

Capítulo 1. Introducción a la informática.....	15
1.1 Estructura y funcionamiento de un ordenador.....	16
1.1.1 ¿Qué es un ordenador?.....	16
1.1.2 Arquitectura de Von Neumann.....	16
1.1.3 La memoria.....	17
1.1.4 La CPU.....	18
1.2 Sistemas Operativos.....	19
1.2.1 ¿Qué es el Sistema Operativo?.....	19
1.2.2 Un poco de historia.....	21
1.2.3 Funciones del Sistema Operativo.....	21
1.3 Creación de programas.....	22
1.3.1 Ingeniería del software.....	22
1.3.2 Lenguajes y paradigmas de programación.....	24
1.3.2.1 Lenguajes de programación.....	24
1.3.2.2 Paradigmas de programación.....	24
1.3.2.3 Programar en un lenguaje de alto nivel.....	25
1.3.2.4 Lenguaje C.....	26
1.3.3 Pasos para crear un programa.....	27
1.3.3.1 Algoritmo.....	27
1.3.3.2 Edición.....	27
1.3.3.3 Compilación.....	27
1.3.3.4 Vinculación o <i>Linking</i>	28
1.3.3.5 Ejecución.....	29
1.4 Conclusiones del capítulo.....	29
 Capítulo 2. Algoritmos.....	 31
2.1 ¿Qué es un algoritmo?.....	32
2.1.1 Definición de algoritmo.....	32
2.1.2 Requisitos de un algoritmo.....	32
2.1.3 Ejercicio lúdico.....	32
2.1.4 Los cuatro pasos para encontrar un algoritmo.....	33
2.2 Escribir un algoritmo.....	34
2.2.1 Análisis del problema.....	34
2.2.2 Abstracción.....	35
2.2.3 Diseño descendente.....	35
2.2.4 Pseudocódigo estructurado.....	37
2.2.4.1 Qué es el pseudocódigo.....	37
2.2.4.2 Composición secuencial.....	37
2.2.4.3 Composición iterativa.....	37
2.2.4.4 Composición selectiva.....	39
2.2.4.5 Entrada de datos y salida de resultados.....	41
2.2.5 Algoritmo completo.....	41

2.2.6 Gráfico del árbol de tarea	43
2.3 Del algoritmo al programa.....	44
2.3.1 Refinamientos sucesivos.....	44
2.3.2 Dos fases principales	45
2.3.3 Los pasos detallados	45
2.4 Ejercicios.....	46
2.4.1 Ejercicio 1: repartir caramelos.....	46
2.4.2 Ejercicio 2: calcular perímetro y superficie.....	51
2.4.2.1 Sin control del dato de entrada.....	51
2.4.2.2 Condicional para controlar los datos de entrada.....	52
2.4.2.3 Repetición para introducir datos de entrada válidos.....	52
2.4.2.4 Repeticiones acotadas.....	53
2.4.2.5 Repetición mientras el usuario desee.....	53
2.5 Conclusiones del capítulo.....	54
Capítulo 3. Componentes básicos de un lenguaje de alto nivel.....	55
3.1 Estructura general de un programa	57
3.2 Variables.....	57
3.2.1 Nombre de variable	58
3.2.2 Tipo de variable	58
3.2.3 Valor de variable.....	60
3.2.4 Declaración de variables.....	60
3.2.5 Declaración de constantes.....	61
3.2.6 Representación gráfica	61
3.2.7 El primer refinamiento: declaración de variables.....	62
3.3 Operaciones	64
3.3.1 Diferentes clases de operaciones.....	64
3.3.2 Operadores aritméticos.....	64
3.3.3 Operadores relacionales.....	65
3.3.4 Operadores lógicos.....	65
3.3.5 Precedencia de operadores.....	67
3.3.6 Condiciones	68
3.3.7 El segundo refinamiento: condiciones formales	69
3.4 Instrucción de asignación	70
3.4.1 Qué es una asignación	70
3.4.2 Cómo se ejecuta la asignación	71
3.4.3 Errores frecuentes.....	71
3.4.3.1 Variable usada sin inicializar	71
3.4.3.2 Incompatibilidad de tipos.....	72
3.4.4 Incremento o decremento	72
3.5 Instrucciones de entrada y salida	73
3.6 Control de instrucciones	75

