7.9. Seguimiento del error	126
7.10. El plan del proyecto	127
Resumen	
Referencias	128
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	129
<b>Capítulo 8: Garantía de calidad del software (SOA/GCS)</b>	131
8.1. conceptos de calidad	132
8.1.1. calidad	
8.1.2. control de calidad	
8.1.3. garantía de calidad	
8.1.4. coste de calidad	133
8.2. la tendencia de la calidad	134
8.3. garantía de calidad del software	135
8.3.1. problemas de fondo	
8.3.2. actividades de SQA	136
8.4. Revisiones del software	137
8.4.1. impacto de los efectos del software sobre el coste	
8.4.2. amplificación eliminación de defectos	138
8.5. Revisiones técnicas formales	
8.5.1. la reunión de revisión	139
8.5.2. registros e informe de la revisión	140
8.5.3. directrices para la revisión	
8.6. Garantía de la calidad estadística	141
8.7. Fiabilidad del software	143
8.7.1. medidas de fiabilidad y de disponibilidad	
8.7.2. seguridad del software	144
8.8. Prueba de errores para el software	145
8.9. El estándar de calidad ISO 90001	146
8.10. El plan de SQA	147
Resumen	148
Referencias	
Problemas y puntos a considerar	149
Otras lecturas y fuentes de información	150

<b>Capítulo 9: Gestión de la configuración del software (GCS/SCM)</b>	151
9.1. Gestión de la configuración del software	
9.1.1. líneas base	152
9.1.2. elementos de configuraciones del software	153
9.2. El proceso de GCS	
9.3. Identificación de objetos en la configuración del software	154
9.4. Control de versiones	155
9.5. Control de cambios	156
9.6. Auditoría de la configuración	158
9.7. Informes de estado	
Resumen	159
Referencias	
Problemas y puntos a considerar	160
Otras lecturas y fuentes de información	161
<b>Parte tercera: Métodos convencionales es para la ingeniería del software</b>	165
<b>Capítulo 10: Ingeniería de sistemas</b>	
10.1. sistemas basados en computadora	
10.2. la jerarquía de la ingeniería de sistemas	167
10.2.1. modelado del sistema	
10.2. simulación del sistema	168
10.3. ingeniería de proceso de negocio: una visión general	169
10.4. ingeniería de producto: una visión general	
10.5. ingeniería de requisitos	171
10.5.1. identificación de requisitos	172
10.5.2. análisis y negociación de requisitos	
10.5.3. especificación de requisitos	173
10.5.4. Modelado del sistema	
10.5.5. validación de requisitos	174
10.5.6. gestión de requisitos	
10.6. Modelado del sistema	175
Resumen	
Referencias	178
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	179
<b>Capítulo 11: Conceptos y principios del análisis</b>	181
11.1. análisis de requisitos	182
11.2. identificación de requisitos para el software	
11.2.1. inicio del proceso	183
11.2.2. técnicas para facilitar las especificaciones de una aplicación	184
11.2.3. despliegue de la función de calidad	
11.2.4. casos de uso	186
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	191
11.4. Creación de prototipos del software	
11.4.1. selección del enfoque de creación de prototipos	192

