
CONTENIDO

Prefacio	xvii
Prefacio a la primera edición	xix
1. Software e ingeniería del software	
1.1. La importancia del software	1
1.1.1. La evolución del software	2
1.1.2. Una perspectiva industrial	5
1.2. El software	5
1.2.1. Características del software	6
1.2.2. Componentes del software	9
1.2.3. Aplicaciones del software	12
1.3. La crisis del software	15
1.3.1. Problemas	15
1.3.2. Causas	16
1.4. Mitos del software	17
1.4.1. Mitos de gestión	17
1.4.2. Mitos del cliente	18
1.4.3. Mitos de los realizadores	19
1.5. Paradigmas de la ingeniería del software	21
1.5.1. Ingeniería del software: una definición	21
1.5.2. El ciclo de vida clásico	23
1.5.3. Construcción de prototipos	25
1.5.4. Técnicas de la cuarta generación	27
1.5.5. Combinación de paradigmas	29
1.6. Una visión genérica de la ingeniería del software	30
1.7. Resumen	33

2.	Ingeniería de sistemas de computadoras	
2.1.	Sistemas basados en computadoras	37
2.2.	Ingeniería de sistemas de computadora	39
2.3.	Consideraciones sobre el hardware	43
2.3.1.	El elemento de hardware	43
2.3.2.	Aplicaciones del hardware	46
2.3.3.	Ingeniería del hardware	47
2.4.	Consideraciones sobre el software	49
2.4.1.	El elemento de software	50
2.4.2.	Aplicaciones del software	50
2.4.3.	Ingeniería del software	52
2.5.	Consideraciones humanas	57
2.5.1.	El elemento humano	57
2.5.2.	Aplicaciones de las IHM	59
2.5.3.	Ingeniería humana	59
2.6.	Consideraciones de la base de datos	60
2.7.	Análisis de sistemas	60
2.7.1.	Identificación de las necesidades	61
2.7.2.	Estudio de viabilidad	62
2.7.3.	Análisis económico	65
2.7.4.	Análisis técnico	69
2.7.5.	Asignación y compromisos	71
2.8.	Una lista de comprobaciones para análisis de sistemas	75
2.9.	La especificación del sistema	84
2.10.	Revisión de la definición del sistema	85
2.11.	Resumen	86
3.	Planificación del proyecto de software	
3.1.	Observaciones sobre estimación	91
3.2.	Objetivos de la planificación del proyecto	92
3.3.	Alcance del software	93
3.4.	Recursos	95
3.4.1.	Recursos humanos	95
3.4.2.	Recursos de hardware	96
3.4.3.	Recursos de software	97
3.5.	Métricas para la productividad y la calidad del software	100
3.5.1.	Medidas del software	100
3.5.2.	Métricas orientadas al tamaño	101
3.5.3.	Métricas orientadas a la función	103
3.5.4.	Reconciliación de las diferentes aproximaciones métricas	106
3.5.5.	Recolección de datos métricos	109
3.6.	Estimación del proyecto de software	111
3.7.	Técnicas de descomposición	114
3.7.1.	Estimación LDC y PF	114
3.7.2.	Un ejemplo	116
3.7.3.	Estimación del esfuerzo	119
3.7.4.	Un ejemplo	119
3.8.	Modelos de estimación empírica	122
3.8.1.	El modelo COCOMO	124
3.8.2.	Modelo de estimación de Putnam	126

3.8.3.	Modelos del punto de función	128
3.8.4.	Un modelo de estudio temporal	128
3.9	Herramientas automáticas de estimación	131
3.10	Planificación temporal del proyecto de software	133
3.10.1.	Relaciones gente-trabajo	133
3.10.2.	Definición de tareas y paralelismo	135
3.10.3.	Distribución de esfuerzos	136
3.10.4.	Métodos de planificación	137
3.10.5.	Un ejemplo de planificación temporal	140
3.11.	Adquisición del software	140
3.12.	Planificación organizativa	143
3.13.	El plan del proyecto de software	147
3.14.	Resumen	148
4.	Fundamentos del análisis de requerimientos	
4.1.	Análisis de requerimientos	153
4.1.1.	Tareas del análisis	154
4.1.2.	El analista	157
4.2.	Areas de problemas	158
4.3.	Principios del análisis	159
4.3.1.	El dominio de la información	159
4.3.2.	Partición	161
4.3.3.	Visiones lógicas y físicas	163
4.4.	Análisis orientado al objeto	164
4.5.	Construcción de prototipos del software	167
4.5.1.	Un escenario de construcción de prototipos	168
4.5.2.	Métodos y herramientas para la construcción de prototipos	170
4.6.	Especificación	171
4.6.1.	Principios de especificación	171
4.6.2.	Representación	175
4.6.3.	Perfil de las especificaciones de los requerimientos del software	177
4.7.	Revisión de la especificación	179
4.8.	Resumen	180
5.	Métodos de análisis de requerimientos	
5.1.	Metodologías de análisis de requerimientos	184
5.1.1.	Características comunes	184
5.1.2.	Métodos y herramientas representativos	185
5.2.	Métodos de análisis orientados al flujo de datos	186
5.2.1.	Diagramas de flujo de datos	186
5.2.2.	Diccionario de datos	193
5.2.3.	Descripciones funcionales	195
5.3.	Métodos orientados a la estructura de datos	196
5.4.	Desarrollo de sistemas estructurados de datos	197
5.4.1.	Diagramas de Warnier	197
5.4.2.	El método DSED	199
5.4.3.	Contexto de la aplicación	199
5.4.4.	Funciones de la aplicación	202
5.4.5.	Resultados de la aplicación	204
5.4.6.	Requerimientos físicos	206

