

Contenido

Prefacio xv

PARTE UNO INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE 1

Capítulo 1

El campo de aplicación de la ingeniería de software 3

Objetivos de aprendizaje 3

- 1.1 Aspectos históricos 4
 - 1.2 Aspectos económicos 6
 - 1.3 Aspectos del mantenimiento 7
 - 1.3.1 *Conceptos tradicional y moderno del mantenimiento* 9
 - 1.3.2 *La importancia del mantenimiento posentrega* 11
 - 1.4 Requisitos, análisis y aspectos de diseño 13
 - 1.5 Aspectos de la formación de equipos 15
 - 1.6 Por qué no hay fase de planeación 16
 - 1.7 Por qué no hay plan fase de pruebas 17
 - 1.8 Por qué no hay fase de documentación 17
 - 1.9 Paradigma orientado a objetos 18
 - 1.10 Paradigma orientado a objetos en perspectiva 23
 - 1.11 Terminología 23
 - 1.12 Temas de ética 26
- Revisión del capítulo 27
Lectura adicional 27
Términos clave 28
Problemas 29
Referencias 30

Capítulo 2

Modelos del ciclo de vida del software 34

Objetivos de aprendizaje 34

- 2.1 Desarrollo del software en teoría 34
- 2.2 Minicaso de estudio Winburg 35

- 2.3 Lecciones del minicaso de estudio Winburg 39
- 2.4 Minicaso de estudio de Teal Tractors
- 2.5 Iteración e incrementación 40
- 2.6 Revisión del minicaso de estudio Winburg 44
- 2.7 Riesgos y otros aspectos de la iteración e incrementación 45
- 2.8 Manejo de iteración e incrementación
- 2.9 Otros modelos del ciclo de vida 39
 - 2.9.1 *Modelo del ciclo de vida de codificación y ajuste* 48
 - 2.9.2 *Modelo del ciclo de vida de cascada* 49
 - 2.9.3 *Modelo del ciclo de vida de elaboración rápida de un prototipo* 51
 - 2.9.4 *Programación extrema y proceso ágil* 52
 - 2.9.5 *Modelo del ciclo de vida de sincronización y estabilización* 54
 - 2.9.6 *Modelo del ciclo de vida en espiral* 54
- 2.10 Comparación de los modelos del ciclo de vida 58
 - Revisión del capítulo 59
 - Lectura adicional 60
 - Términos clave 60
 - Problemas 61
 - Referencias 61

Capítulo 3

El proceso del software 64

- Objetivos de aprendizaje 64
- 3.1 El Proceso Unificado 66
- 3.2 Iteración e incrementación en el paradigma orientado a objetos 67
- 3.3 El flujo de trabajo de los requerimientos 68
- 3.4 El flujo de trabajo del análisis 70
- 3.5 El flujo de trabajo del diseño 72
- 3.6 El flujo de trabajo de la implementación 73
- 3.7 El flujo de trabajo de las pruebas 74
 - 3.7.1 *Artefactos de los requerimientos* 74
 - 3.7.2 *Artefactos del análisis* 74

| | | |
|-------------|--|----|
| | 3.7.3. <i>Artefactos del diseño</i> | 75 |
| | 3.7.4. <i>Artefactos de la implementación</i> | 75 |
| 3.8 | Mantenimiento posentrega | 77 |
| 3.9 | Retiro | 78 |
| 3.10 | Las fases del Proceso Unificado | 78 |
| | 3.10.1 <i>La fase de comienzo</i> | 79 |
| | 3.10.2 <i>La fase de elaboración</i> | 81 |
| | 3.10.3 <i>La fase de construcción</i> | 82 |
| | 3.10.4 <i>La fase de transición</i> | 82 |
| 3.11 | Modelos del ciclo de vida uni contra bidimensionales | 83 |
| 3.12 | Mejoramiento del proceso del software | 84 |
| 3.13 | Modelos de madurez de la capacidad | 85 |
| 3.14 | Otras iniciativas para el mejoramiento de los procesos de software | 88 |
| 3.15 | Costos y beneficios del mejoramiento del proceso del software | 89 |
| | Revisión del capítulo | 91 |
| | Lectura adicional | 91 |
| | Términos clave | 92 |
| | Problemas | 92 |
| | Referencias | 93 |

