

Contenido

Modulo 1

Programación estructurada

1. Introducción a los algoritmos y a la programación de computadoras	1	1.7.7 Las constantes.....	21
1.1 Introducción.....	2	1.7.7.1 La directiva de preprocesador #define	21
1.2 Concepto de algoritmo	2	1.7.7.2 El modificador const.....	22
1.2.1 Definición de algoritmo y problema	2	1.7.8 Nomenclatura para las constantes	22
1.2.2 Análisis del enunciado de un problema	3	1.8 Operadores aritméticos	22
1.2.2.1 Análisis del problema	3	1.8.1 Conversión de tipos de datos (type casting)	25
1.2.2.2 Datos de entrada	3	1.8.2 El operador % (“módulo” o “resto”)	25
1.2.2.3 Datos de salida	4	1.8.3 Operadores relacionales	28
1.2.3 Memoria y operaciones aritméticas y lógicas	4	1.9 Expresiones lógicas	29
1.2.4 Teorema de la programación estructurada.....	4	1.9.1 Operadores lógicos.....	29
1.3 Conceptos de programación	5	1.10 Operadores de bits	30
1.3.1 Lenguajes de programación	5	1.10.1 Representación binaria de los tipos enteros.....	30
1.3.2 Codificación de un algoritmo	6	1.10.2 Operadores de desplazamiento de bits (>> y <<).....	31
1.3.3 Bibliotecas de funciones	6	1.10.3 Representación hexadecimal	31
1.3.4 Programas de computación	6	Representación hexadecimal de números enteros negativos.....	32
1.3.5 Consola	7	1.10.4 Representación octal	32
1.3.6 Entrada y salida de datos	7	1.10.5 Operadores lógicos de bits.....	32
1.3.7 Lenguajes algorítmicos	7	1.11 Resumen	33
1.3.8 Pseudocódigo	7	1.12 Contenido de la página Web de apoyo	33
1.4 Representación gráfica de algoritmos	7	1.12.1 Mapa conceptual	33
1.4.1 Representación gráfica de la estructura secuencial o acción simple	8	1.12.2 Autoevaluaciones	33
1.4.2 Representación gráfica de la estructura de decisión.....	8	1.12.3 Videotutorial	33
1.4.3 Representación gráfica de la estructura de repetición ..	8	1.12.3.1 Instalación y uso de Eclipse para C.....	33
1.4.4 Representación gráfica de módulos o funciones	9	1.12.4 Presentaciones*	33
1.5 Nuestro primer programa	11	2. Estructuras básicas de control y lógica algorítmica	35
1.5.1 Codificación del algoritmo utilizando el lenguaje C	11	2.1 Introducción	36
1.5.2 El archivo de código fuente.....	12	2.2 Estructura secuencial.....	36
1.5.3 Comentarios en el código fuente.....	12	2.3 Estructura de decisión	36
1.5.4 La compilación y el programa ejecutable	13	2.3.1 Estructuras de decisión anidadas.....	38
1.5.5 El entorno integrado de desarrollo (IDE)	13	2.3.2 Selección en línea o if-inline	41
1.6 La memoria de la computadora	14	2.3.3 Macros	42
1.6.1 El byte.....	15	2.3.4 Selección múltiple (switch)	43
1.6.2 Conversión numérica: de base 2 a base 10.....	15	2.3.5 Asignación de valores alfanuméricos (función strcpy).....	45
1.6.3 Dimensionamiento de los datos.....	15	2.4 Estructura de repetición	47
1.6.4 Los números negativos	16	2.4.1 Estructuras de repetición inexactas.....	47
1.6.5 Los caracteres.....	16	2.4.2 Estructuras de repetición exactas.....	49
1.7 Las variables	17	2.4.3 Contadores.....	51
1.7.1 Convención de nomenclatura para variables	17	2.4.4 Acumuladores	52
1.7.2 Los tipos de datos	18	2.4.5 Seguimiento del algoritmo y “prueba de escritorio”	53
1.7.3 Los tipos de datos provistos por el lenguaje C	18	2.4.6 El debugger, la herramienta de depuración.....	54
1.7.3.1 Notación húngara.....	19	2.4.7 Estructuras de repetición anidadas	57
1.7.4 La función de biblioteca printf.....	19	2.4.8 Manejo de valores booleanos	59
1.7.5 La función de biblioteca scanf	20	2.4.9 Máximos y mínimos	60
1.7.6 El operador de dirección &.....	21	2.5 Contextualización del problema	63
		2.6 Resumen	69

