

Índice general

1. Introducción	17
1.1. Introducción	17
1.2. Objetivos	18
1.3. ¿Qué es el software?	18
1.3.1. Calidad del software	19
1.3.2. Tipos de software	21
1.4. ¿Cómo se fabrica el software?	23
1.5. Mitos del software	25
1.6. Conclusiones	27
1.7. Ejercicios propuestos	27
2. El Ciclo de Vida del Software	29
2.1. Introducción	29
2.2. Objetivos	29
2.3. El ciclo de vida de un producto	29
2.4. El ciclo de vida del software	31
2.5. Fases del ciclo de vida del software	32
2.5.1. Análisis	32
2.5.2. Diseño	32
2.5.3. Codificación	33
2.5.4. Integración	33
2.5.5. Explotación	33
2.5.6. Mantenimiento	33
2.6. Documentos que se generan en el ciclo de vida	33
2.6.1. Documento de requisitos del software	34
2.6.2. Documento de diseño del software	34
2.6.3. Código fuente	34
2.6.4. El sistema software	34
2.6.5. Documentos de cambios	34
2.7. Tipos de ciclo de vida del software	35
2.7.1. Ciclo de vida en cascada	35
2.7.2. Ciclo de vida en V	36
2.8. Prototipos	38
2.8.1. Prototipos rápidos	39
2.8.2. Prototipos evolutivos	40
2.9. El modelo en espiral	41
2.10. Programación extrema	43

2.11.	Mantenimiento del software	45
2.11.1.	Evolución de las aplicaciones	46
2.11.2.	Gestión de cambios	47
2.11.3.	Reingeniería	47
2.12.	Garantía de calidad de software	48
2.12.1.	Plan de garantía de calidad	48
2.12.2.	Revisiones	49
2.12.3.	Pruebas	50
2.12.4.	Gestión de configuración	50
2.12.5.	Normas y estándares	52
2.13.	Conclusiones	54
2.14.	Ejercicios propuestos	54
3.	Especificación de Requisitos	55
3.1.	Introducción	55
3.2.	Objetivos	55
3.3.	Modelado de sistemas	56
3.3.1.	Concepto de modelo	57
3.3.2.	Técnicas de modelado	57
3.4.	Análisis de requisitos de software	62
3.4.1.	Objetivos del análisis	63
3.4.2.	Tareas del análisis	66
3.5.	Notaciones para la especificación	69
3.5.1.	Lenguaje natural	70
3.5.2.	Diagramas de flujo de datos	71
3.5.3.	Diagramas de transición de estados	77
3.5.4.	Descripciones funcionales. Pseudocódigo	79
3.5.5.	Descripción de datos	82
3.5.6.	Diagramas de modelo de datos	84
3.6.	Documento de especificación de requisitos	87
3.7.	Ejemplos de especificaciones	95
3.7.1.	Videojuego de las minas	95
3.7.2.	Sistema de gestión de biblioteca	100
3.8.	Conclusiones	111
3.9.	Ejercicios propuestos	112
4.	Fundamentos del Diseño de Software	113
4.1.	Introducción	113
4.2.	Objetivos	113
4.3.	¿Qué es el diseño?	114
4.4.	Conceptos de base	117
4.4.1.	Abstracción	118
4.4.2.	Modularidad	121
4.5.	Refinamiento	122
4.5.1.	Estructuras de datos	123
4.5.2.	Ocultación	124
4.5.3.	Genericidad	125
4.5.4.	Herencia	127