5.5.	Desarrollo de sistemas de Jackson	207
5.5.1.	Paso de las entidades y acciones	208
5.5.2.	Paso de estructuración de entidades	209
5.5.3.	Paso de modelación inicial	210
5.6.	Herramientas para la automatización del análisis de requerimientos	214
5.6.1.	SADT	218
5.6.2.	SREM	220
5.6.3.	PSL/PSA	221
5.6.4.	TAGS	221
5.6.5.	Herramientas de análisis, un resumen	223
5.7.	Requerimientos de las bases de datos	223
5.7.1.	Características de las bases de datos	224
5.7.2.	Pasos del análisis	224
5.7.3.	Normalización	227
5.7.4.	Análisis de la capacidad	229
5.7.5.	Diagramas de datos	231
5.8.	Resumen	232
6.	Fundamentos del diseño de software	
6.1.	La fase de desarrollo y el diseño de software	238
6.2.	El proceso de diseño	239
6.2.1.	Diseño y calidad del software	240
6.2.2.	La evolución del diseño de software	241
6.3.	Fundamentos del diseño	241
6.3.1.	Refinamiento	242
6.3.2.	Arquitectura del software	242
6.3.3.	Estructura del programa	244
6.3.4.	Estructura de datos	245
6.3.5.	Procedimientos de software	247
6.3.6.	Modularidad	247
6.3.7.	Abstracción	251
6.3.8.	Ocultación de información	254
6.4.	Diseño modular efectivo	254
6.4.1.	Tipos de módulos	254
6.4.2.	Independencia funcional	256
6.4.3.	Cohesión	256
6.4.4.	Acoplamiento	258
6.5.	Diseño de datos	262
6.6.	Diseño arquitectónico	264
6.7.	Diseño procedimental	265
6.7.1.	Programación estructurada	265
6.7.2.	Herramientas gráficas de diseño	266
6.7.3.	Herramientas tabulares de diseño	272
6.7.4.	Lenguaje de diseño de programas	274
6.7.5.	Un ejemplo LPD	280
6.7.6.	Comparación de notaciones de diseño	281
6.8.	Documentación del diseño	283
6.9.	Resumen	286

7.	Diseño orientado al flujo de datos,	
7.1.	Diseño y flujo de la información	291
7.1.1.	Contribuciones	292
7.1.2.	Áreas de aplicación	292
7.2.	Consideraciones sobre el proceso de diseño	293
7.2.1.	Flujo de transformación	293
7.2.2.	Flujo de transacción	294
7.2.3.	Un resumen del proceso	295
7.3.	Análisis de transformación	296
7.3.1.	Un ejemplo	296
7.3.2.	Pasos del diseño	299
7.4.	Análisis de transacción	307
7.4.1.	Un ejemplo	307
7.4.2.	Pasos del diseño	309
7.5.	Heurísticas de diseño	315
7.6.	Postprocesamiento en el diseño	318
7.7.	Optimización del diseño	320
7.8.	Resumen	321
8.	Diseño orientado a la estructura de datos,	
8.1.	Diseño y estructura de los datos	326
8.1.1.	Contribuciones	327
8.1.2.	Áreas de aplicación	328
8.1.3.	Técnicas de estructura de datos frente a las de flujo de datos	328
8.2.	Consideraciones sobre el proceso de diseño	329
8.3.	Desarrollo de sistema de Jackson	329
8.3.1.	Pasos del diseño DSJ	331
8.3.2.	Paso funcional	334
8.3.3.	Paso de temporización del sistema	339
8.3.4.	Paso de implementación	340
8.3.5.	Representación procedimental	342
8.4.	Construcción lógica de programas y sistemas	344
8.4.1.	El diagrama de Warnier	344
8.4.2.	El método de diseño CLP	345
8.4.3.	Organización detallada	350
8.4.4.	Estructuras complejas	352
8.5.	Desarrollo de sistemas estructurados de datos	356
8.5.1.	Un método de diseño simplificado	357
8.5.2.	Derivación de la estructura lógica de la salida	359
8.5.3.	Derivación de la estructura lógica del proceso	360
8.5.4.	Lógica de procesos complejos	362
8.6.	Resumen	365
9.	Diseño orientado al objeto	
9.1.	Orígenes del diseño orientado al objeto	371
9.2.	Conceptos del diseño orientado al objeto	372
9.2.1.	Objetos, operaciones y mensajes	372
9.2.2.	Clases, instancias y herencia	374
9.2.3.	Descripciones de objetos	375