11.4.2. métodos y herramientas para el desarrollo de prototipos	
11.5. Especificación	193
11.5.1. principios de la especificación	
11.5.2. representación	194
11.5.3. la especificación de los requisitos del software	
11.6. Revisión de la especificación	195
Resumen	
Referencias	196
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	197
<b>Capítulo 12: Modelado del análisis</b>	199
12.1. Breve historia	
12.2. los elementos del modelo de análisis	200
12.3. modelado de datos	
12.3.1. objetos de datos, atributos y relaciones	201
12.3.2. cardinalidad y moralidad	203
12.3.3. diagramas entidad-relación	204
12.4. Modelado funcional y flujo de información	205
12.4.1. diagramas de flujo de datos	206
12.4.2. ampliaciones para sistemas de tiempo real	
12.4.3. ampliaciones de Ward y Mellor	207
12.4.4. ampliaciones de Hatley y Pirbhal	208
12.5. Modelado del comportamiento	209
12.6. Mecanismos del análisis estructurado	
12.6.1. creación de un diagrama entidad-relación	210
12.6.2. creación de un modelo de flujo de datos	211
12.6.3. creación de un modelo de flujo de control	213
12.6.4. la especificación de control	
12.6.5. la especificación del proceso	214
12.7. El diccionario de datos	215
12.8. Otros métodos clásicos de análisis	
Resumen	216
Referencias	
Problemas y puntos a considerar	217
Otras lecturas y fuentes de información	218
<b>Capítulo 13: Conceptos y principios de diseño</b>	219
13.1. diseño de software e ingeniería del software	220
13.2. el proceso de diseño	
13.2.1. diseño y calidad del software	221
13.2.2. la evolución del diseño del software	
13.3. Principios del diseño	222
13.4. Conceptos del diseño	
13.4.1. abstracción	223
13.4.2. refinamiento	
13.4.3. modularidad	224
13.4.4. arquitectura del software	
13.4.5. jerarquía de control	226
13.4.6. división estructural	227
13.4.7. estructura de datos	228

13.4.8. procedimiento de software	229
13.4.9. ocultación de información	
13.5. Diseño modular efectivo	
13.5.1. independencia funcional	230
13.5.2. cohesión	
13.5.3. acoplamiento	
13.6. Heurística de diseño para una modularidad efectiva	231
13.7. El modelo del diseño	233
13.8. Documentación de el diseño	
Resumen	234
Referencias	
Problemas y puntos a considerar	236
Otras lecturas y fuentes de información	236
<b>Capítulo 14: Diseño arquitectónico</b>	237
14.1. arquitectura del software	238
14.1.1. ¿Qué es arquitectura?	
14.1.2. ¿Por que es importante la arquitectura?	
14.2. Diseño de datos	239
14.2.1. modelado de datos, estructuras de datos, bases de datos y almacén de datos	
14.2.2. diseño de datos a nivel de componentes	
14.3. Estilos arquitectónicos	241
14.3.1. una breve taxonomía de estilos y patrones	
14.3.2. organización y refinamiento	243
14.4. Análisis de diseños arquitectónicos alternativos	
14.4.1. un método de análisis de compromiso para la arquitectura	244
14.4.2. guía cuantitativa para el diseño arquitectónico	
14.4.3. complejidad arquitectónica	
14.5. Conversión de los requisitos en una arquitectura del software	245
14.5.1. flujo de transformación	246
14.5.2. flujo de transacción	
14.6. Análisis de las transformaciones	247
14.6.1. un ejemplo	
14.6.2. pasos del diseño	
14.7. Análisis de las transacciones	252
14.7.1. un ejemplo	
14.7.2. pasos del diseño	
14.8. Refinamiento del diseño arquitectónico	255
Resumen	256
Referencias	
Problemas y puntos a considerar	257
Otras lecturas y fuentes de información	258
<b>Capítulo 15: Diseño de a interfaz de usuario</b>	259
15.1. las reglas de oro	260
15.1.1. DAR el control al usuario	
15.1.2. reducir la carga de memoria del usuario	
15.1.3 construcción de una interfaz consistente	261
15.2. Diseño de a interfaz de usuario	262
15.2.1. modelos de diseño de la interfaz	