2.7	Contenido de la página Web de apoyo	69	4.2.2.2	Determinar si un carácter es una letra (función esLetra)	101
2.7.1	Mapa conceptual	69	4.2.2.3	Determinar si un carácter es una letra mayúscula o minúscula (funciones esMayuscula y esMinuscula)	102
2.7.2	Autoevaluaciones	69	4.2.2.4	Convertir un carácter a minúscula (función aMinuscula)	102
2.7.3	Videotutorial	69	4.2.2.5	Convertir un carácter a mayúscula (función aMayuscula)	102
2.7.3.1	Uso del debugger para depurar un programa	69			
2.7.4	Presentaciones*	69			
3.	Funciones, modularización y metodología top-down ...	71	4.3	Cadenas de caracteres	103
3.1	Introducción	72	4.3.1	El carácter '\0' (barra cero)	104
3.2	Conceptos iniciales	72	4.3.2	Longitud de una cadena	105
3.2.1	Metodología top-down	72	4.3.2.1	La cadena vacía	105
3.2.2	Módulos o subprogramas	72	4.4	Tratamiento de cadenas de caracteres	106
3.2.3	Funciones	72	4.4.1	Inicialización de una cadena de caracteres	106
3.2.4	Funciones de biblioteca	73	4.4.2	Funciones para el tratamiento de cadenas de caracteres	107
3.2.5	Invocación a funciones de biblioteca	73	4.4.2.1	Asignar o copiar una cadena a un char[] (función copiarCadena)	107
3.3	Funciones definidas por el programador	73	4.4.2.2	Determinar la longitud de una cadena (función longitud)	108
3.3.1	Prototipo de una función	74	4.4.2.3	Determinar si una cadena es "vacía" (función esVacía)	109
3.3.2	Invocar a una función	74	4.4.2.4	Concatenar cadenas (función concatenarCadena)	110
3.3.3	Desarrollo de una función	75	4.4.2.5	Comparar cadenas (función compararCadenas)	111
3.3.4	Convención de nomenclatura para funciones	76	4.4.2.6	Convertir cadenas a números enteros (función cadenaAEntero)	113
3.3.5	Funciones que no retornan ningún valor (tipo de datos void)	76	4.5	Funciones de biblioteca para manejo de cadenas	115
3.3.6	Archivos de cabecera (.h)	76	4.5.1	Otras funciones de biblioteca	116
3.3.7	Archivos de funciones (.c)	76	4.5.1.1	Dar formato a una cadena (función sprintf)	116
3.4	Legibilidad y reusabilidad del código	77	4.5.1.2	Interpretar (parsear) el formato de una cadena (función sscanf)	116
3.4.1	Abstracción	78	4.6	Resumen	117
3.4.2	Argumentos y parámetros	80	4.7	Contenido de la página Web de apoyo	117
3.5	Alcance de las variables (scope)	82	4.7.1	Mapa conceptual	117
3.5.1	Variables locales	82	4.7.2	Autoevaluaciones	117
3.5.2	Variables globales	83	4.7.3	Presentaciones*	117
3.6	Argumentos por valor y referencia	84	5.	Punteros a carácter	119
3.6.1	Punteros y direcciones de memoria	85	5.1	Introducción	120
3.6.2	El operador de indirección * (asterisco)	85	5.2	Conceptos iniciales	120
3.6.3	Argumentos por referencia	86	5.2.1	Aritmética de direcciones	121
3.6.4	Funciones que mantienen su estado	91	5.2.2	Prefijos y sufijos	122
3.6.5	Variables estáticas (modificador static)	94	5.2.2.1	Determinar si una cadena es prefijo de otra (función esPrefijo)	122
3.7	Resumen	96	5.2.2.2	Determinar si una cadena es sufijo de otra (función esSufijo)	123
3.8	Contenido de la página Web de apoyo	97	5.3	Funciones que retornan cadenas	124
3.8.1	Mapa conceptual	97	5.3.1	La función malloc	125
3.8.2	Autoevaluaciones	97	5.3.2	Subcadenas (función substring)	126
3.8.3	Videotutorial	97	5.3.2.1	Eliminar los espacios ubicados a la izquierda (función ltrim)	129
3.8.3.1	Mantener archivos de funciones separados del programa principal	97	5.3.2.2	Eliminar los espacios ubicados a la derecha (función rtrim)	130
3.8.4	Presentaciones*	97			
4.	Tipos de datos alfanuméricos	99			
4.1	Introducción	100			
4.2	Carácter	100			
4.2.1	El tipo de datos char	100			
4.2.2	Funciones para tratamiento de caracteres	101			
4.2.2.1	Determinar si un carácter es un dígito numérico (función esDigito)	101			

