

Índice

INTRODUCCIÓN	9
CAPÍTULO 1. DESARROLLO DE SOFTWARE	11
1.1 EL PROGRAMA INFORMÁTICO	12
1.1.1 Interacción con el sistema.....	12
1.2 LENGUAJES DE PROGRAMACIÓN	14
1.2.1 Clasificación y características	14
1.3 OBTENCIÓN DE CÓDIGO EJECUTABLE.....	17
1.3.1 Tipos de código (fuente, objeto y ejecutable).....	17
1.3.2 Compilación.....	17
1.4 PROCESOS DE DESARROLLO	19
1.4.1 Análisis	19
1.4.2 Diseño	19
1.4.3 Codificación	20
1.4.4 Pruebas.....	20
1.4.5 Documentación.....	20
1.4.6 Explotación.....	21
1.4.7 Mantenimiento.....	21
1.5 ROLES QUE INTERACTÚAN EN EL DESARROLLO	21
1.6 ARQUITECTURA DE SOFTWARE.....	22
1.6.1 Patrones de desarrollo.....	22
1.6.2 Desarrollo en tres capas.....	38
RESUMEN DEL CAPÍTULO.....	41
EJERCICIOS PROPUESTOS.....	41
TEST DE CONOCIMIENTOS	44
CAPÍTULO 2. INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO	45
2.1 CARACTERÍSTICAS.....	46
2.1.1 Extensiones y herramientas.....	46
2.1.2 Personalización y configuración	48
2.2 CRITERIOS DE ELECCIÓN DE UN IDE.....	48
2.2.1 Sistema operativo.....	49
2.2.2 Lenguaje de programación y <i>framework</i>	49
2.2.3 Herramientas y disponibilidad.....	49
2.3 USO BÁSICO DE UN IDE.....	51
2.3.1 Edición de programas y generación de ejecutables.....	51
2.3.2 Desarrollo colaborativo	51
2.4 NUESTRA ELECCIÓN VISUAL STUDIO.....	53
2.4.1 Instalación.....	54

2.4.2	Recorrido por las ventanas y paletas principales	55
2.4.3	Personalización y configuración	59
	RESUMEN DEL CAPÍTULO.....	63
	TEST DE CONOCIMIENTOS	64
	CAPÍTULO 3. DEPURACIÓN Y REALIZACIÓN DE PRUEBAS	65
3.1	HERRAMIENTAS DE DEPURACIÓN	66
3.1.1	Puntos de ruptura	66
3.1.2	Puntos de seguimiento.....	67
3.1.3	Inspecciones	67
3.2	ANÁLISIS DE CÓDIGO	68
3.2.1	Analizador estático de código.....	68
3.3	CASOS DE PRUEBA	71
3.3.1	Caja blanca	72
3.3.2	Caja negra	73
3.3.3	Rendimiento	75
3.3.4	Coherencia.....	77
3.4	PRUEBAS UNITARIAS	78
3.4.1	Metodología	78
3.4.2	NUnit.....	79
	RESUMEN DEL CAPÍTULO.....	83
	EJERCICIOS PROPUESTOS.....	84
	TEST DE CONOCIMIENTOS	85
	CAPÍTULO 4. OPTIMIZACIÓN Y DOCUMENTACIÓN	87
4.1	REFACTORIZACIÓN.....	88
4.1.1	Tabulación	90
4.1.2	Patrones de refactorización más usuales	91
4.1.3	Malos olores	101
4.1.4	Refactorización y pruebas	102
4.1.5	Herramientas de Visual Studio	103
4.2	CONTROL DE VERSIONES.....	118
4.2.1	Repositorios	118
4.2.2	Herramientas de control de versiones.....	121
4.3	DOCUMENTACIÓN.....	124
4.3.1	Uso de comentarios	124
4.3.2	Herramientas	126
	RESUMEN DEL CAPÍTULO.....	128
	TEST DE CONOCIMIENTOS	129
	CAPÍTULO 5. DISEÑO ORIENTADO A OBJETOS. DIAGRAMAS DE CLASE	131
5.1	INTRODUCCIÓN A UML	132
5.2	DISEÑO DE CLASES EN UML.....	133
5.2.1	Clases, atributos y métodos	133
5.2.2	Relaciones.....	135

5.3	HERRAMIENTAS	139
5.3.1	Herramienta de modelado de VS	139
5.3.2	UMLPad	144
	RESUMEN DEL CAPÍTULO	146
	EJERCICIOS PROPUESTOS.....	147
	TEST DE CONOCIMIENTOS	147
	CAPÍTULO 6. DISEÑO ORIENTADO A OBJETOS. DIAGRAMAS DE COMPORTAMIENTO	149
6.1	TIPOS Y CAMPO DE APLICACIÓN	150
6.2	DIAGRAMAS DE ACTIVIDAD	150
6.3	DIAGRAMAS DE CASOS DE USO.....	153
6.4	DIAGRAMAS DE SECUENCIA	155
6.4.1	Ingeniería inversa	157
	RESUMEN DEL CAPÍTULO	163
	EJERCICIOS PROPUESTOS.....	164
	TEST DE CONOCIMIENTOS	165
	CAPÍTULO 7. ¡PONLO EN PRÁCTICA!	167
7.1	NUESTRO PROYECTO	168
7.2	PLANTEAMIENTO.....	168
7.2.1	Diseño conceptual	168
7.2.2	Modelado completo.....	168
7.3	¿QUÉ TIPO DE PROYECTO ES?.....	169
7.3.1	Tipos de proyecto.....	169
7.4	DOCUMENTACIÓN.....	170
7.5	OPCIONAL: INSTALACIÓN Y DISTRIBUCIÓN.....	170
7.6	NOTAS.....	170
7.7	PROYECTO PROPUESTO.....	171
	CAPÍTULO 8. COMENTARIOS Y CONCLUSIONES	175
	MATERIAL ADICIONAL.....	177
	ÍNDICE ALFABÉTICO	179