

CONTENIDO

Dedicatoria	xvii
Agradecimientos	xix
Prefacio	xxi

Capítulo 1. Introducción a la Programación Orientada a Objetos

1.1	Introducción	1
1.2	Resolución de problemas mediante la computadora.....	1
1.3	Conceptos de Programación Orientada a Objetos	4
1.3.1	Clase	4
1.3.2	Objeto	4
1.3.3	Atributo	4
1.3.4	Método.....	5
1.3.5	Propiedad	5
1.3.6	Constructor	5
1.3.7	Destructor.....	5
1.3.8	Mensaje.....	5
1.3.9	Instancia	5
1.3.10	Sobrecarga.....	5
1.3.11	Clase abstracta.....	5
1.3.12	Interfase	6
1.3.13	Clase parametrizada.....	6
1.4	Características de la POO	6
1.4.1	Abstracción.....	6
1.4.2	Encapsulamiento	6
1.4.3	Modularidad	7
1.4.4	Herencia	7
1.4.5	Polimorfismo	7
1.5	Ventajas de la POO	8
1.6	Análisis y diseño orientado a objetos	8

1.7	Introducción a UML	10
1.7.1	Diagramas de clases en UML.....	11
1.7.2	Restricciones de acceso de los componentes de una clase	11
1.7.2.1	Público.....	12
1.7.2.2	Privado.....	12
1.7.2.3	Protegido.....	12
1.7.3	Relaciones entre clases	12
1.7.3.1	Asociación.....	14
1.7.3.1.1	Agregación	14
1.7.3.1.2	Composición	15
1.7.3.2	Generalización ó herencia	16
1.7.4	Clases abstractas e interfaces.....	18
1.7.5	Clases parametrizadas.....	20
1.8	Cuestionario	21

Capítulo 2. El lenguaje de programación Microsoft C# .NET

2.1	Introducción	23
2.2	La plataforma .NET	23
2.3	El compilador Microsoft Visual C# 2010 Express.....	25
2.3.1	Descarga de Microsoft Visual C# 2010 Express.	25
2.3.2	Creación de proyectos con Microsoft Visual C# 2010 Express.....	25
2.4	¿Cómo agregar clases al proyecto?	28
2.5	Controles de formas de Windows	30
2.5.1	Propiedades de los controles de formas de Windows.....	30
2.5.2	Controles más comunes.	31
2.5.2.1	label.....	32
2.5.2.2	textBox	33
2.5.2.3	button.....	33
2.5.2.4	MessageBox.....	35
2.5.2.5	checkBox.....	36
2.5.2.6	radioButton.....	38

2.5.2.7	groupBox / panel.....	38
2.5.2.8	comboBox.....	39
2.5.2.9	listBox.....	40
2.5.2.10	dataGridView	41
2.5.2.11	pictureBox.....	45
2.5.2.12	tabControl y tabPage.....	45
2.6	Cuestionario	47

Capítulo 3. Arreglos

3.1.	Introducción	49
3.2.	Definición de arreglo y sus características	49
3.2	Arreglos unidimensionales.....	50
3.2.1	Representación de arreglos unidimensionales.....	50
3.2.2	Operaciones con arreglos unidimensionales.....	51
3.2.3	Ejemplos de arreglos unidimensionales.....	53
3.2.4	Situaciones críticas al trabajar con arreglos unidimensionales.....	54
3.3	Arreglos unidimensionales desordenados.....	55
3.3.1	Definición de una clase para el arreglo desordenado que almacena números enteros. 55	
3.3.2	Prog. 3.1.- Proyecto de consola en C# con un arreglo unidimensional desordenado: Datos enteros.....	62
3.3.3	Prog. 3.2.- Proyecto de formas de Windows en C# con un arreglo unidimensional desordenado: Equipo de beisbol	68
3.3.3.1	Inserción de los datos de un jugador	68
3.3.3.2	Eliminación de los datos de un jugador	69
3.3.3.3	Ordenar en forma ascendente los jugadores de acuerdo al porcentaje de bateo	70
3.3.3.4	Mostrar los datos del mejor bateador	70
3.3.3.5	Definición de las clases para el arreglo unidimensional con objetos desordenados.....	70
3.4	Arreglos unidimensionales ordenados	79
3.4.1	Definición de una clase para el arreglo unidimensional ordenado	80
3.4.2	Prog. 3.3.- Proyecto de consola en C# con un arreglo unidimensional ordenado: Datos enteros.....	85