15.2.2. el proceso de diseño de la interfaz de usuario	263
15.3. Análisis y modelado de tareas	
15.4. Actividades del diseño de la interfaz	264
15.4.1. definición de objetos y acciones de la interfaz	265
15.4.2. problemas del diseño	266
15.5. Herramientas de implementación	
15.6. Evaluación del diseño	268
Resumen	
Referencias	270
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	271
<b>Capítulo 16: Diseño a nivel de componentes</b>	273
16.1. Programación estructurada	
16.1.1. notación grafica del diseño	274
16.1.2. notación tabular de diseño	
16.1.3. lenguaje de diseño de programas	276
16.1.4. un ejemplo de LDP	277
16.2. Comparación de notaciones de diseño	278
Resumen	
Referencias	279
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	280
<b>Capítulo 17. Técnicas de prueba del software</b>	281
17.1. Fundamentos de las pruebas del software	
17.1. objetivos de las pruebas	282
17.1.2. principios de las pruebas	
17.1.3. facilidad de prueba	283
17.2. Diseño de casos de prueba	285
17.3. Prueba de caja blanca	
17.4. Prueba del camino básico	286
17.4.1. notación de grafico de flujo	
17.4.2. complejidad ciclomática	287
17.4.3. obtención de casos de prueba	288
17.4.4. matrices de grafos	290
17.5. Prueba de la estructura de control	
17.5.1. prueba de condición	291
17.5.2. prueba del flujo de datos	292
17.5.3. pruebas de bucles	293
17.6. Prueba de caja negra	
17.6.1. métodos de prueba basados en grafos	294
17.6.2. partición equivalente	296
17.6.3. análisis de valores limite	
17.6.4. prueba de comparación	297
17.6.5. prueba de la tabal ortogonal	298
17.7. Prueba de entornos especializados, arquitecturas y aplicaciones	
17.7.1. prueba de interfaces graficas de usuario (IGUs)	299
17.7.2. prueba de arquitectura cliente/servidor	
17.7.3. prueba de la documentación y facilidades de ayuda	300
17.7.4. Prueba de sistemas de tiempo-real	



Resumen	301
Referencias	
Problemas y puntos a considerar	302
Otras lecturas y fuentes de información	303
<b>Capítulo 18: Estrategias de pruebas de software</b>	<b>305</b>
18.1. Un enfoque estratégico para las pruebas del software	
18.1.1. verificación y validación	306
18.1.2. organización para las pruebas del software	
18.1.3. una estrategias de prueba del software	307
18.1.4. criterios para completar la prueba	308
18.2. Aspectos estratégicos	309
18.3. Prueba de unidad	
18.3.1. consideraciones sobre la prueba de unidad	310
18.3.2. procedimientos de prueba de unidad	311
18.4. Prueba de integración	
18.4.1. integración descendente	312
18.4.2. integración ascendente	313
18.4.3. prueba de regresión	
18.8.4. prueba de humo	314
18.8.5. Comentarios sobre la prueba de integración	315
18.5. Prueba de validación	
18.5.1. criterios de la prueba de validación	316
18.5.2. revisión de la configuración	
18.5.3. pruebas ALFA y BETA	
18.6. Prueba del sistema	
18.6.1. prueba de recuperación	317
18.6.2. prueba de seguridad	
18.6.3. prueba de resistencia (STRESS)	
18.6.4. prueba de rendimiento	318
18.7. El arte de la depuración	
18.7.1. el proceso de depuración	
18.7.2. consideraciones psicológicas	319
18.7.3. enfoques de la depuración	320
Resumen	
Referencias	321
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	322
<b>Capítulo 19: Métricas técnicas del software</b>	<b>323</b>
19.1. Calidad del software	
19.1.1. factores de calidad de McALL	324
19.1.2. FURPS	325
19.1.3. factores de calidad ISO 9126	
19.1.4. La transición a una visión cuantitativa	326
19.2. Una estructura para las métricas técnicas del software	
19.2.1. el reto de las métricas técnicas	327
19.2.2. principios de medición	
19.2.3. características fundamentales de las métricas del software	328
19.3. Métricas del modelo de análisis	
19.3.1. métricas basadas en la función	329

