

Índice general

1. Introducción	19
1.1. Introducción	19
1.2. Objetivos	20
1.3. ¿Qué es el software?	20
1.3.1. Tipos de software	21
1.4. ¿Cómo se fabrica el software?	22
1.4.1. Calidad del software	25
1.4.2. Fábrica de software	28
1.5. Mitos del software	28
1.6. Conclusiones	30
1.7. Ejercicios propuestos	30
2. El Ciclo de Vida del Software	31
2.1. Introducción	31
2.2. Objetivos	31
2.3. El ciclo de vida de un producto	31
2.4. El ciclo de vida del software	33
2.5. Fases del ciclo de vida del software	34
2.5.1. Análisis	34
2.5.2. Diseño	34
2.5.3. Codificación	35
2.5.4. Integración	35
2.5.5. Explotación	35
2.5.6. Mantenimiento	35
2.6. Documentos que se generan en el ciclo de vida	36
2.6.1. Documento de requisitos del software	36
2.6.2. Documento de diseño del software	36
2.6.3. Código fuente	36
2.6.4. El sistema software	36
2.6.5. Documentos de cambios	36
2.7. Tipos de ciclo de vida del software	37
2.7.1. Ciclo de vida en cascada	37
2.7.2. Ciclo de vida en V	38
2.8. Prototipos	40
2.8.1. Prototipos rápidos	41
2.8.2. Prototipos evolutivos	42
2.9. El modelo en espiral	43

2.10. Programación extrema	45
2.11. Mantenimiento del software	47
2.11.1. Evolución de las aplicaciones	48
2.11.2. Gestión de cambios	49
2.11.3. Reingeniería	49
2.12. Garantía de calidad de software	50
2.12.1. Plan de garantía de calidad	50
2.12.2. Revisiones	51
2.12.3. Pruebas	52
2.12.4. Gestión de configuración	52
2.12.5. Normas y estándares	54
2.13. Conclusiones	56
2.14. Ejercicios propuestos	56
3. Especificación de Requisitos	57
3.1. Introducción	57
3.2. Objetivos	57
3.3. Modelado de sistemas	58
3.3.1. Concepto de modelo	59
3.3.2. Técnicas de modelado	59
3.4. Análisis de requisitos de software	64
3.4.1. Objetivos del análisis	65
3.4.2. Tareas del análisis	68
3.5. Notaciones para la especificación	71
3.5.1. Lenguaje natural	72
3.5.2. Diagramas de flujo de datos	73
3.5.3. Diagramas de transición de estados	79
3.5.4. Descripciones funcionales. Pseudocódigo	81
3.5.5. Descripción de datos	84
3.5.6. Diagramas de modelo de datos	86
3.6. Documento de especificación de requisitos	91
3.7. Ejemplos de especificaciones	99
3.7.1. Videojuego de las minas	99
3.7.2. Sistema de gestión de biblioteca	104
3.8. Conclusiones	115
3.9. Ejercicios propuestos	116
3.10. Ejercicios resueltos	116
4. Fundamentos del Diseño de Software	121
4.1. Introducción	121
4.2. Objetivos	121
4.3. ¿Qué es el diseño?	122
4.4. Conceptos de base	125
4.4.1. Abstracción	126
4.4.2. Modularidad	129
4.5. Refinamiento	130
4.5.1. Estructuras de datos	131
4.5.2. Ocultación	132