3.4.3	Prog. 3.4.- Proyecto de formas de Windows en C# con un arreglo unidimensional ordenado: Calificaciones	90
3.4.3.1	Inserción de los datos de un estudiante	91
3.4.3.2	Eliminación de los datos de un estudiante.....	92
3.4.3.3	Calcular el promedio de calificaciones.....	92
3.4.3.4	Vaciar el arreglo	92
3.4.3.5	Definición de las clases para el arreglo unidimensional ordenado con objetos.....	93
3.5	Arreglos bidimensionales.....	101
3.5.1	Representación de arreglos bidimensionales.....	102
3.5.2	Operaciones con arreglos bidimensionales.....	103
3.5.3	Ejemplos de arreglos bidimensionales.....	104
3.5.4	Definición de una clase para un arreglo bidimensional que almacene números reales	104
3.5.5	Prog. 3.5.- Proyecto de consola en C# con un arreglo bidimensional: Suma y multiplicación matricial	108
3.5.6	Proyecto de formas de Windows con un arreglo bidimensional: Ventas	114
3.5.6.1	Inserción de los datos de las ventas a la matriz.....	114
3.5.6.2	Obtener el mes con la mayor venta de dulces	115
3.5.6.3	Promedio anual de venta de bebidas.....	115
3.5.6.4	Obtener el mes con la menor venta de bebidas.....	116
3.5.6.5	Obtener departamento con la menor venta en agosto	116
3.5.6.6	Definición de una clase para un arreglo bidimensional que almacene números reales para la aplicación de formas de Windows.....	117
3.5.6.7	Prog. 3.6.- Proyecto de formas de Windows en C# con un arreglo bidimensional: Ventas mensuales	118
3.6	Arreglos tridimensionales	125
3.6.1	Representación de arreglos tridimensionales	126
3.6.2	Operaciones con arreglos tridimensionales	127
3.6.3	Ejemplos de arreglos tridimensionales	128
3.6.4	Definición de una clase para un arreglo tridimensional que almacene calificaciones	128
3.6.5	Prog. 3.7.- Proyecto de consola en C# con un arreglo tridimensional: Calificaciones	135
3.6.6	Prog. 3.8.- Proyecto de formas de Windows con un arreglo tridimensional: Ventas.....	145
3.6.6.1	Inserción de los datos de las ventas.....	147

3.6.6.2	Cálculo del promedio de ventas por trimestre	147
3.6.6.3	Cálculo del promedio de ventas por producto	148
3.6.6.4	Cálculo del promedio de ventas por año.....	149
3.6.6.5	Definición de la clase para un arreglo tridimensional que almacene las ventas.....	150
3.7	Aplicación de arreglos unidimensionales usando la clase genérica ArrayList.....	159
3.7.1	Principales métodos y propiedades de la clase genérica ArrayList	160
3.7.2	Prog. 3.9.- Proyecto en C# de una aplicación de consola utilizando la clase genérica ArrayList	160
3.8	Cuestionario	165
3.9	Ejercicios con arreglos unidimensionales	167
3.10	Ejercicios con arreglos bidimensionales	168
3.11	Ejercicios con arreglos tridimensionales.....	169

Capítulo 4. Pilas

4.1	Introducción	171
4.2	Representación de pilas.....	171
4.3.	Operaciones con pilas.....	172
4.4.	Ejemplos de pilas.....	173
4.5.	Situaciones críticas al trabajar con pilas	174
4.6.	Definición de la clase para una pila	174
4.7.	Prog. 4.1.- Proyecto de consola en C# con una pila: Datos enteros.....	178
4.8.	Prog. 4.2.- Proyecto de formas de Windows en C# con una pila: Estacionamiento de autos	182
4.8.1	Entrada de auto (Push):.....	183
4.8.2	Salida de auto (Pop):	184
4.8.3	Definición de las clases para la pila.....	184
4.9.	Aplicación de pilas usando la clase genérica Stack.....	191
4.9.1	Principales métodos y propiedades de la clase genérica Stack.....	192
4.9.2	Prog.4.3.- Proyecto de consola en C# con una aplicación utilizando la clase genérica Stack	192
4.10.	Cuestionario	196
4.11.	Ejercicios	197