3.6.1 Composición secuencial	8
3.6.2 Composición condicional	8
3.6.3 Composición iterativa o repetitiva	8
3.6.3.1 Instrucción «while»	8
3.6.3.2 Instrucción «for»	8
3.7 Ejercicios	8
3.7.1 Ejercicio 1: valor absoluto	8
3.7.2 Ejercicio 2: calcular perímetro y superficie	8
3.7.2.1 Sin control de la entrada	8
3.7.2.2 Condicional para controlar la entrada	8
3.7.2.3 Repeticiones acotadas para introducir la entrada válida	8
3.7.2.4 Repetición mientras el usuario desee	8
3.7.3 Ejercicio 3: suma de números	8
3.7.3.1 Secuencia finalizada con número especial	8
3.7.3.2 Secuencia finalizada con respuesta	9
3.7.3.3 Secuencia con cantidad de elementos conocida	9
3.7.4 Ejercicio 4: potencia	9
3.7.4.1 Análisis del problema	9
3.7.4.2 Versión del algoritmo con errores	9
3.7.4.3 Versión del algoritmo corregido	9
3.7.4.4 Versión del algoritmo corregido y generalizado	9
3.7.4.5 Implementación completa	10
3.8 Conclusiones del capítulo	10
 Capítulo 4. Esquemas algorítmicos	 105
4.1 Series matemáticas	107
4.1.1 Sumatorias y productorias	107
4.1.2 Ejercicio 1: factorial	110
4.1.2.1 Análisis del problema	110
4.1.2.2 Algoritmo de cálculo de factorial	110
4.1.2.3 Implementación completa	111
4.1.3 Ejercicio 2: series de Fourier	111
4.2 Tratamientos de secuencias	113
4.3 Recorrido de una secuencia	115
4.3.1 Esquema algorítmico de recorrido	115
4.3.2 Ejercicio de recorrido	116
4.3.3 Esquema de recorrido modificado	118
4.4 Búsqueda de un elemento en una secuencia	118
4.4.1 Esquemas algorítmicos de búsqueda	118
4.4.2 Ejercicio de búsqueda	120
4.4.3 Esquema de búsqueda modificado	124
4.4.4 Igualdad entre elementos complejos	125

4.5 Comparación de dos secuencias.....	126
4.5.1 Esquemas algorítmicos de comparación.....	126
4.5.2 Esquema de comparación de secuencias modificado.....	128
4.5.3 Relación « \Leftrightarrow » entre secuencias.....	129
4.6 Ordenación de una secuencia.....	129
4.6.1 Esquema algorítmico de ordenación por selección.....	129
4.6.2 Esquema algorítmico de ordenación por inserción.....	131
4.6.3 Independencia del criterio de ordenación.....	133
4.7 Conclusiones del capítulo.....	135
Capítulo 5. Programación modular: funciones.....	137
5.1 Conceptos.....	139
5.1.1 Diseño descendente.....	139
5.1.2 Programación modular.....	139
5.1.3 Invocaciones.....	140
5.1.4 Cajas negras.....	140
5.1.5 Diagrama de estructura.....	141
5.1.6 Metáfora de jefes y empleados.....	142
5.1.7 Funciones y procedimientos.....	143
5.2 Invocación de subprogramas de librerías.....	144
5.2.1 ¿Qué son librerías?.....	144
5.2.2 Ficheros de encabezados «h».....	144
5.2.3 Invocación de procedimientos.....	145
5.2.4 Invocación de funciones.....	146
5.3 Ventajas de la programación modular.....	146
5.3.1 Legibilidad.....	146
5.3.2 Reusabilidad.....	147
5.4 El método de los 3 pasos para hacer una función.....	148
5.4.1 Descripción de los tres pasos.....	148
5.4.2 Definición del encabezado.....	149
5.4.3 Invocación.....	150
5.4.4 Codificación del cuerpo.....	151
5.4.5 La implementación completa.....	154
5.4.6 El por qué de los tres pasos.....	154
5.5 Comunicación entre módulos.....	155
5.5.1 Variables locales y globales.....	155
5.5.1.1 ¿Qué son las variables locales?.....	155
5.5.1.2 ¿Qué sucede al ejecutarse un subprograma?.....	156
5.5.1.3 ¿Qué son las variables globales?.....	157
5.5.1.4 Consejos.....	158
5.5.2 Parámetros.....	158
5.5.2.1 ¿Qué son los parámetros?.....	158

