

# CONTENIDO

## PRÓLOGO

XIX

## CAPÍTULO 1

### INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

1.1	Introducción	2
1.2	Resolución de problemas mediante la computadora	3
1.3	Conceptos de programación orientada a objetos	5
1.3.1	Clase	5
1.3.2	Objeto	5
1.3.3	Atributo	5
1.3.4	Método	5
1.3.5	Propiedad	6
1.3.6	Constructor	6
1.3.7	Destructor	6
1.3.8	Mensaje	6
1.3.9	Instancia	6
1.3.10	Sobrecarga	6
1.3.11	Clase abstracta	6
1.3.12	Interface	7
1.3.13	Clase parametrizada	7
1.4	Características de la POO	7
1.4.1	Abstracción	7
1.4.2	Encapsulamiento	8
1.4.3	Modularidad	8
1.4.4	Herencia	8
1.4.5	Polimorfismo	8
1.5	Ventajas de la POO	9
1.6	Análisis y diseño orientado a objetos	9
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	12
1.7.3	Relaciones entre clases	13
1.7.4	Clases abstractas e interfaces	17
1.7.5	Clases parametrizadas	19

Aprende en la



**QUESTIONARIO** LECTURA COMPLEMENTARIA

## CAPÍTULO 2

### EL LENGUAJE DE PROGRAMACIÓN MICROSOFT C#.NET

2.1	Introducción	22
-----	--------------	----

22

2.2	La plataforma.NET	22
2.3	El compilador Microsoft Visual C# 2010 Express	23
2.3.1	Descarga de Microsoft Visual C# 2010 Express	24
2.3.2	Creación de proyectos con Microsoft Visual C# 2010 Express	24
2.4	¿Cómo agregar clases al proyecto?	26
2.5	Controles de formas de Windows	27
2.5.1	Propiedades de los controles de formas de Windows	28
2.5.2	Controles más comunes	28

Apéndice en la



**CUESTIONARIO LECTURA COMPLEMENTARIA**

**CAPÍTULO 3  
ARREGLOS**

3.1	Introducción	42
3.2	Definición de arreglo y sus características	42
3.3	Arreglos unidimensionales	43
3.3.1	Representación de arreglos unidimensionales	43
3.3.2	Operaciones con arreglos unidimensionales	44
3.3.3	Ejemplos de arreglos unidimensionales	46
3.3.4	Situaciones críticas al trabajar con arreglos unidimensionales	46
3.4	Arreglos unidimensionales desordenados	47
3.4.1	Definición de una clase para el arreglo desordenado que almacena números enteros	47
3.4.2	Programa 3.1. Proyecto de consola en C# con un arreglo unidimensional desordenado: Datos enteros	53

Apéndice en la



**PROGRAMA 3.1 CÓDIGO FUENTE**

3.4.3	Programa 3.2. Proyecto de formas de Windows en C# con un arreglo unidimensional desordenado: Equipo de béisbol	56
-------	--	----

Apéndice en la



**PROGRAMA 3.2 CÓDIGO FUENTE**

3.5	Arreglos unidimensionales ordenados	65
3.5.1	Definición de una clase para el arreglo unidimensional ordenado	65
3.5.2	Programa 3.3. Proyecto de consola en C# con un arreglo unidimensional ordenado: Datos enteros	69

Apéndice en la



**PROGRAMA 3.3 CÓDIGO FUENTE**

<b>3.5.3</b>	Programa 3.4. Proyecto de formas de Windows en C# con un arreglo unidimensional ordenado: Calificaciones	73
--------------	--	----



**PROGRAMA 3.4 CÓDIGO FUENTE**

<b>3.6</b>	Arreglos bidimensionales	82
<b>3.6.1</b>	Representación de arreglos bidimensionales	82
<b>3.6.2</b>	Operaciones con arreglos bidimensionales	83
<b>3.6.3</b>	Ejemplos de arreglos bidimensionales	83
<b>3.6.4</b>	Definición de una clase para un arreglo bidimensional que almacene números reales	84
<b>3.6.5</b>	Programa 3.5. Proyecto de consola en C# con un arreglo bidimensional: suma y multiplicación matriciales	86



**PROGRAMA 3.5 CÓDIGO FUENTE**

<b>3.6.6</b>	Proyecto de formas de Windows con un arreglo bidimensional: Ventas	91
--------------	--	----