Capítulo 5. Colas

5.1.	Introducción	199
5.2.	Representación de colas	199
5.3.	Operaciones con colas	200
5.4.	Ejemplo de colas.....	201
5.5.	Situaciones críticas al trabajar con colas	202
5.6.	Definición de la clase para una cola	202
5.7.	Prog. 5.1.- Proyecto de consola en C# con una cola: Datos enteros	206
5.8.	Prog. 5.2.- Proyecto de formas de Windows en C# con una cola: Banco	211
5.8.1	Entrada de un cliente en la ventanilla (Agregar a la cola).....	212
5.8.2	Atender en ventanilla (Eliminación).....	212
5.8.3	Definición de las clases para una cola	213
5.9.	Aplicación de colas usando la clase genérica Queue.....	222
5.9.1	Principales métodos y propiedades de la clase genérica Queue.....	222
5.9.2	Prog. 5.3.- Proyecto de consola en C# con una aplicación utilizando la clase genérica Queue	222
5.10.	Colas circulares.....	226
5.10.1	Definición de una clase para la cola circular	227
5.10.2	Prog. 5.4.- Proyecto de consola en C# con una aplicación de una cola circular: Datos enteros	232
5.10.3	Prog. 5.5.- Proyecto de formas de Windows en C# con una cola circular: marco fotográfico digital.....	237
5.10.3.1	Agregar imagen.....	237
5.10.3.2	Eliminar primera imagen	238
5.10.3.3	Iniciar presentación.....	238
5.10.3.4	Definición de la clase para una cola circular.....	239
5.11.	Cuestionario	247
5.12.	Ejercicios	249

Capítulo 6. Listas enlazadas

6.1.	Introducción	251
6.2.	Representación de listas simples.....	252
6.3.	Operaciones y situaciones críticas que se pueden presentar al trabajar con listas simples	252

6.4.	Ejemplos de listas simples.....	257
6.5.	Definición de las clases para una lista simple ordenada.....	258
6.6.	Prog. 6.1.- Proyecto de consola en C# con una lista enlazada simple ordenada: Datos enteros	267
6.7.	Prog. 6.2.- Proyecto de formas de Windows en C# con listas enlazadas simples ordenadas: Estudiantes y sus libros	277
6.7.1	Insertar objetos.....	279
6.7.2	Borrar un objeto	280
6.7.3	Definición de las clases para una lista simple.....	280
6.8.	Aplicación de listas simples usando la clase genérica List	300
6.8.1	Principales métodos y propiedades de la clase genérica List.....	300
6.8.2	Prog. 6.3.- Proyecto de consola en C# con una aplicación de la clase genérica List.....	300
6.9.	Implementación de pilas mediante listas simples.....	306
6.9.1	Representación de una pila con datos desordenados mediante una lista simple (Pila-Desordenada-Lista)	306
6.9.1.1	Operaciones básicas con una Pila-Desordenada-Lista	307
6.9.1.2	Situaciones críticas al trabajar con una Pila-Desordenada-Lista	307
6.9.1.3	Definición de las clases para una Pila-Desordenada-Lista.....	308
6.9.1.4	Prog. 6.4.- Proyecto de consola en C# con una Pila-Desordenada-Lista: Datos enteros 313	
6.9.2	Representación de una pila con datos ordenados mediante una lista simple (Pila-Ordenada-Lista).....	321
6.9.2.1	Operaciones básicas con una Pila-Ordenada-Lista	322
6.9.2.2	Situaciones críticas al trabajar con una Pila-Ordenada-Lista.....	322
6.9.2.3	Definición de las clases para una Pila-Ordenada-Lista.....	322
6.9.2.4	Prog. 6.5.- Proyecto de consola en C# con una Pila-Ordenada-Lista: Datos enteros ...	326
6.10.	Implementación de una cola mediante una lista simple	335
6.10.1	Representación de una cola mediante una lista simple (Cola-Lista)	335
6.10.2	Operaciones básicas con una Cola-Lista.....	336
6.10.3	Situaciones críticas al trabajar con una Cola-Lista	336
6.10.4	Definición de la clase para una Cola-Lista	336
6.10.5	Prog. 6.6.- Proyecto de consola en C# con una Cola-Lista: Datos enteros.....	343
6.11.	Listas enlazadas dobles	351