19.3.2. la métrica BANG	330
19.3.3. métricas de localidad de la especificación	331
19.4. Métricas del modelo de diseño	
19.4.1. métricas del diseño arquitectónico	332
19.4.2. métricas de diseño a nivel de componentes	333
19.4.3. métricas de diseño de interfaz	335
19.5. métricas del código fuente	336
19.6. métricas para pruebas	337
19.7. métricas de mantenimiento	
Resumen	338
Referencias	
Problemas y puntos a considerar	339
Otras lecturas y fuentes de información	340
<b>Parte cuarta: Ingeniería del software orientada a objetos</b>	
<b>Capítulo 20: Conceptos y principios orientados a objetos</b>	343
20.1. El paradigma orientado a objetos	344
20.2. Conceptos de orientación a objetos	345
20.2.1. clases y objetos	346
20.2.2. atributos	
20.2.3. operaciones, metodos y servicios	347
20.2.4. mensajes	
20.2.5. encapsulamiento, herencia y polimorfismo	348
20.3. Identificación de los elementos de un modelo de objetos	
20.3.1. Identificación de clases y objetos	350
20.3.2. Especificación de atributos	
20.3.3. definición de operaciones	353
20.3.4. fin de la definición del objeto	
20.4. Gestión de proyectos de software orientada a objetos	354
20.4.1. el marco de proceso común para OO	355
20.4.2. métricas y estimación en proyectos orientados a objetos	356
20.4.3. un enfoque OO para estimaciones y planificación	357
20.4.4. seguimiento del progreso en un proyecto orientado a objetos	
Resumen	358
Referencias	
Problemas y puntos a considerar	359
Otras lecturas y fuentes de información	360
<b>Capítulo 21: Análisis orientado a objetos</b>	361
21.1. Análisis orientado a objetos	
21.1.1. enfoques convencionales y enfoques OO	362
21.1.2. el panorama del AOO	
21.1.3. un enfoque unificado para el AOO	363
21.4. Análisis del dominio	
21.2.1. Análisis de reutilización y del dominio	364
21.2.2. el proceso de análisis del dominio	365
21.3. Componentes genéricos del modelo de análisis OO	366
21.4. El proceso de AOO	
21.4.1. casos de uso	367
21.4.2. modelado de clases-responsabilidades-colaboradores	368
21.4.3. definición reestructuras y jerarquías	371

21.4.4. definición de subsistemas	372
21.5. El modelo objeto-relación	373
21.6. El modelo objeto-comportamiento	
21.6.1. identificación de sucesos con casos de uso	374
21.6.2. representación de estados	375
Resumen	376
Referencias	
Problemas y puntos a considerar	377
Otras lecturas y fuentes de información	378
<b>Capítulo 22: Diseño orientado a objetos</b>	379
22.1. diseño para sistemas orientados a objetos	
22.1.1. enfoque convencional Vs. OO	380
22.1.2. aspectos del diseño	381
22.1.3. el panorama de OO	382
22.1.4. un enfoque unificado para el DOO	383
22.2. El proceso de diseño de sistema	
22.2.1. participación el modelo de análisis	384
22.2.2. asignación de concurrencia y subsistemas	385
22.2.3. componente de administración de tareas	
22.2.4. componente de interfaz de usuario	386
22.2.5. componente de la administración de datos	
22.2.6. componente de gestión de recursos	
22.2.7. comunicaciones entre subsistemas	387
22.3. Proceso de diseño de objeto	
22.3.1. descripción de objetos	388
22.3.2. diseño de algoritmos y estructuras de datos	389
22.4. Patrones de diseño	
22.4.1. ¿Qué es un patrón de diseño?	390
22.4.2. otro ejemplo de un patrón	
22.4.3. un ejemplo final de un patrón	391
22.4.4. descripción de un patrón de diseño	
22.4.5. el futuro de los patrones	392
22.5. Programación orientada a objetos	
22.5.1. el modelo de clases	393
22.5.2. generalización	
22.5.3. agregación y composición	394
22.5.4. asociaciones	
22.5.5. casos de uso	395
22.5.6. colaboraciones	396
22.5.7. diagramas de estado	397
22.6. Caso de estudio. Libros en línea	398
22.6.1. libros-en línea	399
22.7. Programación orientada a objetos	400
Resumen	
Referencias	404
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	405
<b>Capítulo 23: Pruebas orientadas a objetos</b>	407
23.1. Ampliando la visión de las pruebas	408