5.3.2.3	Eliminar los espacios en ambos extremos de la cadena (función trim)	131	7.4.2	Agregar un elemento al array	165
5.3.3	Función de biblioteca strtok	132	7.4.3	Búsqueda secuencial	166
5.4	Resumen	134	7.4.4	Buscar y agregar	168
5.5	Contenido de la página Web de apoyo	134	7.4.5	Insertar un elemento	172
5.5.1	Mapa conceptual	134	7.4.6	Eliminar un elemento	174
5.5.2	Autoevaluaciones	134	7.4.7	Insertar en orden	176
5.5.3	Presentaciones*	134	7.4.8	Buscar en orden	178
6.	Punteros, arrays y aritmética de direcciones	135	7.4.9	Buscar e insertar en orden	179
6.1	Introducción	136	7.4.10	Ordenar arrays (algoritmo de la "burbuja")	180
6.2	Punteros y direcciones de memoria	136	7.4.11	Búsqueda binaria o dicotómica	183
6.2.1	El operador de dirección &	136	7.4.11.1	Implementación	183
6.2.2	Los punteros	137	7.5	Arrays multidimensionales	188
6.2.3	El operador de indirección *	137	7.5.1	Arrays bidimensionales (matrices)	189
6.2.4	Funciones que reciben punteros	137	7.5.1.1	Recorrer una matriz por fila/columna	190
6.3	Arrays	139	7.5.1.2	Recorrer una matriz por columna/fila	190
6.3.1	La capacidad del array	139	7.5.2	Arrays tridimensionales (cubos)	192
6.3.2	Acceso a los elementos de un array	139	7.6	Tipos de datos definidos por el programador	192
6.3.3	Dimensionamiento e inicialización de arrays	141	7.6.1	Introducción al encapsulamiento a través de TADs	193
6.3.4	Crear arrays dinámicamente (funciones malloc y sizeof)	142	7.6.2	Estructuras o registros	195
6.3.5	Punteros genéricos void*	142	7.6.3	Representación gráfica de una estructura	196
6.4	Relación entre arrays y punteros	143	7.6.4	Estructuras anidadas	196
6.4.1	Aritmética de direcciones	143	7.6.5	Estructuras con campos de tipo array	197
6.5	Código compacto y eficiente	144	7.6.6	Punteros a estructuras	197
6.5.1	Operadores de incremento y decremento (operadores unarios)	144	7.6.6.1	El operador "flecha" ->	198
6.5.2	"Pre" y "post" incremento y decremento	144	7.6.7	Arrays de estructuras	198
6.5.3	Operadores de asignación	145	7.6.8	Estructuras con campos de tipo "array de estructuras"	199
6.5.4	Incremento de punteros	146	7.7	Resumen	199
6.5.4.1	Implementación compacta de la función copiarCadena	146	7.8	Contenido de la página Web de apoyo	200
6.5.4.2	Implementación compacta de la función longitud	147	7.8.1	Mapa conceptual	200
6.6	Arrays de cadenas	148	7.8.2	Autoevaluaciones	200
6.6.1	Argumentos en línea de comandos (int argc, char* argv[])	151	7.8.3	Videotutoriales	200
6.7	Resumen	153	7.8.3.1	Algoritmo de la burbuja	200
6.8	Contenido de la página Web de apoyo	153	7.8.3.2	Algoritmo de la búsqueda binaria	200
6.8.1	Mapa conceptual	153	7.8.4	Presentaciones*	200
6.8.2	Autoevaluaciones	153	8.	Operaciones sobre archivos	201
6.8.3	Videotutorial	153	8.1	Introducción	202
6.8.3.1	Pasar argumentos en línea de comandos con Eclipse	153	8.1.1	Memoria principal o memoria RAM de la computadora	202
6.8.4	Presentaciones*	153	8.1.2	Medios de almacenamiento (memoria secundaria)	202
7.	Tipos de datos estructurados	155	8.2	Archivos	202
7.1	Introducción	156	8.2.1	Abrir un archivo	203
7.2	Acceso directo sobre arrays	156	8.2.2	Escribir datos en un archivo	203
7.3	Acceso indirecto sobre arrays	163	8.2.3	Leer datos desde un archivo	204
7.4	Operaciones sobre arrays	164	8.2.4	El identificador de posición (puntero)	205
7.4.1	Capacidad vs. longitud de un array	164	8.2.5	Representación gráfica	206
			8.2.6	Valor actual del identificador de posición (función ftell)	206
			8.2.7	Manipular el valor del identificador de posición (función fseek)	207
			8.2.8	Calcular el tamaño de un archivo	208
			8.2.9	Archivos de texto vs. archivos binarios	209

