

# 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 |

Apoyo 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 |

Apoyo 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 |

Apoyo 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 |
|-------|--|----|






Apoyo 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 |

Apoyo en la

**PROGRAMA 3.3 CÓDIGO FUENTE**

|  |     |
|--|-----|
| 3.5.3 Programa 3.4. Proyecto de formas de Windows en C# con un arreglo unidimensional ordenado: Calificaciones                           | 73  |
| <p>Apoyo en la  <b>PROGRAMA 3.4 CÓDIGO FUENTE</b></p>   |     |
| 3.6 Arreglos bidimensionales   | 82  |
| 3.6.1 Representación de arreglos bidimensionales   | 82  |
| 3.6.2 Operaciones con arreglos bidimensionales   | 83  |
| 3.6.3 Ejemplos de arreglos bidimensionales   | 83  |
| 3.6.4 Definición de una clase para un arreglo bidimensional que almacene números reales  | 84  |
| 3.6.5 Programa 3.5. Proyecto de consola en C# con un arreglo bidimensional: suma y multiplicación matriciales                            | 86  |
| <p>Apoyo en la  <b>PROGRAMA 3.5 CÓDIGO FUENTE</b></p>   |     |
| 3.6.6 Proyecto de formas de Windows con un arreglo bidimensional: Ventas   | 91  |
| <p>Apoyo en la  <b>PROGRAMA 3.6 CÓDIGO FUENTE</b></p>   |     |
| 3.7 Arreglos tridimensionales  | 99  |
| 3.7.1 Representación de arreglos tridimensionales  | 99  |
| 3.7.2 Operaciones con arreglos tridimensionales  | 101 |
| 3.7.3 Ejemplos de arreglos tridimensionales  | 101 |
| 3.7.4 Definición de una clase para un arreglo tridimensional que almacene calificaciones   | 101 |
| 3.7.5 Programa 3.7. Proyecto de consola en C# con un arreglo tridimensional: Calificaciones  | 107 |
| <p>Apoyo en la  <b>PROGRAMA 3.7 CÓDIGO FUENTE</b></p> |     |
| 3.7.6 Programa 3.8. Proyecto de formas de Windows con un arreglo tridimensional: Ventas  | 113 |
| <p>Apoyo en la  <b>PROGRAMA 3.8 CÓDIGO FUENTE</b></p> |     |
| 3.8 Aplicación de arreglos unidimensionales usando la clase genérica ArrayList   | 123 |
| 3.8.1 Principales métodos y propiedades de la clase genérica ArrayList   | 123 |
| 3.8.2 Programa 3.9. Proyecto en C# de una aplicación de consola utilizando la clase genérica ArrayList                                   | 123 |

Apoyo en la



**PROGRAMA 3.9 CÓDIGO FUENTE**

Apoyo en la



**CUESTIONARIO LECTURA COMPLEMENTARIA**

Apoyo en la



**EJERCICIOS CON ARREGLOS**

**UNIIDIMENSIONALES LECTURA COMPLEMENTARIA**

Apoyo en la



**EJERCICIOS CON ARREGLOS**

**BIDIMENSIONALES LECTURA COMPLEMENTARIA**

Apoyo en la



**EJERCICIOS CON ARREGLOS TRIDIMENSIONALES**

LECTURA COMPLEMENTARIA

**CAPÍTULO 4  
PILAS**

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

Apoyo en la



**PROGRAMA 4.1 CÓDIGO FUENTE**

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

Apoyo en la



**PROGRAMA 4.2 CÓDIGO FUENTE**

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

Apoyo en la



**CUESTIONARIO LECTURA COMPLEMENTARIA**

Apoyo en la

**EJERCICIOS LECTURA COMPLEMENTARIA****CAPÍTULO 5  
COLAS**

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

Apoyo en la

**PROGRAMA 5.1 CÓDIGO FUENTE**

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

Apoyo en la

**PROGRAMA 5.2 CÓDIGO FUENTE**

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

Apoyo en la

**PROGRAMA 5.3 CÓDIGO FUENTE**

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

Apoyo en la

**PROGRAMA 5.4 CÓDIGO FUENTE**

|               |   |     |
|---------------|---|-----|
| <b>5.10.3</b> | Programa 5.5. Proyecto de formas de Windows<br>en C# con una cola circular: marco fotográfico digital | 174 |
|---------------|---|-----|