23.2. Pruebas de los modelos de AOO y DOO	409
23.2.1. exactitud de los modelos de AOO y DOO	
23.2.2. consistencia de los modelos de AOO y DOO	
23.3. Estrategias de pruebas orientadas a objetos	410
23.3.1. las pruebas de unidad en el contexto de LA OO	
23.3.2. las pruebas de integración en el contexto OO	
23.3. pruebas de validación en un co0ntexto OO	411
23.4. Diseño de casos de prueba para software OO	
23.4.1. implicaciones de los conceptos de OO al diseño de casos de prueba	412
23.4.2. aplicabilidad de los métodos convencionales de diseño de casos de prueba	
23.4.3. pruebas basadas en errores	
23.4.4. el impacto de la programación en las pruebas	413
23.4.5. casos de prueba y jerarquía de clases	
23.4.6. diseño de pruebas basadas ene l escenario	414
23.4.7. las estructuras de pruebas superficiales y profundas	415
23.5. Métodos de prueba aplicables al nivel de clases	
23.5.1. la verificación alazar para clases	416
23.5.2. prueba de partición al nivel de clases	
23.6. Diseño de casos de prueba interclases	
23.6.1. prueba de múltiples clases	417
23.6.2. prueba derivada de los modelos de comportamiento	418
Resumen	
Referencias	419
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	420
<b>Capítulo 24: Métricas técnicas para sistemas orientadas a objetos</b>	421
24.1. el propósito de las métricas orientadas a objetos	
24.2. características distintivas de las métricas orientadas a objetos	422
24.3. Localización	
24.2.2. encapsulación	
24.2.3. ocultación de información	
24.2.4. herencia	423
24.2.5. abstracción	
24.3. Métricas para el modelo de diseño OO	
24.4. Métricas orientadas a clase OO	424
24.4.1. la serie de métricas CK	425
24.4.2. Métricas propuestas por Lorenz y Kidd	426
24.4.3. la colección de métricas MDOO	427
24.5. Métricas orientadas a operaciones	
24.6. Métricas para pruebas orientadas a objetos	428
24.7. Métricas para proyectos orientados a objetos	429
Resumen	
Referencias	430
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	431
<b>Parte quinta: temas avanzados en ingeniería del software</b>	
<b>Capítulo 25: Métodos formales</b>	435

25.1. Conceptos básicos	
25.1.1. deficiencias de los enfoques menos formales	436
25.1.2. matemáticas en el desarrollo del software	437
25.1.3. conceptos de los métodos formales	438
25.2. Preliminares matemáticas	441
25.2.1. conjuntos y especificación constructiva	
25.2.2. operadores de conjuntos	442
25.2.3. operadores lógicos	
25.2.4. sucesiones	443
25.3. aplicación de la notación matemática para la especificación formal	444
25.4. Lenguajes formales de especificación	445
25.5. Uso del lenguaje para representar un componente de software	446
25.6. Métodos formales basados en objetos	447
25.7. Especificación algebraica	450
25.8. Métodos formales concurrentes	452
25.9. los diez mandamientos de los métodos formales	455
25.10. Métodos formales: el futuro	
Resumen	456
Referencias	
Problemas y puntos a considerar	457
Otras lecturas y fuentes de información	458
<b>Capítulo 26: Ingeniería del software de sala limpia</b>	459
26.1. el enfoque de sala limpia	
26.1.1. la estrategia de sala limpia	460
26.1.2. ¿Qué hace diferente la sala limpia?	461
26.2. Especificación funcional	462
26.2.1. especificación de caja negra	
26.2.2. especificación de caja de estado	463
26.2.3. especificación de caja limpia	
26.3. Refinamiento y verificación del diseño	464
26.3.1. refinamiento y verificación del diseño	
26.3.2. ventajas de la verificación del diseño	466
26.4. Prueba de sala limpia	
26.4.1. prueba estadística de casos prácticos	467
26.4.2. certificación	468
Resumen	
Referencias	469
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	470
<b>Capítulo 27. Ingeniería del software basado en componentes</b>	473
27.1. Ingeniería de sistemas basadas en componentes	474
27.2. el proceso de ISBC	475
27.3. Ingeniería del dominio	
27.3.1. el proceso de análisis del dominio	476
27.3.2. funciones de caracterización	477
27.4. Desarrollo basado en componentes	478
27.4.1. cualificación, adaptación y composición de componentes	479
27.4.2. ingeniería de componentes	481