8.3	Archivos de registros	209	10.3	Apareo de archivos	256
8.3.1	Archivos de estructuras	210	10.3.1	Apareo de archivos con corte de control	261
	8.3.1.1 Grabar estructuras (registros) en un archivo	210	10.4	Resumen	266
	8.3.1.2 Leer estructuras (registros) desde un archivo	211	10.5	Contenido de la página Web de apoyo	266
	8.3.1.3 Legibilidad del código fuente.....	212	10.5.1	Mapa conceptual	266
8.3.2	Acceso directo a registros	213	10.5.2	Autoevaluaciones	266
	8.3.2.1 Acceso directo para lectura.....	214	10.5.3	Presentaciones*	266
	8.3.2.2 Acceso directo para escritura.....	215	11.	Estructuras de datos dinámicas lineales	267
	8.3.2.3 Agregar un registro al final del archivo.....	216	11.1	Introducción	268
8.3.3	Calcular la cantidad de registros que tiene un archivo	217	11.2	Estructuras estáticas	269
8.4	Lectura y escritura en bloques (buffers).....	217	11.3	Estructuras dinámicas	269
8.5	Archivos de texto	219	11.3.1	El nodo	269
8.5.1	Apertura de un archivo de texto	220	11.4	Listas enlazadas	269
8.5.2	Leer y escribir caracteres (funciones getc y putc)	220	11.4.1	Estructuras de datos dinámicas lineales.....	270
8.5.3	Escribir líneas (función fprintf)	221	11.4.2	Estructuras de datos dinámicas no lineales	270
8.5.4	Leer líneas (función fgets).....	221	11.4.3	Punteros por referencia.....	270
8.5.5	Leer datos formateados (función fscanf)	222	11.5	Operaciones sobre listas enlazadas	271
8.6	Operaciones lógicas sobre archivos	223	11.5.1	Agregar un elemento nuevo al final de una lista	272
8.6.1	Limitaciones de los archivos secuenciales	223	11.5.2	Recorrer una lista para mostrar su contenido	276
8.6.2	Ordenamiento de archivos en memoria.....	224	11.5.3	Liberar la memoria que utilizan los nodos de una lista enlazada.....	276
8.6.3	Relación entre el número de byte y el número de registro	227	11.5.4	Determinar si la lista contiene un valor determinado	278
8.6.4	Búsqueda binaria sobre archivos	227	11.5.5	Eliminar un elemento de la lista.....	280
8.6.5	Indexación	228	11.5.6	Insertar un valor respetando el ordenamiento de la lista	283
8.6.6	Indexación de archivos	229	11.5.7	Insertar un valor solo si la lista aún no lo contiene	285
8.6.7	Eliminar registros en un archivo (bajas lógicas)	233	11.6	Estructura Pila (LIFO)	286
8.6.8	Bajas lógicas con soporte en un archivo auxiliar	235	11.6.1	Implementación de la estructura pila	286
8.7	Resumen	236	11.6.2	Operaciones poner (push) y sacar (pop).....	286
8.8	Contenido de la página Web de apoyo	236	11.6.3	Determinar si la pila tiene elementos o no	289
8.8.1	Mapa conceptual	236	11.7	Estructura Cola (FIFO).....	289
8.8.2	Autoevaluaciones	236	11.7.1	Lista enlazada circular.....	290
8.8.3	Videotutoriales.....	236	11.7.2	Implementar una cola sobre una lista circular	290
	8.8.3.1 Leer y escribir un archivo.....	236	11.7.3	Operaciones encolar y desencolar	292
	8.8.3.2 Leer y escribir un archivo de registros	236	11.8	Lista doblemente enlazada	294
8.8.4	Presentaciones*	236	11.9	Nodos que contienen múltiples datos.....	295
9.	Tipo Abstracto de Dato (TAD)	237	11.9.1	Nodo con múltiples campos.....	295
9.1	Introducción		11.9.2	Nodo con un único valor de tipo struct	296
9.2	Capas de abstracción		11.10	Estructuras de datos combinadas	297
9.3	Tipos de datos		11.10.1	Lista y sublista.....	298
9.4	Resumen		11.10.2	Arrays de colas.....	300
9.5	Contenido de la página Web de apoyo		11.10.3	Matriz de pilas	307
10.	Análisis de ejercicios integradores	239	11.11	Resumen	311
10.1	Introducción.....	240	11.12	Contenido de la página Web de apoyo	311
10.2	Problemas con corte de control.....	240	11.12.1	Mapa conceptual	311
10.2.1	Archivos de novedades vs. archivos maestros.....	246	11.12.2	Autoevaluaciones	311
10.2.2	Uso de arrays auxiliares	249	11.12.3	Presentaciones*	311
10.2.3	Mantener archivos (pequeños) en memoria.....	250			