Apoyo en la

**PROGRAMA 5.5 CÓDIGO FUENTE**

Apoyo en la



**QUESTIONARIO** LECTURA COMPLEMENTARIA

Apoyo en la



**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 |

Apoyo en la



**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 |

Apoyo en la



**PROGRAMA 6.2** CÓDIGO 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 |

Apoyo en la



**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 |

Apoyo en la



**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 |
|---|-----|

Apoyo en la



**PROGRAMA 6.5** CÓDIGO FUENTE

|               |   |     |
|---------------|---|-----|
| <b>6.10</b>   | Implementación de una cola mediante una lista simple                      | 250 |
| <b>6.10.1</b> | Representación de una cola mediante una lista simple (Cola-Lista)         | 250 |
| <b>6.10.2</b> | Operaciones básicas con una Cola-Lista                                    | 251 |
| <b>6.10.3</b> | Situaciones críticas al trabajar con una Cola-Lista                       | 251 |
| <b>6.10.4</b> | Definición de la clase para una Cola-Lista                                | 252 |
| <b>6.10.5</b> | Programa 6.6. Proyecto de consola en C# con una Cola-Lista: Datos enteros | 256 |

Apoyo en la

**PROGRAMA 6.6 CÓDIGO FUENTE**

|               |  |     |
|---------------|--|-----|
| <b>6.11</b>   | Listas enlazadas dobles  | 263 |
| <b>6.11.1</b> | Representación de listas dobles  | 263 |
| <b>6.11.2</b> | Operaciones y situaciones críticas que se pueden presentar al trabajar con listas dobles | 264 |
| <b>6.11.3</b> | Ejemplo de listas dobles   | 269 |
| <b>6.11.4</b> | Definición de las clases para una lista Doble con nodos ordenados                        | 270 |
| <b>6.12</b>   | Programa 6.7 Proyecto de consola en C# con una lista doble ordenada: Datos enteros       | 280 |

Apoyo en la

**PROGRAMA 6.7 CÓDIGO FUENTE**

|               |   |     |
|---------------|---|-----|
| <b>6.13</b>   | Programa 6.8 Proyecto de formas de Windows con una lista doble desordenada: Ferrocarril | 290 |
| <b>6.13.1</b> | Agregar un vagón a la lista doble   | 291 |
| <b>6.13.2</b> | Eliminar un vagón de la lista doble   | 292 |
| <b>6.13.3</b> | Mostrar desde el principio y desde el final   | 293 |
| <b>6.13.4</b> | Buscar vagón  | 293 |
| <b>6.13.5</b> | Definición de las clases de la lista doble para almacenar los datos del ferrocarril     | 293 |

Apoyo en la

**PROGRAMA 6.8 CÓDIGO FUENTE**

|               |  |     |
|---------------|--|-----|
| <b>6.14</b>   | Aplicación de listas dobles usando las clases genéricas LinkedList y LinkedListNode                                    | 307 |
| <b>6.14.1</b> | Principales métodos y propiedades de las clases genéricas LinkedList y LinkedListNode                                  | 307 |
| <b>6.14.2</b> | Programa 6.9. Proyecto de consola en C# con una aplicación utilizando las clases genéricas LinkedList y LinkedListNode | 308 |

Apoyo en la

**PROGRAMA 6.9 CÓDIGO FUENTE**

Apoyo en la

**CUESTIONARIO LECTURA COMPLEMENTARIA**



**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 |

Apoyo en la



**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 |

Apoyo en la



**GRAPHVIZ Y EL LENGUAJE DOT LECTURA COMPLEMENTARIA**

Apoyo en la



**PROGRAMA 8.2 CÓDIGO FUENTE**

Apoyo en la



**CUESTIONARIO LECTURA COMPLEMENTARIA**

Apoyo en la



**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 |

Apoyo en la




**PROGRAMA 9.1 CÓDIGO FUENTE**


Apoyo en la



**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 |
|  <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 |






**PROGRAMA 9.4 CÓDIGO FUENTE**


**CUESTIONARIO LECTURA COMPLEMENTARIA**


**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 |

|   |   |            |
|---|---|------------|
| 10.5.4  | Métodos de ordenamiento interno de árbol  | 457        |
| 10.5.5  | Definición de las clases para los métodos de ordenamiento interno   | 462        |
| 10.5.6  | Consideraciones importantes para la implementación de las clases  | 463        |
| 10.5.7  | Programa 10.1 Proyecto de consola en C# con métodos de ordenamiento interno: datos enteros                  | 464        |
| <p>Apoyo en la  <b>PROGRAMA 10.1 CÓDIGO FUENTE</b></p>         |   |            |
| 10.5.8  | Programa 10.2 Proyecto de formas de Windows en C# con métodos de ordenamiento interno: datos de estudiantes | 469        |
| <p>Apoyo en la  <b>PROGRAMA 10.2 CÓDIGO FUENTE</b></p>         |   |            |
| 10.5.9  | Análisis de eficiencia de los métodos de ordenamiento interno   | 474        |
| <p>Apoyo en la  <b>CUESTIONARIO LECTURA COMPLEMENTARIA</b></p> |   |            |
| <p>Apoyo en la  <b>EJERCICIOS LECTURA COMPLEMENTARIA</b></p>   |   |            |
| <b>Bibliografía</b>   |   | <b>483</b> |
| <b>Índice analítico</b>   |   | <b>485</b> |