

Contenido

Prefacio	xiii
Agradecimientos	xvii

Parte I: Conceptos básicos

Capítulo 1: Ingeniería del software con componentes	3
1.1. ¿Qué es un buen sistema?	4
1.2. ¿Se tienen buenos sistemas?	4
1.2.1. Problemas...	4
1.2.2. ...fallos incluso drásticos	6
1.2.3. Promesas, promesas	7
1.3. ¿Cómo son los buenos sistemas?	8
1.3.1. Encapsulación: débil acoplamiento	9
1.3.2. Abstracción: fuerte cohesión	12
1.3.3. Arquitectura y componentes	13
1.3.4. Desarrollo basado en componentes: conectividad	15
1.4. ¿Cómo se construye un buen sistema?	16
Capítulo 2: Conceptos de objetos	17
2.1. ¿Qué es un objeto?	18
2.1.1. Ejemplo	19
2.1.2. Mensajes	19
2.1.3. Interfaces	20
2.1.4. Clases	21
2.2. ¿Cómo relacionar esto con los objetivos del capítulo anterior?	24
2.2.1. ¿Qué tienen que ver los objetos con los componentes?	25
2.3. Herencia	26
2.4. Polimorfismo y ligadura dinámica	28
Capítulo 3: Estudio de un caso introductorio	33
3.1. El problema	33
3.1.1. Clarificación de los requisitos	34
3.1.2. Modelo de casos de uso	35

3.2.	Alcance e iteraciones	37
3.3.	Identificación de clases	39
3.4.	Relaciones entre clases	42
3.5.	El sistema en acción	44
3.5.1.	Cambios en el sistema: diagramas de estado	48
3.5.2.	Más trabajo	49

Capítulo 4: El proceso de desarrollo 53

4.1.	Definición de términos	53
4.1.1.	Modelos y lenguajes de modelado	54
4.1.2.	Proceso y calidad	56
4.2.	El proceso de desarrollo	56
4.2.1.	¿Una metodología unificada?	59
4.2.2.	Procesos a utilizar con UML	60
4.3.	Sistema, diseño, modelo, diagrama	62

Parte II: El lenguaje unificado de modelado

Capítulo 5: Fundamentos de los modelos de clases 67

5.1.	Identificación de objetos y clases	68
5.1.1.	¿Qué hace que un modelo de clases sea bueno?	68
5.1.2.	Cómo construir un buen modelo de clases	69
5.1.3.	¿Qué son las clases?	71
5.1.4.	Objetos del mundo real frente a su representación en el sistema .	71
5.2.	Asociaciones	72
5.2.1.	Multiplicidades	75
5.3.	Atributos y operaciones	76
5.3.1.	Operaciones	76
5.3.2.	Atributos	76
5.4.	Generalización	77
5.4.1.	Utilización del lenguaje para comprobar si existe una generaliza- ción	80
5.4.2.	Implementación de la generalización: herencia	81
5.5.	El modelo de clases durante el desarrollo	82
5.6.	Tarjetas CRC	82
5.6.1.	Creación de tarjetas CRC	83
5.6.2.	Utilización de tarjetas CRC en el desarrollo del diseño	83
5.6.3.	Ejemplo de tarjeta CRC	84
5.6.4.	Refactorización	85

Capítulo 6: Más sobre los modelos de clases 87

6.1.	Más sobre asociaciones	88
6.1.1.	Agregación y composición	88

6.1.2.	Roles	90
6.1.3.	Navegabilidad	90
6.1.4.	Asociaciones calificadas	92
6.1.5.	Asociaciones derivadas	93
6.1.6.	Restricciones	94
6.1.7.	Clases de asociación	97
6.2.	Más sobre clases	98
6.2.1.	Interfaces	99
6.2.2.	Clases abstractas	101
6.3.	Clases parametrizadas	103
6.4.	Dependencia	104
6.5.	Componentes y paquetes	105
6.6.	Visibilidad, protección	105
Capítulo 7: Fundamentos de los modelos de casos de uso		107
7.1.	Actores en detalle	109
7.2.	Casos de uso en detalle	111
7.3.	Límite del sistema	112
7.4.	Utilización de los casos de uso	113
7.4.1.	Casos de uso para la captura de requisitos	113
7.4.2.	Casos de uso a través del desarrollo	114
7.5.	Posibles problemas con los casos de uso	116
Capítulo 8: Más sobre los modelos de casos de uso		119
8.1.	Relaciones entre casos de uso	119
8.1.1.	Casos de uso para la reutilización: <<include>>	120
8.1.2.	Componentes y casos de uso	122
8.1.3.	Separación del comportamiento variable: <<extend>>	124
8.2.	Generalizaciones	125
8.3.	Actores y clases	126
8.3.1.	Notación: actores y clases	127
Capítulo 9: Fundamentos de los diagramas de interacción		129
9.1.	Colaboraciones	130
9.2.	Interacciones en los diagramas de colaboración	131
9.3.	Diagramas de secuencia	134
9.4.	Características más avanzadas	137
9.4.1.	Mensajes desde un objeto a sí mismo	137
9.4.2.	Eliminar el comportamiento detallado	138
9.4.3.	Valores de retorno	139
9.4.4.	Creación y borrado de objetos	140
9.4.5.	Sincronización	142
9.5.	Diagramas de interacción para otros propósitos	143
9.5.1.	Muestra de cómo una clase proporciona una operación	144