27.4.3. análisis y diseño para la reutilización	
27.5. Clasificación y recuperación de componentes	
27.5.1. descripción de componentes reutilizables	482
27.5.2. el entorno de reutilización	
27.6. Economía de la ISBC	484
27.6.1. impacto en la calidad, productividad y coste	
27.6.2. análisis de coste empleando puntos de estructura	485
27.6.3. métricas de reutilización	
Resumen	486
Referencias	487
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	488
<b>Capítulo 28: Ingeniería del software del comercio electrónico cliente/servidor</b>	491
28.1. Introducción	
28.2. Sistemas distribuidos	492
28.2.1. clientes y servidores	
28.2.2. categorías de servidores	
28.2.3. software intermedio (MIDDLEWARE)	494
28.2.4. un ejemplo de software intermedio	495
28.3. Arquitecturas estratificadas	496
28.4. Protocolos	
28.4.1. el concepto	497
28.4.2. IP E ICMP	
28.4.3. POP3	498
28.4.4. el protocolo http	
28.5. Un sistema de comercio electrónico	499
28.5.1. ¿Qué es el comercio electrónico?	
28.5.2. Un sistema típico de comercio electrónico	500
28.6. Tecnologías usadas por el comercio electrónico	
28.6.1. conexiones (SOCKETS)	502
28.6.2. objetos distribuidos	
28.6.3. Espacios	
28.6.4. CGI	503
28.6.5. contenido ejecutable	
28.6.6. paquetes cliente/servidor	
28.7. el diseño de sistemas distribuidos	
28.7.1. correspondencia del volumen de transición con los medios de transmisión	504
28.7.2. mantenimiento de los datos más usados en un almacenamiento rápido	
28.7.3. mantenimiento de los datos cerca de donde se utilizan	
28.7.4. utilización de la duplicación de datos todo lo posible	
28.7.5. eliminar cuellos de botella	505
28.7.6. minimizar la necesidad de un gran conocimiento del sistema	
28.7.7. agrupar datos afines en la misma ubicación	
28.7.8. considerar la utilización de servidores dedicados a funciones frecuentes	
28.7.9. correspondencia de la tecnología con las exigencias de	506