9.3.	Métodos de diseño orientado al objeto	377
9.4.	Definición del problema	378
9.5.	Una estrategia informal	380
9.6.	Formalización de la estrategia	380
9.6.1.	Objetos y sus atributos	381
9.6.2.	Operaciones aplicadas a los objetos	383
9.6.3.	Componentes e interfaces del programa	386
9.6.4.	Representaciones gráficas para el DOO	388
9.6.5.	Detalle de implementación	392
9.7.	Un método alternativo	395
9.7.1.	Pasos del diseño	395
9.7.2.	Un ejemplo de diseño	397
9.8.	Resumen	402
10.	Diseño de tiempo real	
10.1.	Consideraciones sobre los sistemas	407
10.2.	Sistemas de tiempo real	407
10.2.1.	Aspectos de integración y rendimiento	408
10.2.2.	Manejo de interrupciones	409
10.2.3.	Bases de datos de tiempo real	411
10.2.4.	Sistemas operativos de tiempo real	412
10.2.5.	Lenguajes de tiempo real	413
10.2.6.	Sincronización y comunicación de tareas	414
10.3.	Análisis de sistemas de tiempo real	415
10.3.1.	Herramientas matemáticas para el análisis de sistemas de tiempo real	416
10.3.2.	Métodos formales de análisis para sistemas de tiempo real	422
10.4.	Métodos de diseño del software	423
10.5.	Un método de diseño orientado al flujo de datos	423
10.5.1.	Requerimientos de un método de diseño de sistemas de tiempo real	424
10.5.2.	DARTS	425
10.5.3.	Diseño de tareas	427
10.5.4.	Ejemplo del método de diseño DARTS	428
10.6.	Otros métodos de diseño de tiempo real	434
10.6.1.	Extensiones del modelo de flujo de datos	435
10.6.2.	Métodos alternativos	441
10.7.	Resumen	441
11.	Lenguajes de programación y codificación	
11.1.	El proceso de traducción	447
11.2.	Características de los lenguajes de programación	447
11.2.1.	Una visión psicológica	448
11.2.2.	Un modelo sintáctico/semántico	451
11.2.3.	Una visión de ingeniería	451
11.2.4.	Elección de un lenguaje	453
11.2.5.	Lenguajes de programación e ingeniería del software	454
11.3.	Fundamentos de los lenguajes de programación	456
11.3.1.	Tipos de datos y tipificación de datos	456
11.3.2.	Subprogramas	458

11.3.3.	Estructuras de control	458
11.4.	Clases de lenguajes	459
11.4.1.	Primera generación de lenguajes	459
11.4.2.	Segunda generación de lenguajes	460
11.4.3.	Tercera generación de lenguajes	461
11.4.4.	Lenguajes de cuarta generación	464
11.5.	El estilo de codificación	466
11.5.1.	Documentación del código	466
11.5.2.	Declaración de datos	468
11.5.3.	Construcción de sentencias	470
11.5.4.	Entrada/Salida	471
11.6.	Eficiencia	473
11.6.1.	Eficiencia en código	473
11.6.2.	Eficiencia en memoria	474
11.6.3.	Eficiencia en la Entrada/Salida	474
11.7.	Resumen	475
12.	Garantía de calidad del software	
12.1.	Calidad del software y garantía de calidad del software	480
12.1.1.	Factores que determinan la calidad del software	481
12.1.2.	Garantía de calidad del software	483
12.1.3.	Actividades de SQA	484
12.2.	Revisiones del software	487
12.2.1.	Impacto sobre el coste de los defectos del software	488
12.2.2.	Amplificación y eliminación de defectos	488
12.3.	Revisiones técnicas formales	491
12.3.1.	La reunión de revisión	491
12.3.2.	Registro e informes de revisión	492
12.3.3.	Directrices de revisión	494
12.3.4.	Una lista de comprobaciones para la revisión	496
12.4.	Métricas de calidad del software	501
12.4.1.	La ciencia del software de Halstead	502
12.4.2.	Medida de complejidad de McCabe	507
12.5.	Fiabilidad del software	508
12.5.1.	Medidas de la fiabilidad y de la disponibilidad	509
12.5.2.	Modelos de fiabilidad del software	510
12.6.	Una aproximación a la garantía de calidad del software	511
12.6.1.	Examinando la necesidad de la SQA	512
12.6.2.	Planificación y estándares de SQA	513
12.7.	Resumen	514
13.	Técnicas de prueba del software	
13.1.	Fundamentos de la prueba del software	519
13.1.1.	Objetivos de la prueba	519
13.1.2.	Flujo de información de la prueba	520
13.1.3.	Diseño de casos de prueba	521
13.2.	Prueba de la caja blanca	523
13.3.	Prueba del camino básico	524
13.3.1.	Notación de grafo de flujo	524
13.3.2.	Complejidad ciclomática	527

