

CONTENIDO BREVE

INTRODUCCIÓN 1

CAPÍTULO 1 *PROCESO* 16

CAPÍTULO 2 *ADMINISTRACIÓN DE PROYECTOS* 73

CAPÍTULO 3 *ANÁLISIS DE REQUERIMIENTOS I* 135

CAPÍTULO 4 *ANÁLISIS DE REQUERIMIENTOS II* 181

CAPÍTULO 5 *ARQUITECTURA DE SOFTWARE* 247

CAPÍTULO 6 *DISEÑO DETALLADO* 302

CAPÍTULO 7 *IMPLEMENTACIÓN DE UNIDADES* 355

CAPÍTULO 8 *PRUEBAS DE UNIDADES* 393

CAPÍTULO 9 *INTEGRACIÓN, VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA* 432

CAPÍTULO 10 *MANTENIMIENTO* 479

ACRÓNIMOS 511

GLOSARIO 515

REFERENCIAS 519

xii CONTENIDO BREVE

CRÉDITOS 523

APÉNDICE 525

ÍNDICE 529

CONTENIDO

INTRODUCCIÓN 1

- 1. Contexto de la ingeniería de software 1
- 2. Actividades de la ingeniería de software 2
- 3. Proceso 3
- 4. Proyecto 5
- 5. Personas 5
- 6. Producto 6
- 7. Calidad 7
- 8. Proyecto de equipo 8
 - 8.1 Comunicación del grupo 9
- 9. Caso de estudio, panorama 10
 - 9.1 Introducción al videojuego *Encuentro* 11
 - 9.2 Requerimientos para *Encuentro* 12
 - 9.3 Diseño de *Encuentro* 12
 - 9.4 Pruebas de *Encuentro* 13
 - 9.5 Documentación del proyecto *Encuentro* 13
- Ejercicios 14

CAPÍTULO 1 PROCESO 16

PARTE I: ESENCIAL 17

- 1. Introducción al proceso de ingeniería de software 17
 - 1.1 Mapa conceptual típico de ingeniería de software 18
- 2. Perspectiva histórica y contemporánea de la ingeniería de software 19
 - 2.1 La ingeniería de software crece 19
 - 2.2 Influencia de la programación estructurada y la orientación a objetos 20
 - 2.3 Reuso y componentes 21
 - 2.4 Métodos formales 22
 - 2.5 Posibilidades de uso 23
- 3. Expectativas para proceso, proyecto, producto y personas 23
 - 3.1 Artefactos y roles 24
- 4. Procesos alternativos 24
 - 4.1 Modelo de proceso en cascada 24
 - 4.2 Modelo de proceso en espiral 26
 - 4.3 Modelo de proceso por incrementos 28
 - 4.4 Proceso de desarrollo de software unificado (USDP) 29
 - 4.5 Trueques del proceso 31
- 5. Documentación 33
 - 5.1 Introducción a la documentación 33
 - 5.2 Estándares de documentación 34

PARTE II: EXTENSIÓN 38

- 6. Calidad 38
 - 6.1 Métricas 39

6.2	Proceso de aseguramiento de la calidad	40
6.3	Técnicas de caja negra y caja blanca	40
6.4	Introducción a la inspección	41
6.5	Estándar de IEEE para planes de aseguramiento de la calidad del software	46
6.6	Verificación y validación	46
6.6.1	Estándar de V&V de IEEE 1012-1986	48
7.	Administración de la documentación	49
7.1	Introducción a la administración de la documentación	49
7.2	Consistencia y documentación con una sola fuente	49
7.3	Administración de la configuración	51
7.3.1	Elementos de configuración	52
7.3.2	Estándar IEEE 829-1990 para los planes de administración de la configuración del software	53
8.	Introducción a la evaluación de capacidades	54
8.1	Introducción a Personal Software Process (PSP)	55
8.2	Introducción al proceso de software de equipo (TSP)	56
8.3	Introducción al modelo de madurez de capacidades (CMM)	57
8.4	Relación entre PSP, TSP y CMM	60
9.	Resumen	60
	Ejercicios	61
	Caso de estudio 1: plan de administración de la configuración del software	63
	Caso de estudio 2: plan de aseguramiento de la calidad del software, parte 1 de 2	68

