

---

# CONTENIDO

---

Prefacio	xiii
<b>Capítulo 1</b> <b>Introducción a la <u>ingeniería de software</u></b>	<b>1</b>
Introducción	1
1.1 Definiciones	5
1.2 Factores de tamaño	8
1.2.1 Esfuerzo dedicado al software	8
1.2.2 Distribución del esfuerzo	10
1.2.3 Categorías de acuerdo con el tamaño	11
1.2.4 <u>Cómo usan su tiempo los programadores</u>	14
1.3 Factores de calidad y productividad	14
1.4 Conceptos de administración	25
1.5 Descripción general del contenido del texto	27
1.6 Notas finales	28
Referencias	30
Ejercicios	30
<b>Capítulo 2</b> <b>Planeación de un proyecto de programación</b>	<b>32</b>
Introducción	32
2.1 Definición del problema	32
2.1.1 Metas y requisitos	34
2.2 Desarrollo de una estrategia de solución	38
2.3 <u>Planeación del proceso de desarrollo</u>	40
2.3.1 Modelo de las fases del ciclo de vida	40
2.3.2 Logros, documentos y revisiones	44
2.3.3 Modelo del costo de un proyecto	50
2.3.4 <u>Modelo de prototipo para el ciclo de vida</u>	52
2.3.5 Versiones sucesivas	55

2.4	Planeación de una estructura organizacional	56
2.4.1	Estructura del proyecto	56
	<i>Formato de proyecto/Formato funcional/ Formato matricial</i>	
2.4.2	Estructura del grupo de programación	58
	<i>Grupos democráticos/Grupos con jefe de programación/Grupos bajo jerarquía administrativa</i>	
2.4.3	Administración por objetivos	61
2.5	Otras actividades en la planeación	62
2.5.1	Planeación para la administración de la configuración y el control de calidad	62
2.5.2	Planeación para verificación y validación externas	63
2.5.3	Planeación de las herramientas y técnicas específicas de cada fase	63
2.5.4	Otras actividades de planeación	63
2.6	Resumen	63
	Referencias	64
	Ejercicios	64
<b>Capítulo 3</b>	<b>Estimación de costos del software</b>	66
	Introducción	66
3.1	Factores en el costo del software	67
3.1.1	Capacidad del programador	67
3.1.2	Complejidad del producto	68
3.1.3	Tamaño del producto	71
3.1.4	Tiempo disponible	72
3.1.5	Nivel de confiabilidad requerido	73
3.1.6	Nivel tecnológico	74
3.2	Técnicas de estimación de costos del software	75
3.2.1	Juicio experto	75
3.2.2	Estimación del costo por la técnica DELFI	76
3.2.3	Estructuras de división de trabajo	78
3.2.4	Modelos de costo por algoritmos o módulos	79
3.3	Estimación del nivel de contratación	83
3.4	Estimación de los <u>costos de mantenimiento de software</u>	86
3.5	Resumen	88
	Referencias	89
	Ejercicios	90
<b>Capítulo 4</b>	<b>Definición de requisitos para la producción de software</b>	92
	Introducción	92
4.1	Especificación de requisitos para la producción de software	93
4.2	Técnicas formales de especificación	98

4.2.1	Notaciones relacionales <i>Ecuaciones implícitas/Relaciones recurrentes/ Axiomas algebraicos/Expresiones regulares</i>	99
4.2.2	Notaciones de estados <i>Tablas de decisión/Tablas de eventos/Tablas de transición/Mecanismos de estados finitos/ Redes de Petri</i>	108
4.2.3	Resumen	119
4.3	Lenguajes y paquetería para la especificación de requisitos	120
4.3.1	PSL/PSA	121
4.3.2	RSL/REVS	124
4.3.3	Técnicas de análisis y diseño estructurado (SADT)	128
4.3.4	Análisis estructurado de sistemas (SSA)	133
4.3.5	GIST	136
4.4	Resumen	139
	Referencias	139
	Ejercicios	140
<b>Capítulo 5</b>	<b>Diseño de software</b>	<b>144</b>
	Introducción	144
5.1	Conceptos fundamentales de diseño	146
5.1.1	Abstracción	146
5.1.2	Cobertura interna de la información	148
5.1.3	Estructura	149
5.1.4	Modularidad	153
5.1.5	Concurrencia	154
5.1.6	Verificación	154
5.1.7	Estética	155
5.2	Módulos y criterios de modulación	155
5.2.1	Acoplamiento y cohesión	156
5.2.2	Otros criterios	160
5.3	Notaciones para el diseño	161
5.3.1	Diagramas de flujo de datos	161
5.3.2	Cartas de estructura	163
5.3.3	Diagramas tipo HIPO	164
5.3.4	Patrones de procedimientos	164
5.3.5	Seudocódigo	168
5.3.6	Diagramas de flujo estructurados	168
5.3.7	Lenguaje natural estructurado	171
5.3.8	Tablas de decisión	171
5.4	Técnicas de diseño	171
5.4.1	Refinamiento por pasos	173
5.4.2	Niveles de abstracción	177
5.4.3	Diseño estructurado	178

