

# Índice general

|   |              |
|---|--------------|
| <b>Agradecimientos</b>  | <b>v</b>     |
| <b>Prefacio</b>   | <b>XIX</b>   |
| <b>Convenios tipográficos</b>                                   | <b>XXIII</b> |
| <b>1. Introducción al lenguaje C++</b>                          | <b>1</b>     |
| 1.1. Historia . . . . .   | 1            |
| 1.2. El paradigma de la orientación a objetos . . . . .         | 4            |
| 1.2.1. Elementos fundamentales . . . . .                        | 4            |
| 1.2.2. Objetos . . . . .  | 9            |
| 1.2.3. Clases . . . . .   | 12           |
| 1.2.4. Relaciones entre clases . . . . .                        | 13           |
| 1.3. El entorno de desarrollo . . . . .                         | 16           |
| 1.3.1. Cómo dar la orden . . . . .                              | 17           |
| 1.3.2. Pasos en la traducción . . . . .                         | 18           |
| 1.3.3. Un primer programa . . . . .                             | 21           |
| 1.4. Diferencias con C . . . . .                                | 25           |
| 1.4.1. Palabras reservadas . . . . .                            | 25           |
| 1.4.2. Cabeceras estándar . . . . .                             | 25           |
| 1.4.3. Tamaño de las constantes literales de carácter . . . . . | 26           |
| 1.4.4. Entrada y salida de datos . . . . .                      | 27           |
| 1.4.5. Comentarios . . . . .                                    | 29           |

---

|           |  |           |
|-----------|--|-----------|
| 1.4.6.    | Empleo de macros . . . . .                         | 30        |
| 1.4.7.    | Declaraciones . . . . .                            | 32        |
| 1.4.8.    | Definiciones de funciones . . . . .                | 37        |
| 1.4.9.    | Uniones anónimas . . . . .                         | 38        |
| 1.4.10.   | Estilo . . . . .                                   | 39        |
| 1.5.      | Recorrido por el lenguaje . . . . .                | 39        |
| 1.5.1.    | Espacios de nombres . . . . .                      | 39        |
| 1.5.2.    | Referencias . . . . .                              | 41        |
| 1.5.3.    | Gestión de la memoria dinámica . . . . .           | 41        |
| 1.5.4.    | Sobrecarga . . . . .                               | 42        |
| 1.5.5.    | Clases . . . . .                                   | 44        |
| 1.5.6.    | Plantillas . . . . .                               | 46        |
| 1.5.7.    | Excepciones . . . . .                              | 51        |
|           | Ejercicios . . . . .                               | 55        |
| <b>2.</b> | <b>Tipos, operadores y expresiones</b>             | <b>57</b> |
| 2.1.      | Introducción . . . . .                             | 57        |
| 2.2.      | Constantes, variables y referencias . . . . .      | 57        |
| 2.2.1.    | Declaración, definición e inicialización . . . . . | 58        |
| 2.2.2.    | Especificación de almacenamiento . . . . .         | 59        |
| 2.2.3.    | Constantes . . . . .                               | 65        |
| 2.2.4.    | Volátiles . . . . .                                | 70        |
| 2.2.5.    | Referencias . . . . .                              | 70        |
| 2.3.      | Tipos de datos . . . . .                           | 75        |
| 2.3.1.    | Conceptos previos . . . . .                        | 75        |
| 2.3.2.    | Tipos de datos incorporados . . . . .              | 76        |
| 2.3.3.    | Cadenas de caracteres: <i>string</i> . . . . .     | 97        |
| 2.3.4.    | Vectores de alto nivel: <i>vector</i> . . . . .    | 105       |
| 2.4.      | Operadores . . . . .                               | 110       |

|  |            |
|--|------------|
| 2.4.1. Cuadro resumen de operadores . . . . .                        | 112        |
| 2.4.2. Operadores de asignación . . . . .                            | 116        |
| 2.4.3. Operadores matemáticos . . . . .                              | 117        |
| 2.4.4. Incremento y decremento . . . . .                             | 118        |
| 2.4.5. Operadores relacionales . . . . .                             | 118        |
| 2.4.6. Operadores lógicos . . . . .                                  | 118        |
| 2.4.7. Operadores de bits . . . . .                                  | 119        |
| 2.4.8. Código en ensamblador . . . . .                               | 120        |
| 2.4.9. Operador condicional . . . . .                                | 122        |
| 2.4.10. Operadores de dirección e indirección . . . . .              | 123        |
| 2.4.11. Operador de tamaño . . . . .                                 | 123        |
| 2.4.12. Operador secuencia o de evaluación . . . . .                 | 124        |
| 2.4.13. Operadores de conversión . . . . .                           | 125        |
| 2.4.14. Operadores de memoria dinámica . . . . .                     | 129        |
| 2.5. Expresiones . . . . .   | 138        |
| Ejercicios . . . . .   | 141        |
| <b>3. Instrucciones de control</b>                                   | <b>145</b> |
| 3.1. Introducción . . . . .  | 145        |
| 3.2. Instrucciones y bloques . . . . .                               | 145        |
| 3.3. Instrucciones condicionales . . . . .                           | 146        |
| 3.3.1. Instrucción condicional simple . . . . .                      | 147        |
| 3.3.2. Instrucción condicional compuesta . . . . .                   | 148        |
| 3.3.3. Instrucción de decisión múltiple . . . . .                    | 150        |
| 3.3.4. Definiciones en las condiciones . . . . .                     | 152        |
| 3.4. Instrucciones de iteración . . . . .                            | 153        |
| 3.4.1. Instrucción <i>for</i> ( <i>para</i> ) . . . . .              | 153        |
| 3.4.2. Instrucción <i>while</i> ( <i>mientras</i> ) . . . . .        | 155        |
| 3.4.3. Instrucción <i>do-while</i> ( <i>haz-mientras</i> ) . . . . . | 156        |