6.11.1	Representación de listas dobles	351
6.11.2	Operaciones y situaciones críticas que se pueden presentar al trabajar con listas dobles	352
6.11.3	Ejemplo de listas dobles.....	359
6.11.4	Definición de las clases para una lista doble con nodos ordenados.....	359
6.12.	Prog. 6.7.- Proyecto de consola en C# con una lista doble ordenada: Datos enteros.....	372
6.13.	Prog. 6.8.- Proyecto de formas de Windows con una lista doble desordenada: Ferrocarril.....	383
6.13.1	Agregar un vagón a la lista doble.....	384
6.13.2	Eliminar un vagón de la lista doble	385
6.13.3	Mostrar desde el principio y desde el final	386
6.13.4	Buscar vagón.....	386
6.13.5	Definición de las clases de la lista doble para almacenar los datos del ferrocarril	387
6.14.	Aplicación de listas dobles usando las clases genéricas LinkedList y LinkedListNode.....	406
6.14.1	Principales métodos y propiedades de las clases genéricas LinkedList y LinkedListNode.....	406
6.14.2	Prog. 6.9.- Proyecto de consola en C# con una aplicación utilizando las clases genéricas LinkedList y LinkedListNode	407
6.15.	Cuestionario	415
6.16.	Ejercicios	416

Capítulo 7. Recursividad

7.1	Introducción	419
7.2	Ejemplos de recursividad en la vida cotidiana	419
7.3	Algoritmos recursivos	420
7.4	Ventajas y desventajas del uso de algoritmos recursivos.....	421
7.5	Ámbito de las variables: variables globales y variables locales.....	422
7.6	Envío de parámetros a un método: por valor y por referencia	422
7.6.1	Parámetros por valor	423
7.6.1.1	Prog. 7.1.- Proyecto de consola en C#: Envío de parámetros por valor.....	424
7.6.2	Parámetros por referencia	425
7.6.2.1	Prog. 7.2.- Proyecto de consola en C#: Envío de parámetros por referencia.....	427
7.6.2.2	Parámetros de salida en C#	428

7.6.2.2.1	Prog. 7.3.- Proyecto de consola en C#: Parámetros de salida	428
7.7	Algoritmo recursivo para calcular el factorial de un número	430
7.6.3	Prog. 7.4.- Proyecto de consola en C# con un algoritmo recursivo: Cálculo del factorial 433	
7.8	Algoritmo recursivo para resolver el juego de las torres de Hanoi	435
7.9	Prog. 7.5.- Proyecto de formas de Windows en C# con un algoritmo recursivo: Las torres de Hanoi	440
7.10	Cuestionario	449
7.11	Ejercicios	449

Capítulo 8. Árboles binarios

8.1	Introducción	451
8.2	Árboles binarios.....	452
8.3	Representación de árboles binarios	453
8.4	Árboles binarios de búsqueda.....	454
8.5	Operaciones y situaciones críticas que se pueden presentar al trabajar con árboles binarios de búsqueda.....	454
8.6	Ejemplos de árboles binarios	465
8.7	Definición de las clases para un árbol binario de búsqueda	466
8.8	Prog. 8.1.- Proyecto de consola en C# con un ABB: Datos enteros.....	478
8.9	Prog. 8.2.- Proyecto de formas de Windows con un ABB: Dibujo de un ABB	489
8.9.1	¿Cómo dibujar una estructura de datos utilizando Graphviz?	490
8.9.2	Graphviz y el lenguaje <i>dot</i>	490
8.9.2.1	Un componente de Graphviz: el lenguaje <i>dot</i>	491
8.9.2.2	Sintaxis de un programa en lenguaje <i>dot</i>	491
8.9.2.3	¿Cómo ejecutar un programa en lenguaje <i>dot</i> ?	492
8.9.2.4	¿Cómo generar una imagen desde una aplicación de formas de Windows utilizando un programa en lenguaje <i>dot</i> ?.....	494
8.9.3	Representación de las clases	494
8.10	Cuestionario	502
8.11	Ejercicios	504