9.5.2.	Descripción de cómo funciona un patrón de diseño	144
9.5.3.	Descripción de cómo utilizar un componente	144
Capítulo 10:	Más sobre los diagramas de interacción	145
10.1.	Diagramas de interacción genéricos	145
10.1.1.	Comportamiento condicional	146
10.1.2.	Iteración	148
10.2.	Concurrencia	148
10.2.1.	Modelización de varios hilos de control	151
Capítulo 11:	Fundamentos de los diagramas de estado y de actividad .	155
11.1.	Diagramas de estado	156
11.1.1.	Mensajes no esperados	157
11.1.2.	Nivel de abstracción	158
11.1.3.	Estado, transiciones, eventos	158
11.1.4.	Acciones	159
11.1.5.	Guardas	160
11.2.	Diagramas de actividad	163
Capítulo 12:	Más sobre los diagramas de estado y de actividad	167
12.1.	Otros tipos de eventos	167
12.2.	Otros tipos de acciones	168
12.3.	Análisis de los estados	169
12.4.	Concurrencia dentro de los estados	171
Capítulo 13:	Diagramas de implementación	173
13.1.	Modelo de componentes	174
13.2.	Modelo de despliegue	177
13.2.1.	La capa física	178
13.2.2.	Despliegue del software sobre el hardware	178
Capítulo 14:	Paquetes, subsistemas, modelos	181
14.1.	Paquetes	181
14.1.1.	Control del espacio de nombres	182
14.2.	Subsistemas	186
14.3.	Modelos	187
Parte III:	Estudio de casos	
Capítulo 15:	Administración del CS4	191
15.1.	El estudio del caso	191
15.1.1.	Modelo de clases	195

15.1.2.	Dinámica	196
15.1.3.	Diagramas de estado	196
15.1.4.	Diagramas de actividad	197
15.2.	Discusión	197
Capítulo 16: Juegos de mesa		199
16.1.	Alcance y análisis preliminar	200
16.1.1.	Tres en raya	201
16.1.2.	Ajedrez	201
16.2.	Interacción	205
16.3.	De vuelta al marco de trabajo (<i>framework</i>)	208
16.4.	Estados	210
Capítulo 17: Simulación de eventos discretos		213
17.1.	Requisitos	214
17.1.1.	Descripción más detallada	214
17.2.	Esquema general del modelo de clases	216
17.3.	Casos de uso	218
17.3.1.	Resumen de crear modelo	219
17.3.2.	Resumen de observar comportamiento	219
17.3.3.	Resumen de recoger estadísticas	219
17.3.4.	Resumen de ejecutar un modelo	219
17.4.	Mecanismos estándar para la simulación basada en el proceso	220
17.5.	Asociaciones y navegabilidad	221
17.6.	Clases en detalle	224
17.6.1.	Clase Planificador	225
17.6.2.	Clase EntidadActiva	225
17.6.3.	Clase EntidadPasiva	227
17.6.4.	Clase Recurso	227
17.7.	Clase Informe	230
17.8.	Clase Estadística	230
17.8.1.	Clase Media	231
17.9.	Construcción de un modelo de simulación completo	232
17.10.	La cena de los filósofos	232
Parte IV: Hacia la práctica		
Capítulo 18: Reutilización: componentes, patrones		237
18.1.	Factibilidad de la reutilización	237
18.1.1.	¿Qué se puede reutilizar, y cómo?	238
18.1.2.	¿Por qué reutilizar?	240
18.1.3.	¿Por qué es difícil reutilizar?	241
18.1.4.	¿Qué componentes son verdaderamente reutilizables?	243
18.1.5.	¿Qué pasa con la construcción de componentes propios?	244

18.1.6.	¿Qué diferencias introduce la orientación a objetos?	245
18.2.	Patrones de diseño	245
18.2.1.	Ejemplo: Fachada	248
18.2.2.	UML y patrones	249
18.3.	Marcos de trabajo	250
Capítulo 19: Calidad del producto: verificación, validación, prueba ..		253
19.1.	Revisión de la calidad	254
19.2.	¿Cómo se alcanza la alta calidad?	254
19.2.1.	Centrada en el producto	254
19.2.2.	Centrada en el proceso	255
19.2.3.	Más información	255
19.3.	Verificación	255
19.4.	Validación	256
19.4.1.	Usabilidad	257
19.5.	Prueba	258
19.5.1.	Selección y ejecución de las pruebas	259
19.5.2.	Problemas particulares de la OO	261
19.5.3.	¿Por qué las pruebas se hacen de forma incorrecta tan a menudo?	263
19.6.	Revisiones e inspecciones	264
19.6.1.	Problemas de las RTFs	265
Capítulo 20: Calidad del proceso: gestión, equipos, QA (GC)		267
20.1.	Gestión	267
20.1.1.	Gestión de proyectos	268
20.1.2.	Estimación de un proyecto iterativo	269
20.1.3.	Gestión del desarrollo basado en componentes	271
20.1.4.	Gestión del personal	271
20.2.	Equipos	272
20.3.	Liderazgo	273
20.3.1.	Reforma del proceso de desarrollo	275
20.4.	Garantía de calidad	275
20.4.1.	Garantía de calidad en proyectos iterativos	277
20.4.2.	Gestión de la Calidad Total (TQM)	277
20.5.	Más información	280
Bibliografía		283
Índice alfabético		287