
CONTENIDO

Prólogo	xxi
PRIMERA PARTE. SOFTWARE. EL PROCESO Y SU GESTION	
1. Software e ingeniería del software	3
1.1. La importancia del software	4
1.1.1. La evolución del software	4
1.1.2. Una perspectiva industrial	7
1.1.3. Una fábrica de software que envejece	8
1.2. El software	10
1.2.1. Características del software	10
1.2.2. Componentes de software	14
1.2.3. Aplicaciones del software	16
1.3. Software: una crisis en el horizonte	18
1.3.1. Problemas	19
1.3.2. Causas	20
1.4. Mitos del software	21
1.5. Paradigmas de la ingeniería del software	24
1.5.1. Ingeniería del software: una definición	25
1.5.2. El ciclo de vida clásico	26
1.5.3. Construcción de prototipos	28
1.5.4. El modelo en espiral	30
1.5.5. Técnicas de cuarta generación	32
1.5.6. Combinación de paradigmas	34
1.6. Una visión genérica de la ingeniería del software	35
1.7. Resumen	38
Referencias	38
Problemas y puntos a considerar	39
Otras lecturas	40

2. Gestión del proyecto: métricas del software	43
2.1. El proceso de gestión del proyecto	43
2.1.1. Comienzo del proyecto de software	44
2.1.2. Medición y métricas	44
2.1.3. Estimación	45
2.1.4. Análisis de riesgos	45
2.1.5. Planificación temporal	46
2.1.6. Seguimiento y control	46
2.2. Métricas para la productividad y la calidad del software	47
2.3. Medición del software	47
2.3.1. Métricas orientadas al tamaño	49
2.3.2. Métricas orientadas a la función	50
2.4. Métricas para la calidad del software	54
2.4.1. Visión general de los factores que afectan a la calidad	54
2.4.2. Medida de la calidad	55
2.5. Reconciliación de las diferentes métricas	56
2.6. Integración de las métricas dentro del proceso de la ingeniería del software	59
2.6.1. Argumentos para las métricas del software	59
2.6.2. Establecimiento de una línea base	60
2.6.3. Recolección, cálculo y evaluación de métricas	61
2.7. Resumen	61
Referencias	65
Problemas y puntos a considerar	65
Otras lecturas	66
3. Gestión del proyecto: estimación	69
3.1. Observaciones sobre la estimación	69
3.2. Objetivos de la planificación del proyecto	71
3.3. Ambito del software	72
3.4. Recursos	75
3.4.1. Recursos humanos	75
3.4.2. Recursos de hardware	76
3.4.3. Recursos de software	76
3.4.4. Reusabilidad	79
3.5. Estimación del proyecto de software	80
3.6. Técnicas de descomposición	81
3.6.1. Estimación de LDC y PF	82
3.6.2. Un ejemplo	84
3.6.3. Estimación del esfuerzo	87
3.6.4. Un ejemplo	88
3.7. Modelos empíricos de estimación	89
3.7.1. COCOMO	90
3.7.2. Modelo de estimación de Putnam	93
3.7.3. Modelos del punto de función	95
3.8. Herramientas automáticas de estimación	95
3.9. Resumen	98
Referencias	98
Problemas y puntos a considerar	99
Otras lecturas	100

4. Gestión del proyecto: planificación	101
4.1. Planificación del proyecto: un repaso	101
4.2. Análisis del riesgo	102
4.2.1. Identificación del riesgo	103
4.2.2. Proyección del riesgo	104
4.2.3. Evaluación del riesgo	106
4.2.4. Gestión y supervisión del riesgo	107
4.3. Planificación temporal del proyecto de software	110
4.3.1. Relaciones gente-trabajo	110
4.3.2. Definición de tareas y paralelismo	112
4.3.3. Distribución de esfuerzos	113
4.3.4. Métodos de planificación temporal	114
4.3.5. Un ejemplo de planificación temporal	115
4.3.6. Seguimiento y control del proyecto	122
4.4. Adquisición de software	122
4.5. Reingeniería del software	127
4.6. Planificación organizativa	128
4.7. El plan del proyecto de software	130
4.8. Resumen	132
Referencias	132
Problemas y puntos a considerar	133
Otras lecturas	135