Capítulo 4 Equipos 96

| | | |
|------------|---|-----|
| | Objetivos de aprendizaje | 96 |
| 4.1 | Organización de equipos | 96 |
| 4.2 | Enfoque del equipo democrático | 98 |
| | 4.2.1 <i>Análisis del enfoque de los equipos democráticos</i> | 99 |
| 4.3 | Enfoque clásico del equipo con un programador jefe | 99 |
| | 4.3.1 <i>El proyecto de New York Times</i> | 101 |
| | 4.3.2 <i>Lo que no es práctico del enfoque clásico del equipo con un programador jefe</i> | 102 |
| 4.4 | Más allá de los equipos de programación con jefe y de los equipos democráticos | 102 |
| 4.5 | Equipos de sincronización y estabilización | 106 |
| 4.6 | Equipos de programación extrema | 106 |
| 4.7 | El modelo de madurez de la capacidad de las personas | 107 |
| 4.8 | Elección de una organización de equipos adecuada | 108 |
| | Revisión del capítulo | 109 |
| | Lectura adicional | 109 |

| | |
|----------------|-----|
| Términos clave | 109 |
| Problemas | 109 |
| Referencias | 110 |

Capítulo 5 Las herramientas del oficio 112

| | | |
|-------------|--|-----|
| | Objetivos de aprendizaje | 112 |
| 5.1 | Depuración paso a paso | 112 |
| | 5.1.1 <i>Minicaso de estudio de la depuración paso a paso</i> | 113 |
| 5.2 | Análisis costo-beneficio | 118 |
| 5.3 | Métrica del software | 119 |
| 5.4 | CASE | 121 |
| 5.5 | Taxonomía de CASE | 122 |
| 5.6 | El alcance de CASE | 123 |
| 5.7 | Versiones del software | 127 |
| | 5.7.1 <i>Revisiones</i> | 127 |
| | 5.7.2 <i>Variaciones</i> | 128 |
| 5.8 | Control de la configuración | 129 |
| | 5.8.1 <i>Control de la configuración durante el mantenimiento posentrega</i> | 131 |
| | 5.8.2 <i>Líneas base</i> | 131 |
| | 5.8.3 <i>Control de la configuración durante el desarrollo</i> | 132 |
| 5.9 | Herramientas de construcción | 132 |
| 5.10 | Ganancias de productividad con la tecnología CASE | 133 |
| | Revisión del capítulo | 135 |
| | Lectura adicional | 135 |
| | Términos clave | 135 |
| | Problemas | 136 |
| | Referencias | 137 |

Capítulo 6 Pruebas 139

| | | |
|------------|---|-----|
| | Objetivos de aprendizaje | 139 |
| 6.1 | Aspectos de calidad | 140 |
| | 6.1.1 <i>Aseguramiento de calidad de software</i> | 141 |
| | 6.1.2 <i>Independencia gerencial</i> | 141 |
| 6.2 | Pruebas basadas en la no ejecución | 142 |
| | 6.2.1 <i>Recorridos</i> | 143 |
| | 6.2.2 <i>La gestión de los recorridos</i> | 143 |
| | 6.2.3 <i>Inspecciones</i> | 144 |
| | 6.2.4 <i>Comparación de las inspecciones y los recorridos</i> | 146 |

- 6.2.5 Fortalezas y debilidades de las revisiones 147
- 6.2.6 Métricas para la inspección 147
- 6.3 Pruebas basadas en la ejecución 147
- 6.4 ¿Qué debe ser probado? 148
 - 6.4.1 Utilidad 149
 - 6.4.2 Confiabilidad 149
 - 6.4.3 Robustez 150
 - 6.4.4 Desempeño 150
 - 6.4.5 Corrección 151
- 6.5 Evaluación comparada con pruebas de corrección 152
 - 6.5.1 Ejemplo de una prueba de corrección 152
 - 6.5.2 Mincaso de estudio de la prueba de corrección 156
 - 6.5.3 Pruebas de corrección e ingeniería de software 157
- 6.6 ¿Quién debe llevar a cabo la prueba basada en la ejecución? 159
- 6.7 Cuándo terminan las pruebas 161
 - Revisión del capítulo 161
 - Lectura adicional 161
 - Términos clave 162
 - Problemas 162
 - Referencias 164