**PROGRAMA 3.6 CÓDIGO FUENTE**

<b>3.7</b>	Arreglos tridimensionales	99
<b>3.7.1</b>	Representación de arreglos tridimensionales	99
<b>3.7.2</b>	Operaciones con arreglos tridimensionales	101
<b>3.7.3</b>	Ejemplos de arreglos tridimensionales	101
<b>3.7.4</b>	Definición de una clase para un arreglo tridimensional que almacene calificaciones	101
<b>3.7.5</b>	Programa 3.7. Proyecto de consola en C# con un arreglo tridimensional: Calificaciones	107



**PROGRAMA 3.7 CÓDIGO FUENTE**

<b>3.7.6</b>	Programa 3.8. Proyecto de formas de Windows con un arreglo tridimensional: Ventas	113
--------------	---	-----



**PROGRAMA 3.8 CÓDIGO FUENTE**

<b>3.8</b>	Aplicación de arreglos unidimensionales usando la clase genérica ArrayList	123
<b>3.8.1</b>	Principales métodos y propiedades de la clase genérica ArrayList	123
<b>3.8.2</b>	Programa 3.9. Proyecto en C# de una aplicación de consola utilizando la clase genérica ArrayList	123

Aprensión en la



**PROGRAMA 3.9** CÓDIGO FUENTE

Aprensión en la



**CUESTIONARIO** LECTURA COMPLEMENTARIA

Aprensión en la



**EJERCICIOS CON ARREGLOS**

**UNIIDIMENSIONALES** LECTURA COMPLEMENTARIA

Aprensión en la



**EJERCICIOS CON ARREGLOS**

**BIDIMENSIONALES** LECTURA COMPLEMENTARIA

Aprensión en la



**EJERCICIOS CON ARREGLOS TRIDIMENSIONALES**

LECTURA COMPLEMENTARIA

**CAPÍTULO 4**  
**PILAS**

4.1	Introducción	126
4.2	Representación de pilas	126
4.3	Operaciones con pilas	127
4.4	Ejemplos de pilas	128
4.5	Situaciones críticas al trabajar con pilas	129
4.6	Definición de la clase para una pila	129
4.7	Programa 4.1 Proyecto de consola en C# con una pila: datos enteros	132

Aprensión en la



**PROGRAMA 4.1** CÓDIGO FUENTE

4.8	Programa 4.2 Proyecto de formas de Windows en C# con una pila: estacionamiento de autos	135
4.8.1	Entrada de auto (Push)	136
4.8.2	Salida de auto (Pop)	136
4.8.3	Definición de las clases para la pila	137

Aprensión en la



**PROGRAMA 4.2** CÓDIGO FUENTE

4.9	Aplicación de pilas usando la clase genérica Stack	142
4.9.1	Principales métodos y propiedades de la clase genérica Stack	143
4.9.2	Programa 4.3. Proyecto de consola en C# con una aplicación utilizando la clase genérica Stack	143

Aprensión en la



**CUESTIONARIO** LECTURA COMPLEMENTARIA



## EJERCICIOS LECTURA COMPLEMENTARIA

### CAPÍTULO 5 COLAS

5.1	Introducción	146
5.2	Representación de colas	146
5.3	Operaciones con colas	147
5.4	Ejemplos de colas	148
5.5	Situaciones críticas al trabajar con colas	149
5.6	Definición de la clase para una cola	149
5.7	Programa 5.1 Proyecto de consola en C# con una cola: datos enteros	153



#### PROGRAMA 5.1 CÓDIGO FUENTE

5.8	Programa 5.2 Proyecto de formas de Windows en C# con una cola: banco	156
5.8.1	Entrada de un cliente en la ventanilla (Agregar a la cola)	157
5.8.2	Atender en ventanilla (Eliminación)	157
5.8.3	Definición de las clases para una cola	158



#### PROGRAMA 5.2 CÓDIGO FUENTE

5.9	Aplicación de colas usando la clase genérica Queue	164
5.9.1	Principales métodos y propiedades de la clase genérica Queue	164
5.9.2	Programa 5.3. Proyecto de consola en C# con una aplicación utilizando la clase genérica Queue	165



#### PROGRAMA 5.3 CÓDIGO FUENTE

5.10	Colas circulares	166
5.10.1	Definición de una clase para la cola circular	167
5.10.2	Programa 5.4. Proyecto de consola en C# con una aplicación de una cola circular: Datos enteros	170