4.5.3.	Genericidad	133
4.5.4.	Herencia	135
4.5.5.	Polimorfismo	137
4.5.6.	Concurrencia	139
4.6.	Notaciones para el diseño	140
4.6.1.	Notaciones estructurales	141
4.6.2.	Notaciones estáticas	148
4.7.	Documentos de diseño	158
4.7.1.	Documento ADD	158
4.7.2.	Documento DDD	162
4.8.	Conclusiones	164
4.9.	Ejercicios propuestos	164
5.	Técnicas Generales de Diseño de Software	165
5.1.	Introducción	165
5.2.	Objetivos	165
5.3.	Descomposición Modular	166
5.3.1.	Independencia funcional	168
5.3.2.	Acoplamiento	168
5.3.3.	Cohesión	172
5.3.4.	Comprensibilidad	175
5.3.5.	Adaptabilidad	175
5.4.	Técnicas de diseño funcional descendente	177
5.4.1.	Desarrollo por refinamiento progresivo	177
5.4.2.	Diseño estructurado	182
5.4.3.	Ejemplo: Sistema de gestión de biblioteca	185
5.5.	Técnicas de diseño basado en abstracciones	186
5.5.1.	Descomposición modular basada en abstracciones	187
5.5.2.	Método de Abbott	188
5.5.3.	Ejemplo: Videjuego de las minas	191
5.6.	Técnicas de diseño orientadas a objetos	193
5.6.1.	Diseño orientado a objetos	194
5.6.2.	Ejemplo: Estación meteorológica	196
5.7.	Técnicas de diseño de datos	204
5.8.	Diseño de bases de datos relacionales	204
5.8.1.	Formas normales	205
5.8.2.	Diseño de las entidades	207
5.8.3.	Tratamiento de las relaciones de asociación	207
5.8.4.	Tratamiento de las relaciones de composición	209
5.8.5.	Tratamiento de la herencia	209
5.8.6.	Diseño de Índices	209
5.8.7.	Ejemplo: Diseño de datos para la gestión de biblioteca	209
5.9.	Diseño de bases de datos de objetos	212
5.10.	Diseño de software con patrones	213
5.11.	Ejemplos de diseños	218
5.11.1.	Ejemplo: Videjuego de las minas	218
5.11.2.	Ejemplo: Sistema de gestión de biblioteca	230
5.12.	Conclusiones	239

5.13. Ejercicios propuestos	240
5.14. Ejercicios resueltos	240
6. UML, Lenguaje Unificado de Modelado	243
6.1. Introducción	243
6.2. Objetivos	243
6.3. ¿Qué es UML?	244
6.4. Orígenes de UML	245
6.5. Objetivos de UML	246
6.6. Estructura de UML	247
6.7. Diagramas UML	254
6.7.1. Diagramas de casos de uso	254
6.7.2. Diagrama de clases	257
6.7.3. Diagramas de secuencia	260
6.7.4. Diagramas de colaboración	261
6.7.5. Diagrama de estado	263
6.7.6. Diagrama de actividad	265
6.7.7. Diagramas de componentes	266
6.7.8. Diagrama de despliegue	267
6.8. Conclusiones	268
6.9. Ejercicios propuestos	268
6.10. Ejercicios resueltos	269
7. La Codificación del Software	273
7.1. Introducción	273
7.2. Objetivos	273
7.3. Los lenguajes de programación	274
7.3.1. Primera generación de lenguajes	274
7.3.2. Segunda generación	275
7.3.3. Tercera generación	277
7.3.4. Cuarta generación	280
7.4. Criterios de selección del lenguaje	282
7.5. Aspectos metodológicos	284
7.5.1. Normas y estilo de codificación	284
7.5.2. Manejo de errores	286
7.5.3. Aspectos de eficiencia	289
7.5.4. Transportabilidad de software	290
7.6. Conclusiones	292
7.7. Ejercicios propuestos	293
8. Pruebas de Software	295
8.1. Introducción	295
8.2. Objetivos	296
8.3. Tipos de pruebas	296
8.4. Pruebas de unidades	297
8.4.1. Pruebas de caja negra	298
8.4.2. Pruebas de caja transparente	304
8.4.3. Estimación de errores no detectados	309

8.5. Pruebas de unidades en programación orientada a objetos	310
8.6. Estrategias de Integración	311
8.6.1. Integración Big Bang	311
8.6.2. Integración descendente	312
8.6.3. Integración ascendente	313
8.6.4. Estrategias de integración en programación orientada a objetos	314
8.7. Pruebas de validación	315
8.8. Pruebas del sistema	316
8.9. Conclusiones	317
8.10. Ejercicios propuestos	318
8.11. Ejercicios resueltos	319