CAPÍTULO 2 ADMINISTRACIÓN DE PROYECTOS 73

PARTE I: ESENCIAL 74

1.	Introducción a la administración de proyectos	74
1.1	Significado de “administración de proyectos”	74
1.2	Las componentes de la administración de proyectos	75
1.3	Variables principales: <i>costo, capacidad, calidad y programa</i>	75
1.4	Mapa conceptual típico de un proceso de administración de un proyecto	76
2.	Administración del personal del proyecto	77
2.1	Profesionalismo	77
2.2	Importancia de administrar a las personas	78
2.3	Perspectivas de la empresa	78
2.4	Perspectivas de la administración	79
2.4.1	Realización de reuniones	79
2.5	Perspectivas de los ingenieros	80
3.	Opciones para la organización del personal	81
3.1	Administración de la comunicación	81
3.2	Opciones para la estructura de responsabilidades	82
3.3	Fuentes de personal para un proyecto	84
4.	Identificación y retiro del riesgo	85
4.1	Definición de “riesgos”	85
4.2	Panorama de la administración del riesgo	86
4.3	Identificación de riesgos	86
4.4	Retiro de riesgo	87
5.	Elección de herramientas de desarrollo y soporte	90
5.1	Métodos del proceso	90
5.2	Herramientas	90

5.3	Decisiones de construir o comprar	90
5.4	Selección del lenguaje	92
5.5	Documentación	92
5.6	Servicios de apoyo	92
6.	Creación de la calendarización: planeación de alto nivel	92
PARTE II:	EXTENSIÓN	94
7.	Integración de aplicaciones heredadas	94
8.	Estimación de costos: cálculos preliminares	96
8.1	Introducción	96
8.2	Estimación de líneas de código sin el proceso de puntos de la función	98
8.3	Puntos de función y líneas de código	100
8.4	Conversión de puntos de función en líneas de código	102
8.5	Otro ejemplo de puntos de función	103
8.6	Referencias de puntos de función	104
9.	Estimación del esfuerzo y la duración a partir de las líneas de código	105
10.	Proceso de software en equipo	107
11.	Plan de administración del proyecto de software	108
12.	Calidad en la administración del proyecto	110
12.1	Métricas del proceso	110
12.1.1	Introducción a la métrica del proceso	110
12.1.2	Ejemplos de métricas del proceso	110
12.2	IEEE 739-1989 PAPS: Parte 2	112
13.	Mejoras al proceso y el modelo de madurez de la capacidad	114
13.1	Mejora al proceso usado en una organización	114
13.2	Mejora del proceso para un proyecto en marcha	114
14.	Varias herramientas y técnicas para la administración de proyectos	116
14.1	Equipos remotos e internacionales	116
14.2	Programación extrema	116
14.3	Toma de decisiones con prioridades urgentes	117
15.	Resumen del proceso de administración de proyectos	118
	Guía para el proyecto: Plan de administración de proyectos para el caso de estudio <i>Encuentro</i>	119
	GPP1. Preparación para la reunión de planeación del proyecto	119
	GPP2. Reunión inicial de planeación del proyecto	120
	GPP3. Terminación del plan de administración del proyecto	120
	Ejercicios	121
	Caso de estudio 1: plan de administración del proyecto para el videojuego <i>Encuentro</i>	123
	Caso de estudio 2: plan de aseguramiento de la calidad del software, parte 2 de 2	131
CAPÍTULO 3	ANÁLISIS DE REQUERIMIENTOS I	135
1.	Introducción al análisis de requerimientos	136
1.1	Significado de análisis de requerimientos	136
1.2	Requerimiento C y requerimientos D	137
1.3	Por qué deben escribirse los requerimientos	137
1.4	Mapa conceptual típico del proceso de análisis de requerimientos	139
1.5	Retos y beneficios del análisis de requerimientos	140