#### PROGRAMA 5.4 CÓDIGO FUENTE

5.10.3	Programa 5.5. Proyecto de formas de Windows en C# con una cola circular: marco fotográfico digital	174
--------	---	-----



#### PROGRAMA 5.5 CÓDIGO FUENTE



### **CUESTIONARIO** LECTURA COMPLEMENTARIA



### **EJERCICIOS** LECTURA COMPLEMENTARIA

## **CAPÍTULO 6** **LISTAS ENLAZADAS**

<b>6.1</b>	Introducción	184
<b>6.2</b>	Representación de listas simples	185
<b>6.3</b>	Operaciones y situaciones críticas que se pueden presentar al trabajar con listas simples	185
<b>6.4</b>	Ejemplos de listas simples	190
<b>6.5</b>	Definición de las clases para una lista simple ordenada	190
<b>6.6</b>	Programa 6.1 Proyecto de consola en C# con una lista enlazada simple ordenada: datos enteros	198



### **PROGRAMA 6.1** CÓDIGO FUENTE

<b>6.7</b>	Programa 6.2 Proyecto de formas de Windows en C# con listas enlazadas simples ordenadas: estudiantes y sus libros	207
<b>6.7.1</b>	Insertar objetos	208
<b>6.7.2</b>	Borrar un objeto	208
<b>6.7.3</b>	Definición de las clases para una lista simple	209



### **PROGRAMA 6.2** CODIGO FUENTE

<b>6.8</b>	Aplicación de listas simples usando la clase genérica List	223
<b>6.8.1</b>	Principales métodos y propiedades de la clase genérica List	224
<b>6.8.2</b>	Programa 6.3 Proyecto de consola en C# con una aplicación de la clase genérica List	224



### **PROGRAMA 6.3** CÓDIGO FUENTE

<b>6.9</b>	Implementación de pilas mediante listas simples	227
<b>6.9.1</b>	Representación de una pila con datos desordenados mediante una lista simple (Pila-Desordenada-Lista)	227



### **PROGRAMA 6.4** CÓDIGO FUENTE

<b>6.9.2</b>	Representación de una pila con datos ordenados mediante una lista simple (pila-ordena-lista)	239
--------------	--	-----



### **PROGRAMA 6.5** CÓDIGO FUENTE



**EJERCICIOS LECTURA COMPLEMENTARIA**

**CAPÍTULO 7  
RECURSIVIDAD**

7.1	Introducción	314
7.2	Ejemplos de recursividad en la vida cotidiana	314
7.3	Algoritmos recursivos	315
7.4	Ventajas y desventajas del uso de algoritmos recursivos	316
7.5	Ámbito de las variables: variables globales y variables locales	316
7.6	Envío de parámetros a un método: por valor y por referencia	317



**PARÁMETROS POR VALOR Y POR REFERENCIA**

LECTURA COMPLEMENTARIA



**PROGRAMA 7.1 CÓDIGO FUENTE**



**PROGRAMA 7.2 CÓDIGO FUENTE**



**PROGRAMA 7.3 CÓDIGO FUENTE**

7.7	Algoritmo recursivo para calcular el factorial de un número	318
7.7.1	Programa 7.4. Proyecto de consola en C# con un algoritmo recursivo: Cálculo del factorial	320



**PROGRAMA 7.4 CÓDIGO FUENTE**

7.8	Algoritmo recursivo para resolver el juego de las torres de Hanoi	320
7.9	Programa 7.5 Proyecto de formas de Windows en C# con un algoritmo recursivo: Las torres de Hanoi	324



**PROGRAMA 7.5 CÓDIGO FUENTE**



**CUESTIONARIO LECTURA COMPLEMENTARIA**



**EJERCICIOS LECTURA COMPLEMENTARIA**

**CAPÍTULO 8  
ÁRBOLES BINARIOS**

8.1	Introducción	328
-----	--------------	-----

<b>8.2</b> Árboles binarios	329
<b>8.3</b> Representación de árboles binarios	330
<b>8.4</b> Árboles binarios de búsqueda	330
<b>8.5</b> Operaciones y situaciones críticas que se pueden presentar al trabajar con árboles binarios de búsqueda	331
<b>8.6</b> Ejemplos de árboles binarios	338
<b>8.7</b> Definición de las clases para un árbol binario de búsqueda	339
<b>8.8</b> Programa 8.1. Proyecto de consola en C# con un ABB: Datos enteros	350