Capítulo 7
De módulos a objetos 166

- Objetivos de aprendizaje 166
- 7.1 ¿Qué es un módulo? 166
- 7.2 Cohesión 170
 - 7.2.1 Cohesión coincidente 170
 - 7.2.2 Cohesión lógica 171
 - 7.2.3 Cohesión temporal 172
 - 7.2.4 Cohesión procedimental 172
 - 7.2.5 Cohesión comunicacional 173
 - 7.2.6 Cohesión funcional 173
 - 7.2.7 Cohesión informacional 174
 - 7.2.8 Ejemplo de cohesión 174
- 7.3 Acoplamiento 175
 - 7.3.1 Acoplamiento por el contenido 176
 - 7.3.2 Acoplamiento común 176
 - 7.3.3 Acoplamiento por control 178
 - 7.3.4 Acoplamiento por molde 178
 - 7.3.5 Acoplamiento de datos 180
 - 7.3.6 Ejemplo de acoplamiento 180
 - 7.3.7 La importancia del acoplamiento 181

- 7.4 Encapsulación de datos 182
 - 7.4.1 Encapsulación de datos y desarrollo 184
 - 7.4.2 Encapsulación de datos y mantenimiento 185
- 7.5 Tipos de datos abstractos 191
- 7.6 Ocultamiento de la información 192
- 7.7 Objetos 194
- 7.8 Herencia, polimorfismo y enlace dinámico 198
- 7.9 El paradigma orientado a objetos 200
 - Revisión del capítulo 203
 - Lectura adicional 203
 - Términos clave 204
 - Problemas 204
 - Referencias 205

Capítulo 8
Reutilización y portabilidad 208

- Objetivos de aprendizaje 208
- 8.1 Conceptos de reutilización 209
- 8.2 Obstáculos para la reutilización 211
- 8.3 Estudios de caso de reutilización 212
 - 8.3.1 División de sistemas de misiles Raytheon 212
 - 8.3.2 Agencia Europea del Espacio 214
- 8.4 Objetos y reutilización 215
- 8.5 Reutilización durante el diseño y la implementación 215
 - 8.5.1 Reutilización del diseño 215
 - 8.5.2 Esquemas de aplicación 217
 - 8.5.3 Patrones de diseño 217
 - 8.5.4 Arquitectura de software 220
 - 8.5.5 Ingeniería de programación basada en componentes 222
- 8.6 Reutilización y mantenimiento posentrega 222
- 8.7 Portabilidad 223
 - 8.7.1 Incompatibilidades del hardware 224
 - 8.7.2 Incompatibilidades del sistema operativo 225
 - 8.7.3 Incompatibilidades del software numérico 225
 - 8.7.4 Incompatibilidades entre los compiladores 226
- 8.8 ¿Por qué portabilidad? 229

- 8.9** Técnicas para obtener portabilidad 230
 - 8.9.1 *Software de sistema portable* 230
 - 8.9.2 *Software de aplicación portable* 231
 - 8.9.3 *Datos portables* 232
- Revisión del capítulo 233
- Lectura adicional 233
- Términos clave 234
- Problemas 234
- Referencias 236

Capítulo 9 Planeación y estimación 240

- Objetivos de aprendizaje 240
- 9.1** Planeación y proceso de software 241
- 9.2** Estimación de duración y costo 242
 - 9.2.1 *Métricas para el tamaño de un producto* 243
 - 9.2.2 *Técnicas de estimación de costos* 247
 - 9.2.3 *COCOMO intermedio* 249
 - 9.2.4 *COCOMO II* 252
 - 9.2.5 *Rastreo de los estimados de duración y costo* 253
- 9.3** Componentes de un plan de administración de proyecto de software 254
- 9.4** Marco de trabajo del plan de administración del proyecto de software 255
- 9.5** Plan de administración de proyectos de software IEEE 257
- 9.6** Planeación de pruebas 260
- 9.7** Planeación de proyectos orientados a objetos 261
- 9.8** Requerimientos de capacitación 261
- 9.9** Estándares de documentación 262
- 9.10** Herramientas CASE para planeación y estimación 263
- 9.11** Prueba del plan de administración de proyectos de software 263
 - Revisión del capítulo 263
 - Lectura adicional 264
 - Términos clave 264
 - Problemas 265
 - Referencias 266