SEGUNDA PARTE. ANALISIS DE REQUISITOS DEL SISTEMA Y DEL SOFTWARE

5. Ingeniería de sistemas de computadora	139
5.1. Sistemas basados en computadora	140
5.2. Ingeniería de sistemas de computadora	142
5.2.1. Hardware e ingeniería del hardware	145
5.2.2. Software e ingeniería del software	148
5.2.3. Factores humanos e ingeniería humana	153
5.2.4. Bases de datos e ingeniería de bases de datos	154
5.3. Análisis del sistema	155
5.3.1. Identificación de las necesidades	155
5.3.2. Estudio de viabilidad	156
5.3.3. Análisis económico	159
5.3.4. Análisis técnico	164
5.3.5. Asignación y compromisos	165
5.4. Modelización de la arquitectura del sistema	166
5.4.1. Diagramas de arquitectura	166
5.4.2. Especificación de la arquitectura del sistema	172
5.5. Simulación y modelización del sistema	174
5.6. La especificación del sistema	175
5.7. Revisión de la especificación del sistema	175
5.8. Resumen	177
Referencias	177
Problemas y puntos a considerar	178
Otras lecturas	179

6. Fundamentos del análisis de requisitos	181
6.1. Análisis de requisitos	182
6.1.1. Tareas de análisis	182
6.1.2. El analista	184
6.2. Areas de problemas	186
6.3. Técnicas de comunicación	187
6.3.1. Inicio del proceso	187
6.3.2. Técnicas para facilitar la especificación de la aplicación	188
6.4. Principios de análisis	192
6.4.1. El ámbito de información	193
6.4.2. Modelización	194
6.4.3. Partición	195
6.4.4. Planteamientos esencial y de implementación	198
6.5. Construcción de prototipos del software	199
6.5.1. Un guión para la construcción de prototipos	199
6.5.2. Métodos y herramientas para la construcción de prototipos	201
6.6. Especificación	202
6.6.1. Principios de especificación	203
6.6.2. Representación	206
6.6.3. La especificación de requisitos del software	208
6.7. Revisión de la especificación	210
6.8. Resumen	212
Referencias	212
Problemas y puntos a considerar	213
Otras lecturas	214
7. Análisis estructurado y sus ampliaciones	217
7.1. Un poco de historia	218
7.2. Notación básica y sus ampliaciones	218
7.2.1. Diagramas de flujo de datos	219
7.2.2. Ampliaciones para sistemas de tiempo real	222
7.2.3. Ampliaciones de Ward y Mellor	222
7.2.4. Ampliaciones de Hatley y Pirbhai	226
7.2.5. Modelización del comportamiento	229
7.2.6. Ampliaciones para aplicaciones de datos masivos	231
7.3. La mecánica del análisis estructurado	232
7.3.1. Creación de un modelo de flujo de datos	232
7.3.2. Creación de un modelo de flujo de control	236
7.3.3. La especificación de control	239
7.3.4. La especificación de procesamiento	240
7.4. Diccionario de requisitos	241
7.5. Análisis estructurado e ingeniería del software asistida por computadora (CASE)	245
7.6. Resumen	246
Referencias	247
Problemas y puntos a considerar	247
Otras lecturas	249