2.	Interacción con el cliente	141
2.1	Fuentes de requerimientos	141
2.2	Identificación de interesados	142
2.3	Ejemplos de deseos de los clientes	143
2.4	Proceso de entrevista y documentación	144
3.	Descripción de los requerimientos C (o del cliente)	145
3.1	Concepto de operaciones	145
3.2	Casos de uso	145
3.3	Diagramas de flujo de datos para la comunicación con el cliente	148
3.4	Diagramas de transición de estados para comunicación con el cliente	149
3.5	Diseño preliminar de interfaces de usuario y otras	151
3.5.1	Pasos para desarrollar interfaces de usuario	151
3.5.2	Ejemplos de requerimientos preliminares para el caso de estudio	158
3.5.3	Otras interfaces	158
3.6	Resumen y guía par expresar los requerimientos C	159
4.	Uso de metodologías, herramientas e internet para los requerimientos C	160
5.	Prototipos rápidos, estudios de factibilidad y pruebas de concepto	161
5.1	Prototipos rápidos	161
5.2	Estudios de factibilidad	164
6.	Actualización del proyecto para reflejar el análisis de requerimientos C	165
6.1	Requerimientos C y escalas del proyecto	165
6.2	Efectos del análisis de requerimientos C en el plan del proyecto	165
7.	Tendencias futuras y resumen de requerimientos C	166
7.1	Tendencias futuras	166
7.2	Resumen	166
	Guía para el proyecto: requerimientos C para el caso de estudio <i>Encuentro</i>	168
	GPP1. Preparación	168
	GPP2. Entrevista con el cliente	169
	GPP3. Descripción de la especificación de requerimientos de software	170
	GPP4. Seguimiento	171
	GPP5. Métricas y posmortem	171
	Ejercicios	171
	Caso de estudio: especificación de requerimientos de software (ERS) para el juego de video <i>Encuentro</i> , parte 1 de 2	174

CAPÍTULO 4 ANÁLISIS DE REQUERIMIENTOS II 181

PARTE I: ESENCIAL 182

1.	Introducción a los requerimientos específicos (D)	182
1.1	Significado de los requerimientos específicos (D)	182
1.2	Mapa conceptual típico de análisis de requerimientos D	183
2.	Tipos de requerimientos D	184
2.1	Requerimientos funcionales	184
2.2	Requerimientos no funcionales: requerimientos de desempeño	185
2.3	Requerimientos no funcionales: confiabilidad y disponibilidad	185
2.4	Requerimientos no funcionales: manejo de errores	185
2.5	Requerimientos no funcionales: requerimientos de interfaz	186
2.6	Requerimientos no funcionales: restricciones	186

2.7	Requerimientos inversos	187
2.8	Correspondencia de tipos de requerimientos con el ERS estándar IEEE 830-1993	187
3.	Propiedades deseadas de los requerimientos D	187
3.1	Rastreabilidad	187
3.1.1	Rastreo de requerimientos funcionales	187
3.1.2	Rastreo de requerimientos no funcionales	190
3.2	Comprobación y no ambigüedad	191
3.3	Prioridad	191
3.4	Completo	194
3.5	Condiciones de error	194
3.6	Consistencia	195
3.7	Resumen del proceso de escribir un requerimiento detallado	196
4.	Diagramas de secuencia	196
5.	Organización de los requerimientos D	200
5.1	Por qué es importante la organización de los requerimientos detallados	200
5.2	Métodos para organizar los requerimientos específicos	201
5.3	Organización de los requerimientos detallados por caso de uso	203
5.4	Organización de requerimientos por clase	203
5.5	Identificación de clases	205
5.6	Selección de la clase correcta para un requerimiento dado	210
5.7	Clasificación de <i>entidades</i> (instancias)	211
5.8	Liga con la documentación de pruebas	211
PARTE II: EXTENSIÓN 212		
6.	Calidad de los requerimientos específicos	212
6.1	Intervención de QA en el análisis de requerimientos D	212
6.2	Métricas para el análisis de requerimientos D	213
6.3	Inspección del análisis de requerimientos D	214
6.3.1	Ejemplo de requerimientos D no inspeccionados	214
6.3.2	Ejemplo de resultados de inspección de los requerimientos D	215
7.	Uso de herramientas e Internet para el análisis de requerimientos	217
8.	Métodos formales para especificación de requerimientos	219
8.1	Introducción a las especificaciones formales	219
8.1.1	Notación matemática seleccionada	219
8.2	Ejemplos de especificaciones formales	220
8.3	¿Cuándo debe usarse la especificación formal?	224
8.4	Precondiciones y poscondiciones	225
9.	Efectos del proceso de requerimientos D sobre los proyectos	226
9.1	Efectos sobre el PAPS	225
9.2	Efectos de la gran escala sobre los requerimientos D	225
10.	Resumen del proceso de requerimientos D	226
	Guía para el proyecto: requerimientos D para el caso de estudio <i>Encuentro</i>	227
	GPP1. Preparación	227
	GPP2. Clasificación de los requerimientos D	228
	GPP3. Escritura de los requerimientos D	228
	GPP4. Seguimiento: métricas y posmortem	229
	Ejercicios	229
	Caso de estudio: especificación de requerimientos de software (ERS) para el videojuego <i>Encuentro</i> , parte 2 de 2	231