Modulo 2

Programación orientada a objetos

12. Encapsulamiento a través de clases y objetos	313
12.1 Introducción.....	314
12.2 Clases y objetos.....	314
12.2.1 Las clases.....	314
12.2.2 Miembros de la clase.....	315
12.2.3 Interfaz y encapsulamiento.....	316
12.2.4 Estructura de una clase.....	317
12.2.5 El constructor y el destructor.....	318
12.2.6 Los métodos.....	319
12.2.7 Los objetos.....	319
12.2.8 Instanciar objetos.....	320
12.2.9 Operadores new, delete y punteros a objetos.....	320
12.2.10 Sobrecarga de métodos.....	320
12.3 Encapsulamiento de estructuras lineales.....	321
12.3.1 Análisis de la clase Pila.....	321
12.3.2 Templates y generalizaciones.....	323
12.4 El lenguaje de programación Java.....	326
12.4.1 El programa principal en Java.....	328
12.4.2 Templates en C++, generics en Java.....	328
12.4.3 Los wrappers (envoltorios) de los tipos de datos primitivos.....	330
12.4.4 Autoboxing.....	330
12.5 Resumen.....	331
12.6 Contenido de la página Web de apoyo.....	331
12.6.1 Mapa conceptual.....	331
12.6.2 Autoevaluaciones.....	331
12.6.3 Presentaciones*.....	331
13. Introducción al lenguaje de programación Java	333
13.1 Introducción.....	334
13.2 Comencemos a programar.....	334
13.2.1 El Entorno Integrado de Desarrollo (IDE).....	335
13.2.2 Entrada y salida estándar.....	335
13.2.3 Comentarios en el código fuente.....	336
13.3 Tipos de datos, operadores y estructuras de control.....	337
13.3.1 El bit de signo para los tipos de datos enteros.....	337
13.3.2 El compilador y la máquina virtual (JVM o JRE).....	337
13.3.3 Estructuras de decisión.....	337
13.3.4 Estructuras iterativas.....	340
13.3.5 El tipo de datos boolean y las expresiones lógicas.....	342
13.3.6 Las constantes.....	343
13.3.7 Arrays.....	344
13.3.8 Matrices.....	346
13.3.9 Literales de cadenas de caracteres.....	348
13.3.10 Caracteres especiales.....	350
13.3.11 Argumentos en línea de comandos.....	351
13.4 Tratamiento de cadenas de caracteres.....	351
13.4.1 Acceso a los caracteres de un string.....	351
13.4.2 Mayúsculas y minúsculas.....	352
13.4.3 Ocurrencias de caracteres.....	353
13.4.4 Subcadenas.....	353
13.4.5 Prefijos y sufijos.....	354
13.4.6 Posición de un substring dentro de la cadena.....	354
13.4.7 Conversión entre números y cadenas.....	355
13.4.8 Representación en diferentes bases numéricas.....	355
13.4.9 La clase StringTokenizer.....	356
13.4.10 Comparación de cadenas.....	357
13.5 Resumen	359
13.6 Contenido de la página Web de apoyo	359
13.6.1 Mapa conceptual.....	359
13.6.2 Autoevaluaciones.....	359
13.6.3 Videotutorial.....	359
13.6.3.1 Instalar y utilizar Eclipse para Java.....	359
13.6.4 Presentaciones*.....	359
14. Programación orientada a objetos	361
14.1 Introducción.....	362
14.2 Clases y objetos.....	362
14.2.1 Los métodos.....	363
14.2.2 Herencia y sobrescritura de métodos.....	365
14.2.3 El método toString.....	365
14.2.4 El método equals.....	366
14.2.5 Declarar y "crear" objetos.....	367
14.2.6 El constructor.....	368
14.2.7 Repaso de lo visto hasta aquí.....	369
14.2.8 Convenciones de nomenclatura.....	370
14.2.8.1 Los nombres de las clases.....	370
14.2.8.2 Los nombres de los métodos.....	370
14.2.8.3 Los nombres de los atributos.....	371
14.2.8.4 Los nombres de las variables de instancia.....	371
14.2.8.5 Los nombres de las constantes.....	371
14.2.9 Sobrecarga de métodos.....	371
14.2.10 Encapsulamiento.....	374
14.2.11 Visibilidad de los métodos y los atributos.....	376
14.2.12 Packages (paquetes).....	377
14.2.13 Estructura de paquetes y la variable CLASSPATH.....	377
14.2.14 Las APIs (Application Programming Interface).....	378
14.2.15 Representación gráfica UML.....	379
14.2.16 Importar clases de otros paquetes.....	380
14.3 Herencia y polimorfismo.....	380
14.3.1 Polimorfismo.....	383
14.3.2 Constructores de subclases.....	385
14.3.3 La referencia super.....	385
14.3.4 La referencia this.....	388
14.3.5 Clases abstractas.....	389
14.3.6 Constructores de clases abstractas.....	392
14.3.7 Instancias.....	395
14.3.8 Variables de instancia.....	396
14.3.9 Variables de la clase.....	399
14.3.10 El garbage collector (recolector de residuos).....	399
14.3.11 El método finalize.....	399