### PROGRAMA 8.1 CÓDIGO FUENTE

<b>8.9</b> Programa 8.2. Proyecto de formas de Windows en C# con un ABB: Dibujo de un ABB	360
<b>8.9.1</b> ¿Cómo dibujar una estructura de datos utilizando Graphviz?	360



### GRAPHVIZ Y EL LENGUAJE DOT LECTURA COMPLEMENTARIA



### PROGRAMA 8.2 CÓDIGO FUENTE



### CUESTIONARIO LECTURA COMPLEMENTARIA



### EJERCICIOS LECTURA COMPLEMENTARIA

## CAPÍTULO 9 GRAFOS

<b>9.1</b> Introducción	362
<b>9.2</b> Representación de grafos	364
<b>9.2.1</b> Representación secuencial de grafos: matriz de adyacencia	365




### PROGRAMA 9.1 CÓDIGO FUENTE




### PROGRAMA 9.2 CÓDIGO FUENTE

<b>9.2.2</b> Representación enlazada de grafos: listas enlazadas	376
<b>9.3</b> Operaciones con grafos representados mediante listas	377
<b>9.4</b> Ejemplos de grafos	379
<b>9.5</b> Definición de las clases para un grafo dirigido no ponderado representado mediante listas	380
<b>9.5.1</b> Búsqueda de un camino entre dos nodos: recorridos en profundidad y en anchura	389



9.5.2 Representación de los recorridos en profundidad y anchura	392
9.6 Programa 9.3. Proyecto de consola en C# con un grafo dirigido no ponderado: Datos enteros	396
<small>Añadir en la</small>	
 <b>PROGRAMA 9.3 CÓDIGO FUENTE</b>	
9.7 Definición de las clases para un grafo no dirigido y ponderado representado mediante listas	398
9.7.1 Búsqueda de un camino más corto entre dos nodos: recorrido del camino mínimo	405
9.7.2 Representación del recorrido del camino mínimo	410
9.7.3 Representación de las clases necesarias para dibujar el grafo	413
9.8 Programa 9.4. Proyecto de formas de Windows en C# con un grafo no dirigido y ponderado: Mapa carretero entre ciudades	415
9.8.1 Insertar ciudad	416
9.8.2 Eliminar ciudad	416
9.8.3 Cargar un ejemplo prediseñado	417
9.8.4 Dibujar el mapa	417
9.8.5 Insertar una carretera	418
9.8.6 Eliminar una carretera	418
9.8.7 Buscar camino más corto (camino mínimo)	419
9.8.8 Representación de las clases	419

Añadir en la

 **PROGRAMA 9.4 CÓDIGO FUENTE**

Añadir en la

 **CUESTIONARIO LECTURA COMPLEMENTARIA**

Añadir en la

 **EJERCICIOS LECTURA COMPLEMENTARIA**

**CAPÍTULO 10**  
**MÉTODOS DE ORDENAMIENTO**

10.1 Introducción	424
10.2 Criterios de ordenamiento	424
10.3 Tipos de ordenamiento	425
10.4 Consideraciones importantes para el pseudocódigo de los métodos	425
10.5 Métodos de ordenamiento interno	426
10.5.1 Métodos de ordenamiento interno por intercambio	427
10.5.2 Métodos de ordenamiento interno por selección	447
10.5.3 Métodos de ordenamiento interno por inserción	450

<b>10.5.4</b> Métodos de ordenamiento interno de árbol	457
<b>10.5.5</b> Definición de las clases para los métodos de ordenamiento interno	462
<b>10.5.6</b> Consideraciones importantes para la implementación de las clases	463
<b>10.5.7</b> Programa 10.1 Proyecto de consola en C# con métodos de ordenamiento interno: datos enteros	464



## PROGRAMA 10.1 CÓDIGO FUENTE

<b>10.5.8</b> Programa 10.2 Proyecto de formas de Windows en C# con métodos de ordenamiento interno: datos de estudiantes	469
---	-----



## PROGRAMA 10.2 CÓDIGO FUENTE

<b>10.5.9</b> Análisis de eficiencia de los métodos de ordenamiento interno	474
---	-----



## CUESTIONARIO LECTURA COMPLEMENTARIA



## EJERCICIOS LECTURA COMPLEMENTARIA

<b>Bibliografía</b>	<b>483</b>
---------------------	------------

<b>Índice analítico</b>	<b>485</b>
-------------------------	------------