CAPÍTULO 5 ARQUITECTURA DE SOFTWARE 247

PARTE I: ESENCIAL 248

1. Introducción a ingeniería de sistemas y arquitectura de software **248**
 - 1.1 Panorama global: ingeniería de sistemas **248**
 - 1.2 Significado de “arquitectura de software” **250**
 - 1.3 Metas de la selección de la arquitectura **250**
 - 1.4 Descomposición **251**

PARTE II: EXTENSIÓN 253

2. Modelos, marcos de trabajo y patrones de diseño **254**
 - 2.1 Uso de “modelos” **254**
 - 2.2 Lenguaje de modelado unificado **256**
 - 2.3 Marcos de trabajo **257**
 - 2.4 Una clasificación de las arquitecturas **260**
 - 2.5 Patrones de diseño I: introducción **260**
 - 2.6 Componentes **262**
 3. Arquitecturas alternativas de software y sus modelos de clase **263**
 - 3.1 Arquitecturas de flujo de datos **263**
 - 3.2 Componentes independientes **266**
 - 3.2.1 Arquitecturas cliente-servidor y patrones de diseño *Apariencia* **266**
 - 3.2.2 Arquitectura de *procesadores de comunicación paralelos* **268**
 - 3.2.2.1 Patrón de diseño del *observador* **269**
 - 3.2.3 Arquitecturas de *sistemas por eventos* y patrón de diseño *Estado* **270**
 - 3.3 Máquinas virtuales **272**
 - 3.4 Arquitecturas de almacenamiento **276**
 - 3.4.1 “Visita” a los miembros de un almacenamiento con el patrón de diseño *Iterador* **277**
 - 3.5 Arquitecturas de capas **278**
 - 3.6 Uso de arquitecturas múltiples dentro de una aplicación **281**
 - 3.7 Resumen y procedimientos para seleccionar arquitecturas **281**
 4. Notación de la arquitectura, estándares y herramientas **281**
 - 4.1 Notación **281**
 - 4.2 Herramientas **281**
 - 4.2.1 Herramientas a nivel de arquitectura contra herramientas para diseño detallado e implementación **281**
 - 4.3 Estándares IEEE/ANSI para expresar diseños **282**
 5. Aseguramiento de la calidad (QA) de la arquitectura elegida **283**
 - 5.1 Calidad en la selección de la arquitectura **283**
 - 5.1.1 Métricas para la selección de la arquitectura **283**
 - 5.2 Elección de una arquitectura entre las alternativas **285**
 - 5.3 Verificación de arquitecturas con casos de uso **287**
 - 5.4 Inspección de la selección de arquitectura **288**
 - 5.5 Efectos de la selección de arquitectura sobre el PAPS **288**
 6. Resumen **289**
- Guía para el proyecto: arquitectura del caso de estudio *Encuentro* **290**
- GPP1. Preparación **290**
 - GPP2. Selección de arquitecturas **290**
 - GPP3. Junta de equipo (“revisión del diseño preliminar”) **292**

GPP4. Refinación de la arquitectura	292
GPP5. Documentación de la arquitectura	292
Ejercicios	293
Caso de estudio: documento de diseño de software, parte 1 de 2	294
I. Marco de trabajo de la arquitectura del juego de personajes	294
II. Arquitectura del juego de personajes <i>Encuentro</i> parte 1 de 2 del documento de diseño de software	296

CAPÍTULO 6 DISEÑO DETALLADO 302

PARTE I: ESENCIAL 303

1. Introducción al diseño detallado	303
1.1 Significado de “diseño detallado”	303
1.2 Relación entre casos de uso, arquitectura y diseño detallado	303
1.3 Mapa conceptual típico del proceso de “diseño detallado”	304
1.4 Diseño en el proceso de desarrollo unificado	305
1.5 Diseño contra interfaces	308
1.6 Reuso de componentes	308
2. Diagramas de secuencia y flujo de datos para el diseño detallado	309
2.1 Diagramas de secuencia detallados	309
2.2 Diagramas de flujo de datos detallados	311
3. Especificación de clases y funciones	312
3.1 Invariantes de clase	314
3.2 Invariantes, precondiciones y poscondiciones de funciones	314
4. Especificación de algoritmos	315
4.1 Diagramas de flujo	315
4.1.1 Un ejemplo de diagrama de flujo	315
4.2 Seudocódigo	317
4.3 Cuándo usar diagramas de flujo y pseudocódigo	318