PARTE DOS LOS FLUJOS DE TRABAJO DEL CICLO DE VIDA DEL SOFTWARE 269

Capítulo 10 Requerimientos 271

- Objetivos de aprendizaje 271
- 10.1** Cómo determinar lo que el cliente necesita 271
- 10.2** Repaso del flujo de trabajo de los requerimientos 272
- 10.3** Comprender el dominio 273
- 10.4** El modelo de negocio 274
 - 10.4.1 *Cómo entrevistar* 274
 - 10.4.2 *Otras técnicas* 275
 - 10.4.3 *Casos de uso* 276
- 10.5** Requerimientos iniciales 277
- 10.6** Comprensión inicial del dominio: el caso de estudio Osbert Oglesby 278
- 10.7** Modelo de negocio inicial: el caso de estudio Osbert Oglesby 279
- 10.8** Requerimientos iniciales: el caso de estudio Osbert Oglesby 282
- 10.9** Continuación del flujo de trabajo de los requerimientos: el caso de estudio Osbert Oglesby 284
- 10.10** El flujo de trabajo de las pruebas: el caso de estudio Osbert Oglesby 291
- 10.11** La fase de requerimientos clásica 292
- 10.12** Elaboración rápida de un prototipo 293
- 10.13** Factores humanos 294
- 10.14** Reutilización del prototipo rápido 295
- 10.15** Herramientas CASE para el flujo de trabajo de los requerimientos 296
- 10.16** Métricas para el flujo de trabajo de los requerimientos 297
- 10.17** Retos del flujo de trabajo de los requerimientos 298
 - Revisión del capítulo 299
 - Lectura adicional 299
 - Términos clave 300
 - Términos clave del estudio de caso 300
 - Problemas 300
 - Referencias 301

Capítulo 11**Análisis clásico 303**

- Objetivos de aprendizaje 303
- 11.1 El documento de especificaciones 303
- 11.2 Especificaciones informales 305
 - 11.2.1 Prueba de rectificación minicaso de estudio Redux 306
- 11.3 Análisis de sistemas estructurados 307
 - 11.3.1 Minicaso de estudio de la tienda de Software de Sally 307
- 11.4 Análisis de sistemas estructurados: el caso de estudio Osbert Oglesby 315
- 11.5 Otras técnicas semiformales 316
- 11.6 Modelo entidad-relación 317
- 11.7 Máquinas de estado finito 319
 - 11.7.1 Máquinas de estado finito: el caso de estudio del problema del elevador 321
- 11.8 Redes de Petri 325
 - 11.8.1 Redes de Petri: el caso de estudio del problema del elevador 328
- 11.9 Z 330
 - 11.9.1 Z: el caso de estudio del problema del elevador 330
 - 11.9.2 Análisis de Z 333
- 11.10 Otras técnicas formales 334
- 11.11 Comparación de las técnicas de análisis clásico 335
- 11.12 Probando durante el análisis clásico 335
- 11.13 Herramientas CASE para el análisis clásico 336
- 11.14 Métricas para el análisis clásico 337
- 11.15 Plan de administración del proyecto de software: el caso de estudio Osbert Oglesby 338
- 11.16 Retos del análisis clásico 338
 - Revisión del capítulo 339
 - Lectura adicional 339
 - Términos clave 340
 - Términos clave del caso de estudio 340
 - Problemas 340
 - Referencias 342

Capítulo 12**Análisis orientado a objetos 346**

- Objetivos de aprendizaje 346
- 12.1 El flujo de trabajo del análisis 347
- 12.2 Extracción de las clases de entidad 348
- 12.3 Análisis orientado a objetos: el caso de estudio del problema del elevador 349
- 12.4 Modelado funcional: el caso de estudio del problema del elevador 349
- 12.5 Modelado de clases de entidad: el caso de estudio del problema del elevador 351
 - 12.5.1 Extracción de sustantivos 352
 - 12.5.2 Tarjetas CRC 354
- 12.6 Modelado dinámico: el caso de estudio del problema del elevador 355
- 12.7 El flujo de trabajo de las pruebas: análisis orientado a objetos 358
- 12.8 Extracción de las clases de control y de frontera 362
- 12.9 El modelo funcional inicial: el caso de estudio de Osbert Oglesby 363
- 12.10 El diagrama de clases inicial: el caso de estudio de Osbert Oglesby 365
- 12.11 El modelo dinámico inicial: el caso de estudio de Osbert Oglesby 371
- 12.12 Extracción de las clases de frontera: el caso de estudio de Osbert Oglesby 373
- 12.13 Extracción de las clases de control: el caso de estudio de Osbert Oglesby 374
- 12.14 Refinación de los casos de uso: el caso de estudio de Osbert Oglesby 374
- 12.15 Realización del caso de uso: el caso de estudio de Osbert Oglesby 377
 - 12.15.1 El caso de uso Comprar una Obra Maestra 377
 - 12.15.2 Caso de Uso Comprar un Trabajo Maestro 382
 - 12.15.3 Caso de Uso Comprar Otra Pintura 383
 - 12.15.4 Los cinco casos de uso restantes 384
- 12.16 Incrementos en el diagrama de clases: el caso de estudio Osbert Oglesby 386
- 12.17 El flujo de trabajo de las pruebas: el caso de estudio Osbert Oglesby 388
- 12.18 El documento de especificaciones en el Proceso Unificado 388
- 12.19 Más sobre actores y casos de uso 389
- 12.20 Herramientas CASE para el flujo de trabajo del análisis orientado a objetos 391