4.5.5.	Polimorfismo	129
4.5.6.	Concurrencia	131
4.6.	Notaciones para el diseño	132
4.6.1.	Notaciones estructurales	133
4.6.2.	Notaciones estáticas	140
4.7.	Documentos de diseño	150
4.7.1.	Documento ADD	150
4.7.2.	Documento DDD	154
4.8.	Conclusiones	155
4.9.	Ejercicios propuestos	156
5.	Técnicas Generales de Diseño de Software	157
5.1.	Introducción	157
5.2.	Objetivos	157
5.3.	Descomposición Modular	158
5.3.1.	Independencia funcional	159
5.3.2.	Adaptabilidad	167
5.4.	Técnicas de diseño funcional descendente	169
5.4.1.	Desarrollo por refinamiento progresivo	169
5.4.2.	Diseño estructurado	174
5.4.3.	Ejemplo: Sistema de gestión de biblioteca	177
5.5.	Técnicas de diseño basado en abstracciones	178
5.5.1.	Descomposición modular basada en abstracciones	179
5.5.2.	Método de Abbott	180
5.5.3.	Ejemplo: Videjuego de las minas	183
5.6.	Técnicas de diseño orientadas a objetos	185
5.6.1.	Diseño orientado a objetos	186
5.6.2.	Ejemplo: Estación meteorológica	188
5.7.	Técnicas de diseño de datos	196
5.8.	Diseño de bases de datos relacionales	196
5.8.1.	Formas normales	197
5.8.2.	Diseño de las entidades	199
5.8.3.	Tratamiento de las relaciones de asociación	199
5.8.4.	Tratamiento de las relaciones de composición	201
5.8.5.	Tratamiento de la herencia	201
5.8.6.	Diseño de Índices	201
5.8.7.	Ejemplo: Diseño de datos para la gestión de biblioteca	201
5.9.	Diseño de bases de datos de objetos	204
5.10.	Diseño de software con patrones	205
5.11.	Ejemplos de diseños	210
5.11.1.	Ejemplo: Videjuego de las minas	210
5.11.2.	Ejemplo: Sistema de gestión de biblioteca	222
5.12.	Conclusiones	231
5.13.	Ejercicios propuestos	232
6.	UML, Lenguaje Unificado de Modelado	233
6.1.	Introducción	233
6.2.	Objetivos	233

6.3.	¿Qué es UML?	234
6.4.	Orígenes de UML	235
6.5.	Objetivos de UML	236
6.6.	Estructura de UML	237
6.7.	Diagramas UML	244
6.7.1.	Diagramas de casos de uso	244
6.7.2.	Diagrama de clases	247
6.7.3.	Diagramas de secuencia	250
6.7.4.	Diagramas de colaboración	251
6.7.5.	Diagrama de estado	253
6.7.6.	Diagrama de actividad	255
6.7.7.	Diagramas de componentes	256
6.7.8.	Diagrama de despliegue	257
6.8.	Conclusiones	258
6.9.	Ejercicios propuestos	258
7.	La Codificación del Software	261
7.1.	Introducción	261
7.2.	Objetivos	261
7.3.	Los lenguajes de programación	262
7.3.1.	Primera generación de lenguajes	262
7.3.2.	Segunda generación	263
7.3.3.	Tercera generación	265
7.3.4.	Cuarta generación	268
7.4.	Criterios de selección del lenguaje	270
7.5.	Aspectos metodológicos	272
7.5.1.	Normas y estilo de codificación	272
7.5.2.	Manejo de errores	274
7.5.3.	Aspectos de eficiencia	277
7.5.4.	Transportabilidad de software	278
7.6.	Conclusiones	280
7.7.	Ejercicios propuestos	281
8.	Pruebas de Software	283
8.1.	Introducción	283
8.2.	Objetivos	284
8.3.	Tipos de pruebas	284
8.4.	Pruebas de unidades	285
8.4.1.	Pruebas de caja negra	286
8.4.2.	Pruebas de caja transparente	291
8.4.3.	Estimación de errores no detectados	296
8.5.	Pruebas de unidades en programación orientada a objetos	297
8.6.	Estrategias de Integración	298
8.6.1.	Integración Big Bang	298
8.6.2.	Integración descendente	298
8.6.3.	Integración ascendente	300
8.6.4.	Estrategias de integración en programación orientada a objetos	301
8.7.	Pruebas de validación	301

8.8. Pruebas del sistema	302
8.9. Conclusiones	303
8.10. Ejercicios propuestos	304