8. Análisis orientado a los objetos y modelización de datos	251
8.1. Conceptos orientados a los objetos	252
8.1.1. Identificación de objetos	254
8.1.2. Especificación de atributos	258
8.1.3. Definición de las operaciones	259
8.1.4. Comunicación entre objetos	260
8.1.5. Fin de la definición del objeto	261
8.2. Modelo de análisis orientado a los objetos	262
8.2.1. Estructuras de clasificación y ensamblaje	263
8.2.2. Definición de temas	265
8.2.3. Conexiones de instancias y caminos de mensajes	267
8.2.4. AOO y prototipos	269
8.3. Modelización de datos	270
8.3.1. Objetos de datos, atributos y relaciones	270
8.3.2. Diagramas de entidad-relación	273
8.4. Resumen	276
Referencias	276
Problemas y puntos a considerar	277
Otras lecturas	278
9. Técnicas alternativas de análisis y métodos formales	279
9.1. Métodos de análisis de requisitos	280
9.1.1. Características comunes	280
9.1.2. Diferencias entre los métodos de análisis	281
9.2. Métodos orientados a la estructura de los datos	282
9.3. Desarrollo de sistemas estructurados en datos	282
9.3.1. Diagramas de Warnier	283
9.3.2. El método DSED	284
9.3.3. Contexto de la aplicación	285
9.3.4. Funciones de la aplicación	287
9.3.5. Resultados de la aplicación	287
9.4. Desarrollo de sistemas de Jackson	289
9.4.1. Paso de entidad acción	289
9.4.2. Paso de estructura de entidad	291
9.4.3. Paso del modelo inicial	292
9.5. SADT	293
9.6. Técnicas formales de especificación	296
9.6.1. Estado actual de los métodos formales	300
9.6.2. Atributos de los lenguajes de especificación formal	301
9.7. Una especificación formal en Z	302
9.7.1. Acerca del núcleo	303
9.7.2. Documentación	304
9.7.3. Estados del núcleo	305
9.7.4. Procesamiento en segundo plano	310
9.7.5. Manejo de interrupciones	314
9.7.6. Métodos formales: perspectivas	317
9.8. Técnicas automatizadas de análisis de requisitos	317
9.8.1. Software Requirements Engineering Methodology	318
9.8.2. PSL/PSA	319

9.8.3.	TAGS	320
9.8.4.	Entornos de especificación	320
9.8.5.	Herramientas para métodos formales	321
9.8.6.	Técnicas automatizadas. Resumen	322
9.9.	Resumen	322
	Referencias	323
	Problemas y puntos a considerar	324
	Otras lecturas	325

TERCERA PARTE. DISEÑO E IMPLEMENTACION DEL SOFTWARE

10.	Fundamentos del diseño de software	329
10.1.	Ingeniería del software y diseño del software	330
10.2.	El proceso de diseño	331
10.2.1.	Diseño y calidad del software	332
10.2.2.	La evolución del diseño de software	333
10.3.	Fundamentos del diseño	333
10.3.1.	Abstracción	334
10.3.2.	Refinamiento	337
10.3.3.	Modularidad	338
10.3.4.	Arquitectura del software	340
10.3.5.	Jerarquía de control	341
10.3.6.	Estructura de datos	343
10.3.7.	Procedimientos del software	345
10.3.8.	Ocultamiento de información	345
10.4.	Diseño modular efectivo	346
10.4.1.	Tipos de módulos	347
10.4.2.	Independencia funcional	349
10.4.3.	Cohesión	349
10.4.4.	Acoplamiento	351
10.5.	Diseño de datos	354
10.6.	Diseño arquitectónico	356
10.7.	Diseño procedimental	357
10.7.1.	Programación estructurada	358
10.7.2.	Notaciones gráficas de diseño	358
10.7.3.	Notaciones tabulares de diseño	363
10.7.4.	Lenguaje de diseño de programas	366
10.7.5.	Un ejemplo en LDP	372
10.7.6.	Comparación de las notaciones de diseño	374
10.8.	Documentación del diseño	376
10.9.	Resumen	379
	Referencias	379
	Problemas y puntos a considerar	381
	Otras lecturas	382
11.	Diseño orientado al flujo de datos	385
11.1.	Diseño y flujo de la información	385
11.1.1.	Contribuciones	386
11.1.2.	Areas de aplicación	386