- 12.21** Retos del flujo de trabajo del análisis orientado a objetos 391
 - Revisión del capítulo 392
 - Lectura adicional 392
 - Términos clave 393
 - Problemas 393
 - Referencias 395

- Capítulo 13**
Diseño 397
 - Objetivos de aprendizaje 397
- 13.1** Diseño y abstracción 398
- 13.2** Diseño orientado a operaciones 398
- 13.3** Análisis del flujo de datos 399
 - 13.3.1 *Contador de palabras* 400
 - 13.3.2 *Extensiones del análisis del flujo de datos* 405
- 13.4** Análisis de transacciones 405
- 13.5** Diseño orientado a datos 407
- 13.6** Diseño orientado a objetos 408
- 13.7** Diseño orientado a objetos: el caso de estudio del problema del elevador 409
- 13.8** Diseño orientado a objetos: el caso de estudio Osbert Oglesby 412
- 13.9** El flujo de trabajo del diseño 416
- 13.10** El flujo de trabajo de las pruebas: diseño 418
- 13.11** El flujo de trabajo de las pruebas: el caso de estudio Osbert Oglesby 419
- 13.12** Técnicas formales para el diseño detallado 419
- 13.13** Técnicas de diseño de tiempo-real 419
- 13.14** Herramientas CASE para el diseño 421
- 13.15** Métricas para el diseño 421
- 13.16** Retos del flujo de trabajo del diseño 422
 - Revisión del capítulo 423
 - Lectura adicional 423
 - Términos clave 424
 - Problemas 424
 - Referencias 425

- Capítulo 14**
Implementación 428
 - Objetivos de aprendizaje 428
- 14.1** Elección del lenguaje de programación 428
- 14.2** Lenguajes de cuarta generación 431

- 14.3** Buenas prácticas de programación 433
 - 14.3.1 *Uso consistente y significativo de nombres de variables* 434
 - 14.3.2 *El punto de auto-documentar el código* 435
 - 14.3.3 *Uso de parámetros* 436
 - 14.3.4 *Disposición del código para hacer más fácil la lectura* 437
 - 14.3.5 *Sentencias **if** anidadas* 437
- 14.4** Estándares de codificación 439
- 14.5** Reutilización del código 439
- 14.6** Integración 440
 - 14.6.1 *Integración descendente (Top-down)* 440
 - 14.6.2 *Integración ascendente (bottom up)* 442
 - 14.6.3 *Integración sándwich* 443
 - 14.6.4 *Integración de productos orientados a objetos* 443
 - 14.6.5 *Administración de la integración* 445
- 14.7** El flujo de trabajo de la implementación 445
- 14.8** El flujo de trabajo de la implementación: el caso de estudio Osbert Oglesby 445
- 14.9** El flujo de trabajo de las pruebas: implementación 446
- 14.10** Selección de casos de prueba 446
 - 14.10.1 *Prueba de especificaciones contra prueba de código* 446
 - 14.10.2 *Viabilidad de las pruebas de especificaciones* 446
 - 14.10.3 *Viabilidad de la prueba de código* 447
- 14.11** Técnicas de pruebas unitarias de caja-negra 450
 - 14.11.1 *Pruebas de equivalencia y análisis del valor frontera* 450
 - 14.11.2 *Pruebas funcionales* 451
- 14.12** Casos de prueba de caja-negra: el caso de estudio de Osbert Oglesby 452
- 14.13** Técnicas de pruebas unitarias de caja de cristal 455
 - 14.13.1 *Pruebas estructurales: sentencias, ramas y cobertura de ruta* 455
 - 14.13.2 *Métricas de complejidad* 457
- 14.14** Recorridos e inspecciones al código 458
- 14.15** Comparación de las técnicas de pruebas unitarias 458
- 14.16** Cleanroom 459