5.4.4	Desarrollo integrado jerárquico hacia abajo	183
5.4.5	Programación estructurada de Jackson	185
5.4.6	Resumen de técnicas de diseño	191
5.5	Consideraciones de diseño al detalle	193
5.6	Diseño en sistemas distribuidos y de tiempo real	194
5.7	Planes de pruebas piloto	196
5.8	Logros, recorridos e inspecciones	198
5.8.1	Recorridos e inspecciones	199
5.9	Directrices de diseño	200
5.10	Resumen	201
	Referencias	202
	Ejercicios	203
<b>Capítulo 6</b>	<b>Aspectos de la instrumentación</b>	<b>205</b>
	Introducción	205
6.1	Técnicas de codificación estructurada	206
6.1.1	Una entrada, una salida	207
6.1.2	Consideraciones de eficiencia	208
6.1.3	Violaciones a una entrada, una salida	215
6.1.4	Encapsulado de datos	216
6.1.5	La proposición Goto	217
6.1.6	Recursividad	220
6.2	Estilo en la codificación	223
6.3	Estándares y directrices	233
6.4	Directrices de documentación	234
6.4.1	Documentos de apoyo	234
6.4.2	Notas de unidad de programa	235
6.4.3	Documentación interna	237
6.5	Resumen	239
	Referencias	240
	Ejercicios	240
<b>Capítulo 7</b>	<b>Aspectos de un lenguaje de programación moderno</b>	<b>243</b>
	Introducción	243
7.1	Verificación de tipos de datos	245
7.1.1	Lenguajes sin caracterización de tipos	246
7.1.2	Coerción automática de tipos	246
7.1.3	Modos mixtos	247
7.1.4	Verificación robusta de tipos	248
7.1.5	Verificación ligera de tipos	252
7.2	Compilación separada	252
7.3	Definición de tipos por el usuario	253

7.3.1	Subtipos	254
7.3.2	Tipos derivados	255
7.3.3	Tipos enumerables	256
7.3.4	Registros	257
7.3.5	Apuntadores	258
7.4	Abstracción de datos	260
7.4.1	Encapsulado de datos	261
7.4.2	Tipos de datos abstractos	262
7.4.3	Facilidades genéricas	264
7.5	Reglas de alcance	267
7.6	Manejo de excepciones	269
7.6.1	Manejo de excepciones en PL/1	269
7.6.2	Manejo de excepciones en Ada	270
7.7	Mecanismos de concurrencia	274
7.7.1	Variables compartidas	274
7.7.2	Paso asincrónico de mensajes	276
7.7.3	Paso sincrónico de mensajes	276
7.8	Resumen	281
	Referencias	283
	Ejercicios	283
<b>Capítulo 8</b>	<b>Técnicas de verificación y validación</b>	<b>286</b>
	Introducción	286
8.1	Control de calidad	288
8.2	Recorridos e inspecciones	291
8.2.1	Recorridos	292
8.2.2	Inspecciones	294
8.3	Análisis estático	295
8.4	Ejecución simbólica	300
8.5	Pruebas de unidad y depuración	304
8.5.1	Pruebas de unidad	304
8.5.2	Depuración	309
8.6	Pruebas del sistema	315
8.6.1	Pruebas de integración	316
8.6.2	Pruebas de aceptación	319
8.7	Verificación formal	320
8.7.1	Afirmaciones de entrada y salida	320
8.7.2	Precondiciones más débiles	322
8.7.3	Inducción estructural	326
8.8	Resumen	330
	Referencias	330
	Ejercicios	332
<b>Capítulo 9</b>	<b>Mantenimiento de software</b>	<b>334</b>
	Introducción	334

9.1	Mejoramiento del mantenimiento durante el desarrollo	336
9.2	Aspectos administrativos del mantenimiento de software	339
9.3	Administración de la configuración	344
9.4	Métricas del código fuente	346
9.5	Otras herramientas y técnicas de mantenimiento	350
9.6	Resumen	352
	Referencias	353
<b>Capítulo 10</b>	<b>Resumen</b>	<b>355</b>
	Introducción	355
10.1	Planeación y estimación de costos	356
10.2	Definición de los requisitos para la producción de software	358
10.3	Conceptos de diseño de software	359
10.4	Aspectos de la implantación	360
10.5	Características de los lenguajes modernos	361
10.6	Técnicas de verificación y validación	362
10.7	Mantenimiento de software	364
10.8	Conclusión	366
<b>Apéndice:</b>	<b>Proyectos semestrales</b>	<b>367</b>
	Proyectos semestrales sugeridos	371
	Referencias	375
	<b>Índices</b>	<b>377</b>
	Índice onomástico	379
	Índice analítico	