11.2.	Consideraciones sobre el proceso de diseño	387
11.2.1.	Flujo de transformación	387
11.2.2.	Flujo de transacción	388
11.2.3.	Un resumen del proceso	389
11.3.	Análisis de transformación	390
11.3.1.	Un ejemplo	390
11.3.2.	Pasos del diseño	390
11.4.	Análisis de transacción	400
11.4.1.	Un ejemplo	401
11.4.2.	Pasos del diseño	402
11.5.	Heurísticas de diseño	405
11.6.	Postprocesamiento en el diseño	408
11.7.	Optimización del diseño	409
11.8.	Resumen	410
	Referencias	410
	Problemas y puntos a considerar	411
	Otras lecturas	413
12.	Diseño orientado a los objetos	415
12.1.	Orígenes del diseño orientado a los objetos	416
12.2.	Conceptos del diseño orientado a los objetos	417
12.2.1.	Objetos, operaciones y mensajes	417
12.2.2.	Aspectos de diseño	418
12.2.3.	Clases, instancias y herencia	420
12.2.4.	Descripciones de los objetos	421
12.3.	Métodos de diseño orientados a los objetos	423
12.4.	Definición de clases y de objetos	424
12.5.	Refinamiento de las operaciones	427
12.6.	Componentes de programa e interfaces	428
12.7.	Una notación para el DOO	431
12.7.1.	Representación de relaciones entre clases y entre objetos	431
12.7.2.	Modularización del diseño	433
12.8.	Diseño detallado para la implementación	434
12.9.	Una estrategia alternativa de diseño orientado a los objetos	437
12.9.1.	Pasos de diseño	437
12.9.2.	Un ejemplo de diseño	438
12.10.	Integración del DOO con el análisis estructurado y el diseño estructurado	443
12.11.	Resumen	445
	Referencias	446
	Problemas y puntos a considerar	447
	Otras lecturas	448
13.	Métodos de diseño orientados a los datos	449
13.1.	Diseño y estructura de los datos	449
13.1.1.	Contribuciones	450
13.1.2.	Áreas de aplicación	451
13.1.3.	Técnicas de estructura de datos frente a las de flujo de datos	451
13.1.4.	Diseño orientado a la estructura de los datos frente al diseño orientado a los objetos	452

13.2.	Consideraciones sobre el proceso de diseño	452
13.3.	Desarrollo de sistemas de Jackson	453
13.3.1.	Pasos de diseño del DSJ	455
13.3.2.	Paso de las funciones	457
13.3.3.	Paso de temporización del sistema	462
13.3.4.	Paso de implementación	463
13.3.5.	Representación procedimental	466
13.4.	Desarrollo de sistemas estructurados en datos	467
13.4.1.	Un método de diseño simplificado	468
13.4.2.	Derivación de la estructura lógica de la salida	469
13.4.3.	Derivación de la estructura lógica del proceso	470
13.4.4.	Lógica de procesos complejos	471
13.5.	Resumen	474
	Referencias	475
	Problemas y puntos a considerar	475
	Otras lecturas	477
14.	Diseño de la interfaz del usuario	479
14.1.	Factores humanos	480
14.1.1.	Fundamentos de la percepción humana	480
14.1.2.	Nivel de habilidad humana y comportamiento	482
14.1.3.	Tareas y factores humanos	482
14.2.	Estilos de interacción entre hombre y máquina	483
14.3.	Diseño de la interfaz hombre-máquina	486
14.3.1.	Modelos de diseño de interfaces	486
14.3.2.	Análisis y modelización de tareas	487
14.3.3.	Aspectos de diseño	490
14.3.4.	Herramientas de implementación	493
14.3.5.	Evaluación del diseño	493
14.4.	Directrices para el diseño de interfaces	495
14.4.1.	Interacción general	496
14.4.2.	Visualización de la información	497
14.4.3.	Entrada de datos	498
14.5.	Estándares de interfaz	499
14.6.	Resumen	499
	Referencias	500
	Problemas y puntos a considerar	500
	Otras lecturas	501
15.	Diseño de tiempo real	503
15.1.	Consideraciones sobre los sistemas	504
15.2.	Sistemas de tiempo real	504
15.2.1.	Aspectos de integración y de rendimiento	505
15.2.2.	Manejo de interrupciones	506
15.2.3.	Bases de datos de tiempo real	508
15.2.4.	Sistemas operativos de tiempo real	509
15.2.5.	Lenguaje de tiempo real	510
15.2.6.	Sincronización y comunicación de tareas	511
15.3.	Análisis y simulación de sistemas de tiempo real	512