14.3.12	Constantes	400
14.3.13	Métodos de la clase	400
14.3.14	Clases utilitarias.....	403
14.3.15	Referencias estáticas	403
14.3.16	Colecciones (primera parte)	404
14.3.17	Clases genéricas	409
14.4	Interfaces	412
14.4.1	Desacoplamiento de clases	413
14.4.2	El patrón de diseño de la factoría de objetos	415
14.4.3	Abstracción a través de interfaces.....	416
14.4.4	La interface Comparable.....	416
14.4.5	Desacoplar aún más	420
14.4.6	La interface Comparator	423
14.5	Colecciones de objetos	424
14.5.1	Cambio de implementación	426
14.5.2	El método Collections.sort	427
14.6	Excepciones	430
14.6.1	Errores lógicos vs. errores físicos	433
14.6.2	Excepciones declarativas y no declarativas	433
14.6.3	El bloque try-catch-finally	435
14.6.4	El método printStackTrace.....	438
14.7	Resumen	438
14.8	Contenido de la página Web de apoyo	438
14.8.1	Mapa conceptual	438
14.8.2	Autoevaluaciones	438
14.8.3	Videotutorial	438
14.8.3.1	Uso del javadoc.....	438
14.8.4	Presentaciones*	438
15.	Estructuras de datos dinámicas lineales en Java	439
15.1	Introducción.....	440
15.2	Listas (implementaciones de List).....	440
15.2.1	La clase ArrayList.....	440
15.2.2	La clase LinkedList.....	442
15.2.3	Comparación entre ArrayList y LinkedList	443
15.2.4	Desarrollo de la clase Performance	443
15.2.5	Introducción al análisis de complejidad algorítmica..	444
15.2.5.1	Acceso aleatorio a los elementos de la colección.....	445
15.2.5.2	Eliminar un elemento de la colección	446
15.2.5.3	Insertar un elemento en la colección.....	448
15.2.6	Pilas y colas.....	448
15.3	Mapas (implementaciones de Map).....	449
15.3.1	Tablas de dispersión (Hashtable)	449
15.3.2	Iterar una hashtable	451
15.3.3	Iterar una hashtable respetando el orden en que se agregaron los datos	452
15.4	Estructuras de datos combinadas.....	454
15.4.1	Ejemplo de una situación real	455
15.5	Resumen	459
15.6	Contenido de la página Web de apoyo	459
15.6.1	Mapa conceptual	459
15.6.2	Autoevaluaciones	459
15.6.3	Presentaciones*	459

Modulo 3

Aplicación práctica



16.	Compresión de archivos mediante el algoritmo de Huffman.....	461
16.1	Introducción	
16.2	El algoritmo de Huffman	
16.3	Aplicación práctica	
16.4	Análisis de clases y objetos	
16.5	Interfaces e implementaciones	
16.6	Manejo de archivos en Java	
16.7	Clases utilitarias	
16.8	Resumen	
16.9	Contenido de la página Web de apoyo	