Capítulo 9. Grafos

9.1.	Introducción	505
9.2	Representación de grafos	508
9.2.1	Representación secuencial de grafos: Matriz de adyacencia.....	508
9.2.1.1	El algoritmo de Warshall (matriz de caminos).....	510
9.2.1.1.1	Definición de la clase para el algoritmo de Warshall	513
9.2.1.1.2	Prog. 9.1.- Proyecto de consola en C# con el algoritmo de Warshall	515
9.2.1.2	Matriz de pesos para grafos ponderados.	522
9.2.1.3	Algoritmo de Floyd (matriz de caminos mínimos)	523
9.2.1.3.1	Definición de la clase para el algoritmo de Floyd	528
9.2.1.3.2	Prog. 9.2.- Proyecto de consola en C# con el algoritmo de Floyd.....	529
9.2.2	Representación enlazada de grafos: Listas enlazadas	537
9.3	Operaciones con grafos representados mediante listas.....	538
9.4	Ejemplos de grafos	540
9.5	Definición de las clases para un grafo dirigido no ponderado representado mediante listas	542
9.5.1	Búsqueda de un camino entre dos nodos: Recorridos en profundidad y en anchura.....	554
9.5.2	Representación de los recorridos en profundidad y anchura	559
9.6	Prog. 9.3.- Proyecto de consola en C# con un grafo dirigido no ponderado: Datos enteros....	564
9.7	Definición de las clases para un grafo no dirigido y ponderado representado mediante listas	599
9.7.1	Búsqueda de un camino más corto entre dos nodos: Recorrido del camino mínimo.....	608
9.7.2	Representación del recorrido del camino mínimo	614
9.7.3	Representación de las clases necesarias para dibujar el grafo.....	618
9.8	Prog. 9.4.- Proyecto de formas de Windows en C# con un grafo no dirigido y ponderado: Mapa carretero entre ciudades	621
9.8.1	Insertar ciudad.....	622
9.8.2	Eliminar ciudad	622
9.8.3	Cargar un ejemplo prediseñado	623
9.8.4	Dibujar el mapa.....	624
9.8.5	Insertar una carretera	625
9.8.6	Eliminar una carretera	626

9.8.7	Buscar camino más corto (camino mínimo).....	626
9.8.8	Representación de las clases.....	627
9.9	Cuestionario.....	650
9.10	Ejercicios.....	653

Capítulo 10. Métodos de ordenamiento

10. 1.	Introducción.....	655
10. 2.	Criterios de ordenamiento.....	655
10. 3.	Tipos de ordenamiento.....	655
10. 4.	Consideraciones importantes para el pseudocódigo de los métodos.....	656
10. 5.	Métodos de ordenamiento interno.....	657
10.5.1.	Métodos de ordenamiento interno por intercambio.....	658
10.5.1.1.	Burbuja con desplazamientos hacia la izquierda del arreglo.....	658
10.5.1.2.	Burbuja con desplazamientos hacia la derecha del arreglo.....	662
10.5.1.3.	Burbuja con señal.....	665
10.5.1.4.	Shaker Sort (sacudida).....	669
10.5.1.5.	Quick Sort (ordenamiento rápido).....	674
10.5.2.	Métodos de ordenamiento interno por selección.....	682
10.5.2.1.	Selección directa.....	682
10.5.3.	Métodos de ordenamiento interno por inserción.....	685
10.5.3.1.	Inserción directa.....	685
10.5.3.2.	Inserción binaria.....	688
10.5.3.3.	Shell (inserción con saltos decrecientes).....	690
10.5.4.	Métodos de ordenamiento interno de árbol.....	693
10.5.4.1.	Heap Sort (árbol ó montículo).....	693
10.5.5.	Definición de las clases para los métodos de ordenamiento interno.....	698
10.5.6.	Consideraciones importantes para la implementación de las clases.....	700
10.5.7.	Prog. 10.1.- Proyecto de consola en C# con métodos de ordenamiento interno: Datos enteros.....	702
10.5.8.	Prog. 10.2.- Proyecto de formas de Windows en C# con métodos de ordenamiento interno: Datos de estudiantes.....	726
10.5.8.1.	Insertar los datos de un estudiante.....	727

10.5.8.2.	Ordenar los datos	727
10.5.8.3.	Representación de las clases	727
10.5.9.	Análisis de eficiencia de los métodos de ordenamiento interno	736
10.5.9.1.	Casos de análisis: Mejor, peor y promedio.....	737
10.5.9.2.	Notación O-grande	737
10.5.9.3.	Eficiencia del método de la Burbuja.....	738
10.5.9.4.	Eficiencia del método de la Burbuja con señal	739
10.5.9.5.	Eficiencia del método Shaker Sort (sacudida).....	739
10.5.9.6.	Eficiencia del método Quick Sort (ordenamiento rápido)	739
10.5.9.7.	Eficiencia del método de Selección directa	740
10.5.9.8.	Eficiencia del método de Inserción directa.....	740
10.5.9.9.	Eficiencia del método de Inserción binaria.....	742
10.5.9.10.	Eficiencia del método Shell (inserción con saltos decrecientes)	742
10.5.9.11.	Eficiencia del método Heap Sort (árbol ó montículo)	743
10.5.9.12.	Comparación entre métodos: ventajas y desventajas	743
10.5.9.13.	Elección de un método de ordenamiento	744
10.7.	Cuestionario.....	746
10.8.	Ejercicios.....	748
	BIBLIOGRAFÍA.....	749