15.3.1.	Herramientas matemáticas para el análisis de sistemas de tiempo real	512
15.3.2.	Técnicas de simulación y de modelización para sistemas de tiempo real	517
15.4.	Métodos de diseño	523
15.5.	Un método de diseño orientado al flujo de datos	524
15.5.1.	Requisitos de un método de diseño de sistemas de tiempo real	524
15.5.2.	DARTS	525
15.5.3.	Diseño de tareas	527
15.5.4.	Ejemplo del método de diseño DARTS	528
15.6.	Resumen	533
	Referencias	533
	Problemas y puntos a considerar	534
	Otras lecturas	535
16.	Lenguajes de programación y codificación	537
16.1.	El proceso de traducción	538
16.2.	Características de los lenguajes de programación	538
16.2.1.	Un planteamiento psicológico	539
16.2.2.	Un modelo sintáctico/semántico	542
16.2.3.	Un planteamiento de ingeniería	542
16.2.4.	Elección de un lenguaje	544
16.2.5.	Lenguajes de programación e ingeniería del software	545
16.3.	Fundamentos de los lenguajes de programación	547
16.3.1.	Tipos de datos y tipificación de datos	547
16.3.2.	Subprogramas	549
16.3.3.	Estructuras de control	549
16.3.4.	Soporte para el enfoque orientado a los objetos	550
16.4.	Clases de lenguajes	552
16.4.1.	Primera generación de lenguajes	552
16.4.2.	Segunda generación de lenguajes	553
16.4.3.	Tercera generación de lenguajes	554
16.4.4.	Lenguajes de cuarta generación	557
16.5.	El estilo de codificación	559
16.5.1.	Documentación del código	560
16.5.2.	Declaración de datos	563
16.5.3.	Construcción de sentencias	563
16.5.4.	Entrada/salida	564
16.6.	Eficiencia	565
16.6.1.	Eficiencia en código	566
16.6.2.	Eficiencia en memoria	566
16.6.3.	Eficiencia en la entrada/salida	567
16.7.	Resumen	567
	Referencias	568
	Problemas y puntos a considerar	569
	Otras lecturas	570

CUARTA PARTE. GARANTIA, VERIFICACION Y MANTENIMIENTO DE LA INTEGRIDAD DEL SOFTWARE

17. Garantía de calidad del software	575
17.1. Calidad del software y garantía de calidad del software	576
17.1.1. Factores que determinan la calidad del software	576
17.1.2. Garantía de calidad del software	581
17.1.3. Actividades de SQA	583
17.2. Revisiones del software	584
17.2.1. Impacto de los defectos del software sobre el coste	585
17.2.2. Amplificación y eliminación de defectos	585
17.3. Revisiones técnicas formales	587
17.3.1. La reunión de revisión	588
17.3.2. Registro e informe de la revisión	589
17.3.3. Directrices para la revisión	590
17.3.4. Una lista de comprobaciones para la revisión	593
17.4. Métricas de calidad del software	597
17.4.1. Índices de calidad del software	598
17.4.2. La ciencia del software de Halstead	600
17.4.3. Medida de complejidad de McCabe	602
17.5. Enfoques formales a la SQA	604
17.5.1. Prueba de corrección	604
17.5.2. Garantía de calidad estadística	605
17.5.3. Proceso limpio	608
17.6. Fiabilidad del software	608
17.6.1. Medidas de fiabilidad y de disponibilidad	609
17.6.2. Modelos de fiabilidad del software	610
17.6.3. Seguridad del software	611
17.7. Un enfoque para la garantía de calidad del software	613
17.7.1. Necesidad de SQA	614
17.7.2. Planificación y estándares de SQA	615
17.8. Resumen	615
Referencias	617
Problemas y puntos a considerar	619
Otras lecturas	620
18. Técnicas de prueba del software	623
18.1. Fundamentos de la prueba de software	624
18.1.1. Objetivos de la prueba	624
18.1.2. Flujo de información en la prueba	625
18.2.3. Diseño de casos de prueba	626
18.2. Prueba de la caja blanca	628
18.3. Prueba del camino básico	629
18.3.1. Notación de grafo de flujo	630
18.3.2. Complejidad ciclomática	633
18.3.3. Derivación de casos de prueba	634
18.3.4. Matrices de grafos	639
18.4. Prueba de la estructura de control	640
18.4.1. Prueba de condiciones	641
18.4.2. Prueba de flujo de datos	643