Modulo 4

Conceptos avanzados

17.	Recursividad.....	463
17.1	Introducción.....	464
17.2	Conceptos iniciales.....	464
17.2.1	Funciones recursivas.....	464
17.2.2	Finalización de la recursión	465
17.2.3	Invocación a funciones recursivas	465
17.2.4	Funcionamiento de la pila de llamadas (stack)	467
17.2.5	Funciones recursivas vs. funciones iterativas	470
17.3	Otros ejemplos de recursividad	471
17.4	Permutar los caracteres de una cadena	472
17.5	Búsqueda binaria	476
17.6	Ordenamiento por selección	478
17.7	La función de Fibonacci	480
17.7.1	Optimización del algoritmo recursivo de Fibonacci... ..	485
17.8	Resumen	486
17.9	Contenido de la página Web de apoyo	486
17.9.1	Mapa conceptual	486
17.9.2	Autoevaluaciones	486
17.9.3	Presentaciones*	486
18.	Árboles.....	487
18.1	Introducción.....	488
18.1.1	Tipos de árbol	488
18.1.2	Implementación de la estructura de datos	488
18.2	Árbol binario.....	489
18.2.1	Niveles de un árbol binario.....	490
18.2.2	Recorrer los nodos de un árbol binario	490
18.2.3	Recorrido en amplitud o "por niveles"	490
18.2.4	Recorridos en profundidad (preorden, postorden e inorden).....	490
18.2.5	Implementación iterativa del recorrido en preorden ..	492
18.2.6	Implementación iterativa del recorrido en postorden	493

18.3	Árbol binario en Java, objetos.....	495	20.6	Heapsort (ordenamiento por montículos)	538
18.3.1	Enfoque basado en una clase utilitaria	496	20.6.1	Árbol binario semicompleto	538
18.3.2	Recorrido por niveles o "en amplitud"	497	20.6.2	Representar un árbol binario semicompleto en un array. 538	
18.3.3	Recorridos preorden, postorden e inorden.....	500	20.6.3	Montículo (heap).....	539
18.3.4	Recorrido iterativo	500	20.6.4	Transformar un árbol binario semicompleto en un montículo.....	539
18.3.5	Iteradores	501	20.6.5	El algoritmo de ordenamiento por montículos.....	541
18.3.6	Iteradores vs. callback methods	503	20.7	Shellsort (ordenamiento Shell)	544
18.3.7	Enfoque basado en objetos.....	504	20.8	Binsort (ordenamiento por cajas).....	545
18.4	Árbol Binario de Búsqueda	504	20.9	Radix sort (ordenamiento de raíz).....	546
18.4.1	Crear un Árbol Binario de Búsqueda (ABB).....	504	20.9.1	Ordenar cadenas de caracteres con radix sort	547
18.4.2	Encapsulamiento de la lógica y la estructura de datos (clase Abb).....	507	20.10	Resumen	548
18.4.3	Agregar un elemento al ABB (método agregar).....	508	20.11	Contenido de la página Web de apoyo.....	548
18.4.4	Ordenar valores mediante un ABB (recorrido inOrden)	509	20.11.1	Mapa conceptual	548
18.4.5	Búsqueda de un elemento sobre un ABB (método buscar).....	510	20.11.2	Autoevaluaciones	548
18.4.6	Eliminar un elemento del ABB (método eliminar).....	512	20.11.3	Videotutorial	548
18.5	Árbol n-ario	513	20.11.3.1	Algoritmo heapsort, ordenamiento por montículos.....	548
18.5.1	Nodo del árbol n-ario	514	20.11.4	Presentaciones*	548
18.5.2	Recorridos sobre un árbol n-ario	514	21.	Estrategia algorítmica	549
18.5.3	Permutar los caracteres de una cadena.....	515	21.1	Introducción	
18.5.4	Implementación de un "AutoSuggest".....	516	21.2	Divide y conquista	
18.6	Resumen	518	21.3	Greddy, algoritmos voraces	
18.7	Contenido de la página Web de apoyo.....	519	21.4	Programación dinámica	
18.7.1	Mapa conceptual	519	21.5	Resumen	
18.7.2	Autoevaluaciones.....	519	21.6	Contenido de la página Web de apoyo	
18.7.3	Presentaciones*	519			
19.	Complejidad algorítmica	521	22.	Algoritmos sobre grafos	551
19.1	Introducción.....	522	22.1	Introducción	
19.2	Conceptos iniciales	522	22.2	Definición de grafo	
19.2.1	Análisis del algoritmo de la búsqueda secuencial	522	22.3	El problema de los caminos mínimos	
19.3	Notación O grande (cota superior asintótica).....	523	22.4	Árbol de cubrimiento mínimo (MST)	
19.3.1	Análisis del algoritmo de la búsqueda binaria.....	524	22.5	Resumen	
19.3.2	Análisis del algoritmo de ordenamiento por burbujeo.....	526	22.6	Contenido de la página Web de apoyo	
19.4	Cota inferior (Ω) y cota ajustada asintótica (Θ).....	527			
19.5	Resumen	527	Bibliografía.....		553
19.6	Contenido de la página Web de apoyo.....	528			
19.6.1	Mapa conceptual	528			
19.6.2	Autoevaluaciones.....	528			
19.6.3	Presentaciones*	528			
20.	Algoritmos de ordenamiento	529			
20.1	Introducción.....	530			
20.2	Bubble sort (ordenamiento por burbujeo).....	531			
20.2.1	Bubble sort optimizado.....	534			
20.3	Selection sort (ordenamiento por selección)	534			
20.4	Insertion sort (ordenamiento por inserción)	535			
20.5	Quicksort (ordenamiento rápido)	536			
20.5.1	Implementación utilizando arrays auxiliares.....	536			
20.5.2	Implementación sin arrays auxiliares.....	537			