5.5.2.2	Parámetros formales en la definición	159
5.5.2.3	Parámetros reales en la invocación.....	160
5.5.2.4	¿Qué sucede al ejecutarse un subprograma?	161
5.5.3	Paso de parámetros	162
5.5.3.1	Paso de parámetros en C	162
5.5.3.2	Paso de parámetros por valor	163
5.5.3.3	Paso de parámetros por referencia.....	164
5.5.4	Devolución de resultados.....	166
5.5.4.1	Un resultado: « <i>return</i> » de una función	166
5.5.4.2	Procedimientos	169
5.5.4.3	Más de un resultado: parámetros por referencia.....	170
5.6	Del algoritmo al programa modular.....	174
5.6.1	Resumen de los pasos	174
5.6.2	Encabezado	175
5.6.3	Invocación	176
5.6.4	Cuerpo	176
5.6.5	Programación ascendente	177
5.6.6	Organización en ficheros	178
5.7	Ejercicios	180
5.7.1	Ejercicio 1: Seno	180
5.7.2	Ejercicio 2: Ecuación de segundo grado.....	183
5.7.3	Ejercicio 3: Cálculo de perímetro y superficie.....	187
5.8	Conclusiones del capítulo.....	191
Capítulo 6.	Estructuras de datos: « <i>arrays</i> » y « <i>structs</i> ».....	193
6.1	<i>Arrays</i>	195
6.1.1	Almacenar secuencias en memoria	195
6.1.2	<i>Arrays</i> unidimensionales	195
6.1.2.1	Definición de un <i>array</i> en C.....	195
6.1.2.2	Uso de una constante para la longitud	196
6.1.2.3	Error de compilación si el largo no es constante.....	197
6.1.2.4	Acceso a cada elemento	197
6.1.2.5	Error en tiempo de ejecución	199
6.1.2.6	¿Elemento final o longitud efectiva?	200
6.1.3	<i>Arrays</i> multidimensionales	201
6.1.3.1	Definición de un <i>array</i> multidimensional en C.....	201
6.1.3.2	Acceso a cada elemento.....	202
6.1.4	<i>Strings</i>	204
6.1.5	Paso de parámetros de tipo <i>array</i> en C.....	205
6.1.6	Definición de nuevos tipos basados en <i>arrays</i>	207
6.1.7	Funciones para manipular el nuevo tipo	208
6.1.8	Tratamiento de secuencias en memoria.....	210

6.1.8.1 Almacenamiento de la secuencia.....	211
6.1.8.2 Recorrido	214
6.1.8.3 Búsqueda.....	216
6.1.8.4 Comparación	217
6.1.8.5 Ordenación.....	220
6.2 Registros o <i>Structs</i>	223
6.2.1 Almacenar datos de entidades complejas	223
6.2.2 Definición de nuevos tipos basados en <i>structs</i> en C.....	224
6.2.3 Acceso a cada componente.....	225
6.2.4 Funciones para manipular el nuevo tipo.....	225
6.2.5 Registros con campos complejos	228
6.2.6 Secuencias de elementos complejos.....	231
6.2.7 Tratamiento de secuencias de elementos complejos	235
6.3 Conclusiones del capítulo.....	237
Capítulo 7. Almacenamiento permanente: ficheros.....	239
7.1 Conceptos básicos de ficheros	240
7.1.1 ¿Qué es un fichero?	240
7.1.2 Nombre de fichero.....	240
7.1.3 Operaciones con ficheros.....	240
7.1.4 Dos maneras de guardar los datos: modo texto o binario	241
7.2 Leer o escribir ficheros desde un programa.....	242
7.2.1 Los 5 pasos para trabajar con ficheros.....	242
7.2.2 Incluir la librería.....	242
7.2.3 Variable para identificar al fichero.....	243
7.2.4 Abrir el fichero	243
7.2.5 Lectura del fichero	245
7.2.5.1 Lectura de fichero de texto.....	245
7.2.5.2 Lectura de fichero binario.....	246
7.2.6 Escritura del fichero	248
7.2.6.1 Escritura de fichero de texto	248
7.2.6.2 Escritura de fichero binario.....	248
7.2.7 Cerrar el fichero	249
7.2.8 Ejemplo completo: guardar y recuperar un entero	250
7.3 Otras funciones útiles	252
7.3.1 Fin de fichero	252
7.3.2 Mover el cabezal	253
7.3.3 Dónde está el cabezal	253
7.4 Almacenar secuencias en ficheros	254
7.4.1 Recordatorio	254
7.4.2 Tres opciones para saber el final de la secuencia	254
7.4.3 Ejercicio 1: secuencia de elementos simples	255

7.4.4 Ejercicio 2: secuencia de registros	260
7.5 Conclusiones del capítulo.....	271
Bibliografía	273