18.4.3.	Prueba de bucles	645
18.5.	Prueba de la caja negra	647
18.5.1.	Partición equivalente	648
18.5.2.	Análisis de valores límite	649
18.5.3.	Técnicas de grafos de causa-efecto	650
18.5.4.	Prueba de comparación	653
18.6.	Prueba de sistemas de tiempo real	654
18.7.	Herramientas automáticas de prueba	656
18.8.	Resumen	657
	Referencias	658
	Problemas y puntos a considerar	659
	Otras lecturas	660
19.	Estrategias de prueba del software	661
19.1.	Un enfoque estratégico para la prueba del software	662
19.1.1.	Verificación y validación	662
19.1.2.	Organización para la prueba del software	663
19.1.3.	Una estrategia de prueba del software	665
19.1.4.	Criterios para la realización de la prueba	666
19.2.	Prueba de unidad	668
19.2.1.	Consideraciones sobre la prueba de unidad	668
19.2.2.	Procedimientos de prueba de unidad	671
19.3.	Prueba de integración	672
19.3.1.	Integración descendente	673
19.3.2.	Integración ascendente	676
19.3.3.	Comentarios sobre la prueba de integración	677
19.3.4.	Documentación de la prueba de integración	678
19.4.	Prueba de validación	680
19.4.1.	Criterios para la prueba de validación	680
19.4.2.	Repaso de la configuración	681
19.4.3.	Pruebas alfa y beta	681
19.5.	Prueba del sistema	682
19.5.1.	Prueba de recuperación	683
19.5.2.	Prueba de seguridad	683
19.5.3.	Prueba de resistencia	684
19.5.4.	Prueba de rendimiento	684
19.6.	El arte de la depuración	685
19.6.1.	El proceso de depuración	686
19.6.2.	Consideraciones psicológicas	687
19.6.3.	Enfoques para la depuración	688
19.7.	Resumen	690
	Referencias	691
	Problemas y puntos a considerar	691
	Otras lecturas	692
20.	Mantenimiento del software	693
20.1.	Una definición de mantenimiento del software	694
20.2.	Característica del mantenimiento	695
20.2.1.	Mantenimiento estructurado frente al no estructurado	695
20.2.2.	Costes del mantenimiento	697

20.2.3.	Problemas	698
20.3.	Facilidad de mantenimiento	699
20.3.1.	Factores de control	699
20.3.2.	Medidas cuantitativas	700
20.3.3.	Revisiones	701
20.4.	Tareas de mantenimiento	702
20.4.1.	Una organización de mantenimiento	702
20.4.2.	Informes	703
20.4.3.	Flujo de sucesos	704
20.4.4.	Registro de información	706
20.4.5.	Evaluación	707
20.5.	Efectos secundarios del mantenimiento	708
20.5.1.	Efectos secundarios sobre el código	708
20.5.2.	Efectos secundarios sobre los datos	709
20.5.3.	Efectos secundarios sobre la documentación	709
20.6.	Mantenimiento de código "ajeno"	710
20.7.	Ingeniería inversa y reingeniería	711
20.7.1.	Mantenimiento preventivo	712
20.7.2.	Elementos de ingeniería inversa	713
20.7.3.	Una técnica de reestructuración para la reingeniería	714
20.8.	Resumen	717
	Referencias	720
	Problemas y puntos a considerar	721
	Otras lecturas	722
21.	Gestión de la configuración del software	723
21.1.	Gestión de las configuraciones del software	724
21.1.1.	Líneas base	724
21.1.2.	Elementos de configuración del software	727
21.2.	El proceso de GCS	729
21.3.	Identificación de objetos en la configuración del software	729
21.4.	Control de versiones	732
21.5.	Control de cambios	734
21.6.	Auditoría de la configuración	737
21.7.	Informes de estado	738
21.8.	Estándares de GCS	738
21.9.	Resumen	738
	Referencias	739
	Problemas y puntos a considerar	740
	Otras lecturas	741