13.3.3.	Derivación de casos de prueba	529
13.3.4.	Matrices de grafos	533
13.4.	Prueba de bucles	535
13.5.	Prueba de la caja negra	537
13.5.1.	Partición equivalente	538
13.5.2.	Análisis de valores límite	539
13.5.3.	Técnicas de grafos de causa-efecto	540
13.5.4.	Prueba de validación de datos	542
13.6.	Prueba de corrección	544
13.7.	Herramientas automáticas de prueba	545
13.8.	Resumen	549
14.	Estrategias de prueba de software	
14.1.	Un enfoque estratégico de la prueba de software	554
14.1.1.	Verificación y validación	554
14.1.2.	Organización de la prueba del software	556
14.1.3.	Una estrategia de prueba del software	557
14.2.	Prueba de unidad	559
14.2.1.	Consideraciones sobre la prueba de unidad	559
14.2.2.	Procedimientos de prueba de unidad	562
14.3.	Prueba de integración	563
14.3.1.	Integración descendente	564
14.3.2.	Integración ascendente	567
14.3.3.	Comentarios sobre la prueba de integración	567
14.3.4.	Documentación de la prueba de integración	569
14.4.	Prueba de validación	571
14.4.1.	Criterios para la prueba de validación	572
14.4.2.	Repaso de la configuración	572
14.4.3.	Pruebas alfa y beta	572
14.5.	Prueba del sistema	573
14.5.1.	Prueba de recuperación	574
14.5.2.	Prueba de seguridad	574
14.5.3.	Prueba de resistencia	575
14.5.4.	Prueba de rendimiento	576
14.6.	El arte de la depuración	577
14.6.1.	El proceso de depuración	577
14.6.2.	Consideraciones psicológicas	579
14.6.3.	Enfoques de la depuración	579
14.7.	Resumen	581
15.	Mantenimiento de software y gestión de configuraciones	
15.1.	Una definición de mantenimiento de software	585
15.2.	Características del mantenimiento	587
15.2.1.	Mantenimiento estructurado frente al no estructurado	587
15.2.2.	Costes de mantenimiento	588
15.2.3.	Problemas	590
15.3.	Facilidad de mantenimiento	591
15.3.1.	Factores de control	591

15.3.2.	Medidas cuantitativas	592
15.3.3.	Revisiones	593
15.4.	Tareas de mantenimiento	593
15.4.1.	Una organización de mantenimiento	593
15.4.2.	Informes	595
15.4.3.	Flujo de sucesos	595
15.4.4.	Registro de información	597
15.4.5.	Evaluación	599
15.5.	Efectos secundarios del mantenimiento	600
15.5.1.	Efectos secundarios sobre el código	600
15.5.2.	Efectos secundarios sobre los datos	601
15.5.3.	Efectos secundarios sobre la documentación	601
15.6.	Aspectos del mantenimiento	602
15.6.1.	Mantenimiento de "código ajeno"	602
15.6.2.	Mantenimiento preventivo	603
15.6.3.	Una estrategia de "repuestos"	604
15.7.	Gestión de configuraciones del software	605
15.7.1.	Líneas base	605
15.7.2.	Elementos de configuración del software	606
15.8.	El proceso de GCS	607
15.8.1.	Identificación	608
15.8.2.	Control de cambios	609
15.8.3.	Auditorías de configuración	611
15.8.4.	Informes de estado	611
15.8.5.	Estándares de GCS	613
15.9.	Resumen	613
	Epilogo	617
	Indice	619