- 14.17 Problemas potenciales cuando se prueban objetos 460
- 14.18 Aspectos administrativos de las pruebas unitarias 462
- 14.19 Cuándo reescribir en lugar de depurar un artefacto de código 463
- 14.20 Pruebas de integración 464
- 14.21 Pruebas de producto 465
- 14.22 Pruebas de aceptación 466
- 14.23 El flujo de trabajo de las pruebas: el estudio de caso Osbert Oglesby 467
- 14.24 Herramientas CASE para la implementación 467
 - 14.24.1 Herramientas CASE para el proceso completo de software 467
 - 14.24.2 Entornos de desarrollo integrados 468
 - 14.24.3 Entornos para aplicaciones de negocio 469
 - 14.24.4 Infraestructuras de herramientas públicas 469
 - 14.24.5 Problemas potenciales con los entornos 470
- 14.25 Métricas para el flujo de trabajo de la implementación 470
- 14.26 Retos del flujo de trabajo de la implementación 470
 - Revisión del capítulo 471
 - Lectura adicional 472
 - Términos clave 473
 - Problemas 473
 - Referencias 475
- 15.4.3 *Aseguramiento de la capacidad de mantenimiento* 486
- 15.4.4 *El problema del mantenimiento repetido* 486
- 15.5 Mantenimiento del software orientado a objetos 487
- 15.6 Diferencia entre habilidades de mantenimiento posentrega y habilidades de desarrollo 489
- 15.7 Ingeniería en reversa 490
- 15.8 Pruebas hechas durante el mantenimiento posentrega 491
- 15.9 Herramientas CASE para el mantenimiento posentrega 492
- 15.10 Métricas para el mantenimiento posentrega 492
- 15.11 Mantenimiento posentrega: el caso de estudio Osbert Oglesby 493
- 15.12 Retos del mantenimiento posentrega 493
 - Revisión del capítulo 493
 - Lectura adicional 493
 - Términos clave 494
 - Problemas 494
 - Referencias 495

Capítulo 15

Mantenimiento posentrega 479

Objetivos de aprendizaje 479

- 15.1 Por qué es necesario el mantenimiento posentrega 480
- 15.2 ¿Qué se les exige a los programadores durante el mantenimiento posentrega? 480
- 15.3 Mini caso de estudio mantenimiento posentrega 482
- 15.4 Administración del mantenimiento posentrega 484
 - 15.4.1 *Informes de defectos* 484
 - 15.4.2 *La autorización de cambios al producto* 485

Capítulo 16

Más sobre UML 497

Objetivos de aprendizaje 497

- 16.1 UML *no* es una metodología 497
- 16.2 Diagramas de clases 498
 - 16.2.1 *Agregación* 499
 - 16.2.2 *Multiplicidad* 500
 - 16.2.3 *Composición* 501
 - 16.2.4 *Generalización* 502
 - 16.2.5 *Asociación* 502
- 16.3 Notas 503
- 16.4 Diagramas de casos de uso 503
- 16.5 Estereotipos 503
- 16.6 Diagramas de interacción 504
- 16.7 Diagramas de estados 506
- 16.8 Diagramas de actividades 509
- 16.9 Paquetes 511
- 16.10 Diagramas de componentes 511
- 16.11 Diagramas de implementación 511
- 16.12 Repaso de los diagramas UML 511

| | | |
|--------------|-----------------------|-----|
| 16.13 | UML y la iteración | 512 |
| | Revisión del capítulo | 512 |
| | Lectura adicional | 513 |
| | Términos clave | 513 |
| | Problemas | 513 |
| | Referencias | 514 |

Bibliografía 515

Apéndice A

**Proyecto final: el Ophelia's Oasis
en el desierto de Amlet 539**

Apéndice B

**Recursos de la ingeniería
de software 542**

Apéndice C

**Flujo de trabajo de los requerimientos:
el caso de estudio Osbert Oglesby 544**

Apéndice D

**Análisis de sistemas estructurados:
el caso de estudio Osbert Oglesby 545**

Apéndice E

**Flujo de trabajo del análisis: el caso de
estudio Osbert Oglesby 549**

Apéndice F

**Plan de administración del proyecto
de software: el caso de estudio Osbert
Oglesby 550**

Apéndice G

**Flujo de trabajo del diseño: el caso
de estudio Osbert Oglesby 555**

Apéndice H

**Flujo de trabajo de la implementación:
el caso de estudio Osbert Oglesby
(Versión C++) 563**

Apéndice I

**Flujo de trabajo de la implementación:
el caso de estudio Osbert Oglesby
(Versión Java) 564**

Apéndice J

**Flujo de trabajo de las pruebas: el caso
de estudio Osbert Oglesby 565**

Índice de autores 567

Índice analítico 570