PARTE II: EXTENSIÓN 319

5. Patrones de diseño II: técnicas del diseño detallado	319
5.1 Patrones creativos para el diseño detallado	320
5.1.1 Solitario	320
5.1.2 Fábrica	320
5.1.3 Prototipo	321
5.1.4 Fábrica y constructor abstractos	322
5.2 Patrones estructurales para diseño detallado	325
5.2.1 Compuesto y Decorador	325
5.2.2 Apariencia	326
5.2.3 Adaptador	326
5.2.4 Peso mosca	328
5.2.5 Proxy	328
5.3 Patrones de comportamiento para diseño detallado	329
5.3.1 Intérprete, Observador y Estado	329
5.3.2 Iterador	329
5.3.3 Mediador	330
6. Librería de plantillas estándar	331
7. Estándares, notación y herramientas para el diseño detallado	332
7.1 Estándar IEEE 890 para el diseño detallado	332

XX CONTENIDO

7.2	Lenguaje de modelado unificado (UML) para modelos de objetos detallados	333
7.3	Herramientas basadas en el código fuente: Javadoc	333
8.	Efectos del diseño detallado en los proyectos	335
8.1	Estimación de tamaño a partir de diseños detallados	336
9.	Calidad en los diseños detallados	338
9.1	Cualidades y métricas para el diseño detallado	340
9.2	Inspección del diseño detallado	340
9.2.1	Clasificación de defectos	340
9.2.2	Ejemplos de inspección de requerimientos detallados	341
10.	Resumen	342
	Ejercicios	343
	Caso de estudio	344
I.	Diseño detallado para el marco de trabajo del juego de personajes (partes restantes del documento de diseño de software)	344
II.	Diseño detallado de Encuentro (partes restantes del documento de diseño de software)	346

CAPÍTULO 7 *IMPLEMENTACIÓN DE UNIDADES* **355**

1.	Introducción a la implementación	356
1.1	Definición de “implementación de unidades”	356
1.2	Metas de la implementación	356
1.3	Mapa conceptual típico del proceso de implementación de unidades	356
1.4	Implementación en el proceso de unificado de desarrollo de software	358
1.5	Lenguajes de programación	360
2.	Programación y estilo	361
2.1	Principios generales de una implementación acertada	362
2.2	Indicadores y referencias	363
2.3	Funciones	363
2.4	Excepciones	363
2.5	Manejo de errores	364
2.6	Otros puntos de la práctica	367
3.	Estándares de programación	367
3.1	Convenciones de nombres, ejemplos de Java	368
3.2	Documentación de atributos	369
3.3	Constantes	369
3.4	Inicialización de atributos	370
4.	Programas con demostración formal de que son correctos	370
5.	Herramientas y entornos para programación	375
6.	La calidad en la implementación	376
6.1	Métricas estándar para el código fuente	378
6.1.1	Cuenta de líneas	378
6.1.2	Métricas de IEEE	378
6.2	Métricas especiales para el código fuente	379
6.3	Inspección de código	379
6.4	Documentación personal de software	381

7. Resumen del proceso de implementación	382
Ejercicios	382
Caso de estudio	384
I. Actualización de PAQS	384
II. Actualización del apéndice PACS: modelo de implementación	385
III. Documentación personal del software, parte 1 de 2	385
IV. Código fuente (sin código de prueba): <i>PersonajesEncuentro</i>	386