Información del contenido de la página Web

El material marcado con asterisco (*) solo está disponible para docentes.

Capítulo 1

Introducción a los algoritmos y a la programación de computadoras

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Instalación y uso de Eclipse para C
- Presentaciones*

Capítulo 2

Estructuras básicas de control y lógica algorítmica

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Uso del debugger para depurar un programa
- Presentaciones*

Capítulo 3

Funciones, modularización y metodología top-down

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Mantener archivos de funciones separados del programa principal
- Presentaciones*

Capítulo 4

Tipos de datos alfanuméricos

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 5

Punteros a carácter

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 6

Punteros, arrays y aritmética de direcciones

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Pasar argumentos en línea de comandos con Eclipse

- Presentaciones*

Capítulo 7

Tipos de datos estructurados

- Mapa conceptual
- Autoevaluación
- Videotutoriales:
 - Algoritmo de la burbuja
 - Algoritmo de la búsqueda binaria
- Presentaciones*

Capítulo 8

Operaciones sobre archivos

- Mapa conceptual
- Autoevaluación
- Videotutoriales:
 - Leer y escribir un archivo
 - Leer y escribir un archivo de registros
- Presentaciones*

Capítulo 9

Tipo Abstracto de Dato (TAD)

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 10

Análisis de ejercicios integradores

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 11

Estructuras de datos dinámicas lineales

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 12

Encapsulamiento a través de clases y objetos

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 13**Introducción al lenguaje de programación Java**

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Instalar y utilizar Eclipse para Java
- Presentaciones*

Capítulo 14**Programación orientada a objetos**

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Uso del javadoc
- Presentaciones*

Capítulo 15**Estructuras de datos dinámicas lineales en Java**

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 16**Compresión de archivos mediante el algoritmo de Huffman**

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Algoritmo de Huffman
- Presentaciones*

Capítulo 17**Recursividad**

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 18**Árboles**

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 19**Complejidad algorítmica**

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Algoritmo heapsort, ordenamiento por montículos
- Presentaciones*

Capítulo 20**Algoritmos de ordenamiento**

- Mapa conceptual
- Autoevaluación
- Presentaciones*

Capítulo 21**Estrategia algorítmica**

- Mapa conceptual
- Autoevaluación
- Videotutorial:
 - Problema de los billetes por programación dinámica
- Presentaciones*

Capítulo 22**Algoritmos sobre grafos**

- Mapa conceptual
- Autoevaluación
- Videotutoriales:
 - Algoritmo de Dijkstra por greedy
 - Algoritmo de Dijkstra por dinámica
 - Algoritmo de Prim
 - Algoritmo de Kruskal
- Presentaciones*

Código fuente de cada capítulo**Hipervínculos de interés****Fe de erratas****Guía para el docente de las competencias específicas que se desarrollan con este libro ***