QUINTA PARTE. EL PAPEL DE LA AUTOMATIZACION

22.	Ingeniería del software asistida por computadora	745
22.1.	¿Qué es CASE?	746
22.1.1.	Un taller de ingeniería del software	747
22.1.2.	Una analogía	747
22.2.	Bloques que componen el CASE	748
22.3.	Una clasificación de las herramientas CASE	750
22.4.	Herramientas de planificación de sistemas de gestión	751

22.5.	Herramientas de gestión de proyectos	752
22.5.1.	Herramientas de planificación de proyectos	752
22.5.2.	Herramientas de seguimiento de requisitos	753
22.5.3.	Herramientas de gestión y medida	753
22.6.	Herramientas de soporte	754
22.6.1.	Herramientas de documentación	754
22.6.2.	Herramientas para software de sistemas	755
22.6.3.	Herramientas de control de calidad	755
22.6.4.	Herramientas de bases de datos y de GCS	755
22.7.	Herramientas de análisis y diseño	756
22.7.1.	Herramientas de AE/DE	756
22.7.2.	Herramientas PRO/SIM	756
22.7.3.	Herramientas para el diseño y el desarrollo de interfaces	757
22.7.4.	Máquinas de análisis y diseño	757
22.8.	Herramientas de programación	758
22.8.1.	Herramientas de codificación convencionales	758
22.8.2.	Herramientas de codificación de cuarta generación	758
22.8.3.	Herramientas de programación orientadas a los objetos	759
22.9.	Herramientas de integración y prueba	760
22.9.1.	Herramientas de análisis estático	761
22.9.2.	Herramientas de análisis dinámico	761
22.9.3.	Herramientas de gestión de pruebas	762
22.10.	Herramientas de creación de prototipos	762
22.11.	Herramientas de mantenimiento	764
22.11.1.	Herramientas de ingeniería inversa	764
22.11.2.	Herramientas de reingeniería	765
22.12.	Herramientas de estructura	765
22.13.	CASE e IA	766
22.14.	Resumen	766
	Referencias	766
	Problemas y puntos a considerar	767
	Otras lecturas	768
23.	Entornos CASE integrados	771
23.1.	Definición de requisitos de integración	772
23.2.	Opciones de integración	773
23.3.	Arquitectura de integración	777
23.4.	Integración de las herramientas	778
23.5.	Integración datos-datos y datos-herramienta	780
23.6.	El depósito CASE	782
23.6.1.	El papel del depósito en el I-CASE	782
23.6.2.	Características y contenido	783
23.6.3.	Estándares de depósito	789
23.7.	Resumen	790
	Referencias	791
	Problemas y puntos a considerar	791
	Otras lecturas	792
24.	Perspectivas futuras	793
24.1.	La importancia del software, de nuevo	794

xx **CONTENIDO**

24.2.	El ámbito del cambio	794
24.3.	Las personas y la forma en que construyen sistemas	796
24.4.	El “nuevo” proceso de ingeniería del software	799
24.5.	Nuevas formas de representación de la información	800
24.6.	La tecnología como conductor	802
24.7.	Comentario final	804
	Referencias	805
	Problemas y puntos a considerar	805
	Otras lecturas	806
	Índice	807