CAPÍTULO 8 *PRUEBAS DE UNIDADES* **393**

1. Introducción a las pruebas de unidades	394
1.1 Metas de las pruebas	394
1.2 Significado de “pruebas de unidades”	395
1.3 Mapa conceptual típico de las pruebas de unidades	396
2. Tipos de pruebas	397
2.1 Pruebas con caja negra, caja blanca y caja gris	397
2.2 Particiones equivalentes para pruebas de caja negra	398
2.3 Análisis del valor de la frontera para pruebas de caja negra	399
2.4 Cobertura de la declaración para pruebas de caja blanca	400
2.5 Cobertura de decisiones para pruebas de caja blanca	400
2.6 Pruebas basadas en afirmaciones	402
2.6.1 Límites de la verificación de afirmaciones automática	403
2.7 Aleatoriedad en las pruebas	404
3. Planeación de pruebas de unidades	405
4. Listas de verificación y ejemplos para las pruebas de métodos	407
4.1 Pruebas de unidades de métodos	407
4.2 Ejemplo de prueba de método	408
4.2.1 Prueba de los métodos completos de una clase	412
5. Listas de verificación y ejemplos para pruebas de clase	415
5.1 Ejemplo de prueba de combinación de métodos	415
5.2 Pruebas orientadas a atributos	416
5.3 Pruebas de invariantes de clase	416
5.4 Pruebas basadas en los estados	417
6. Resumen	418
Ejercicios	419
Caso de estudio: <i>PersonajesEncuentro</i> , documentación personal de software (DPS) en Java, parte 2 de 2	421

CAPÍTULO 9 *INTEGRACIÓN, VERIFICACIÓN Y VALIDACIÓN DEL SISTEMA* **432**

1. Introducción	433
1.1 Significado de “integración”	433
1.2 Verificación, validación y pruebas del sistema	434
2. Proceso de integración	437
2.1 Descripción de “integración”	437
2.2 Mapa conceptual típico del proceso de integración y pruebas del sistema	441
3. Proceso de pruebas	443
3.1 Pruebas de integración	443
3.2 Pruebas de desarrolladores y artefactos	447

xxii CONTENIDO

3.3	Prueba de interfaces	448
3.4	Prueba del sistema	449
3.5	Prueba de utilidad	451
3.5.1	Prueba para los requerimientos de interfaz de usuario	451
3.5.2	Medidas de utilidad	451
3.6	Prueba de regresión	452
3.7	Pruebas de aceptación	452
3.8	Pruebas de instalación	452
4.	Documentación de integración y pruebas	453
4.1	Estándares para la documentación de pruebas	453
4.2	Organización de la documentación de integración y pruebas	454
5.	Iteraciones de transición	455
5.1	Versiones alfa y beta	456
5.2	Mapa conceptual de las iteraciones de transición	457
6.	Calidad en integración, verificación y validación	458
6.1	Metas de calidad	458
6.2	Métricas para la integración y pruebas del sistema	458
6.3	Inspección de integración y pruebas del sistema	459
6.4	Participación de aseguramiento de la calidad (QA) en la integración y pruebas del sistema	460
6.5	Integración del sistema y el modelo de madurez de capacidades	460
7.	Herramientas para integración y pruebas del sistema	461
8.	Resumen	464
	Ejercicios	464
	Caso de estudio	466
	I. PACS: apéndice A. Plan para las bases de integración	466
	II. Documentación de pruebas de software para <i>Encuentro</i>	468

CAPÍTULO 10 *MANTENIMIENTO* 479

1.	Introducción	480
1.1	Significado de “mantenimiento de software”	481
1.2	Aspectos de mantenimiento de software	481
1.3	Mapa conceptual típico para establecer el proceso de mantenimiento	483
2.	Tipos de mantenimiento de software	484
3.	Técnicas de mantenimiento	485
3.1	Análisis del impacto	485
3.2	Ingeniería inversa	486
3.3	Reingeniería	487
3.3.1	Refactorización	487
3.4	Aplicaciones heredadas	488
3.5	Actualización de documentación	489
4.	Estándar 1219-1992 de IEEE	490
4.1	Identificación del problema de mantenimiento	491
4.2	Análisis del problema de mantenimiento	491
4.2.1	Ejemplos de análisis del problema de mantenimiento	491
4.3	Diseño para una solicitud de mantenimiento	494
4.4	Implementación de una solicitud de mantenimiento	494
5.	Administración de mantenimiento	496

6. Cualidades en mantenimiento	499
6.1 Métricas de mantenimiento	499
6.2 Aplicación de métricas de mantenimiento	500
6.3 Posibilidad de mantenimiento	502
7. Resumen	504
Ejercicios	504
Caso de estudio: mantenimiento de <i>Encuentro</i>	506

ACRÓNIMOS **511**

GLOSARIO **515**

REFERENCIAS **519**

CRÉDITOS **523**

APÉNDICE **525**

ÍNDICE **529**