rendimiento	
28.7.10. empleo del paralelismo todo lo posible	
28.7.11. utilización de la comprensión de datos todo lo posible	
28.7.12. diseño para el fallo	
28.7.13. minimizar la latencia	
28.7.14. epílogo	
28.8. Ingeniería de seguridad	
28.8.1. encriptación	507
28.8.2. funciones de compendio de mensajes	
28.8.3. firmas digitales	508
28.8.4. Certificaciones digitales	
28.9. componentes de software para sistemas C/S	
28.9.1. Introducción	509
28.9.2. Distribuciones de componentes de software	
28.9.3. líneas generales para la distribución de componentes de aplicaciones	510
28.9.4. enlazado de componentes de software C/S	511
28.9.5. software intermedio (MIDDLEWARE) y arquitecturas de agentes de solicitud de objetos	
28.10. Ingeniería de software para sistemas C/S	512
28.11. Problemas de modelación de análisis	
28.12. diseño de sistemas C/S	
28.12.1. diseño arquitectónico para sistemas cliente/servidor	513
28.12.2. enfoques de diseño convencionales para software de aplicaciones	514
28.12.3. diseño de bases de datos	
28.12.4. visión general de un enfoque de diseño	515
28.12.5. iteración del diseño de procesos	
28.13. Problemas de las pruebas	516
28.13.1. estrategia general de pruebas C/S	
28.13.2. táctica de pruebas C/S	
Resumen	518
Referencias	
Problemas y puntos a considerar	519
Otras lecturas y fuentes de información	
<b>Capítulo 29. ingeniería Web</b>	<b>521</b>
29.1. Los atributos de aplicaciones basadas en Web	522
29.1.1. atributos de calidad	523
29.1.2. las tecnologías	524
29.2. El proceso de IWEB	
29.3. Un marco de trabajo para la IWEB	525
29.4. Formulación y análisis de sistemas basados en Web	
29.4.1. formulación	526
29.4.2. análisis	
29.5. Diseño para aplicaciones basadas en Web	527
29.5.1. Diseño arquitectónico	528
29.5.2. Diseño de navegación	530
29.5.3. Diseño de la interfaz	531
29.6. Pruebas de las aplicaciones basadas en Web	532

29.7. Problemas de gestión	
29.7.1. el equipo de IWEB	533
29.7.2. gestión de proyecto	534
29.7.3. problemas GCS para la IWEB	536
Resumen	537
Referencias	538
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	539
<b>Capítulo 30: Reingeniería</b>	541
30.1. Reingeniería de procesos de	
30.1.2. principios de reingeniería de procesos de negocios	542
30.1.1. procesos de negocios	
30.1.3. un modelo de RPN	543
30.1.4. advertencias	54
30.2. Reingeniería del software	
30.2.1. mantenimiento del software	544
30.2.2. un modelo de procesos de reingeniería del software	545
30.3. Ingeniería inversa	547
30.3.1. ingeniería inversa para comprender el procesamiento	548
30.3.2. ingeniería inversa para comprender los datos	549
30.3.3. ingeniería inversa de interfaces de usuario	
30.4. Reestructuración	550
30.4.1. Reestructuración del código	
30.4.2. Reestructuración de los datos	
30.5. Ingeniería directa (Forward Engineering)	551
30.5.1. Ingeniería directa para arquitecturas cliente/servidor	552
30.5.2. Ingeniería directa para arquitectura orientadas a objetos	553
30.6. La economía de la reingeniería	554
Resumen	
Referencias	555
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	556
<b>Capítulo 31: Ingeniería del software asistida por computadora</b>	559
31.1. ¿Qué significa case?	
31.2. construcción de bloques básicos para case	560
31.3. Una taxonomía de herramientas case	561
31.4. entornos case integrados	565
31.5. la arquitectura de integración	566
31.6. El repositorio case	
31.6.1. el papel del repositorio en I-CASE	567
31.6.2. características y contenidos	568
Resumen	
Referencias	571
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	572
<b>Capítulo 32: Perspectivas futuras</b>	573
32.1. importancia del software-segunda parte	
32.2. el ámbito del cambio	574
32.3. las personas y la forma en que construyen sistemas	



32.4. El nuevo proceso de ingeniería del software	575
32.5. Nuevos modos de representar la información	576
32.6. Las tecnología como impulsos	577
32.7. Comentario final	
Referencias	578
Problemas y puntos a considerar	
Otras lecturas y fuentes de información	579
<b>Apéndice</b>	<b>581</b>
<b>Índice</b>	<b>589</b>