|   |            |
|---|------------|
| 3.5. Instrucciones de ruptura de control . . . . .                      | 157        |
| 3.5.1. Instrucción <code>break</code> ( <i>ruptura</i> ) . . . . .      | 157        |
| 3.5.2. Instrucción <code>continue</code> ( <i>continuar</i> ) . . . . . | 158        |
| 3.5.3. Instrucción <code>goto</code> ( <i>ir a</i> ) . . . . .          | 158        |
| 3.6. Funciones . . . . .  | 159        |
| 3.6.1. Declaración: prototipo . . . . .                                 | 159        |
| 3.6.2. Definición . . . . .   | 161        |
| 3.6.3. Uso de una función . . . . .                                     | 164        |
| 3.6.4. La función <code>main()</code> . . . . .                         | 165        |
| 3.6.5. Sustitución «en línea» . . . . .                                 | 166        |
| 3.6.6. Parámetros predeterminados . . . . .                             | 168        |
| 3.6.7. Sobrecarga . . . . .   | 170        |
| 3.6.8. Recursividad . . . . .   | 176        |
| Ejercicios . . . . .  | 181        |
| <br>  |            |
| <b>4. Clases y objetos</b> . . . . .                                    | <b>183</b> |
| 4.1. Introducción . . . . .   | 183        |
| 4.2. Estructura de una clase . . . . .                                  | 183        |
| 4.2.1. Control de acceso . . . . .                                      | 184        |
| 4.2.2. Miembros de datos y funciones miembro . . . . .                  | 186        |
| 4.2.3. Miembros estáticos . . . . .                                     | 188        |
| 4.2.4. Sobrecarga de funciones miembro . . . . .                        | 191        |
| 4.2.5. Clases y funciones amigas . . . . .                              | 196        |
| 4.3. Objetos . . . . .  | 197        |
| 4.3.1. El puntero <code>this</code> . . . . .                           | 197        |
| 4.3.2. Construcción y destrucción . . . . .                             | 201        |
| 4.3.3. El constructor de copia . . . . .                                | 210        |
| 4.3.4. Sobrecarga del operador de asignación . . . . .                  | 212        |
| 4.3.5. Conversiones . . . . .   | 217        |

---

|   |            |
|---|------------|
| 4.4. Relaciones entre clases . . . . .                          | 225        |
| 4.4.1. Dependencias . . . . .                                   | 226        |
| 4.4.2. Asociaciones . . . . .                                   | 226        |
| 4.4.3. Generalizaciones y especializaciones . . . . .           | 239        |
| 4.4.4. Realizaciones . . . . .                                  | 245        |
| 4.5. Diseño de las relaciones entre clases . . . . .            | 246        |
| 4.5.1. Diseño de las dependencias . . . . .                     | 246        |
| 4.5.2. Diseño de las asociaciones . . . . .                     | 246        |
| 4.5.3. Diseño de las generalizaciones . . . . .                 | 249        |
| 4.5.4. Diseño de las realizaciones . . . . .                    | 251        |
| 4.6. Implementación de las relaciones en C++ . . . . .          | 251        |
| 4.6.1. Implementación de las asociaciones . . . . .             | 251        |
| 4.6.2. Implementación de las agregaciones . . . . .             | 260        |
| 4.6.3. Implementación de las generalizaciones . . . . .         | 261        |
| 4.7. Polimorfismo en tiempo de ejecución . . . . .              | 275        |
| 4.7.1. Identificación de tipos en tiempo de ejecución . . . . . | 286        |
| 4.8. Plantillas . . . . .                                       | 289        |
| 4.8.1. Clases paramétricas . . . . .                            | 289        |
| 4.8.2. Parámetros de plantilla . . . . .                        | 301        |
| 4.8.3. Omisión de parámetros de plantilla . . . . .             | 302        |
| 4.8.4. Exportación . . . . .                                    | 302        |
| Ejercicios . . . . .  | 305        |
| <b>5. Excepciones</b>   | <b>313</b> |
| 5.1. Introducción . . . . .                                     | 313        |
| 5.2. Lanzamiento de excepciones . . . . .                       | 314        |
| 5.2.1. Excepciones . . . . .                                    | 315        |
| 5.2.2. La instrucción <code>throw</code> . . . . .              | 315        |
| 5.2.3. La lista <code>throw</code> . . . . .                    | 318        |

---

|  |            |
|--|------------|
| 5.3. Gestión de excepciones . . . . .                            | 320        |
| 5.3.1. La instrucción <code>try</code> . . . . .                 | 320        |
| 5.3.2. La instrucción <code>catch</code> . . . . .               | 320        |
| 5.3.3. La función <code>terminate()</code> . . . . .             | 324        |
| 5.3.4. La función <code>unexpected()</code> . . . . .            | 326        |
| 5.4. Jerarquía de excepciones . . . . .                          | 328        |
| 5.4.1. Excepciones estándares . . . . .                          | 331        |
| 5.5. Programación con excepciones . . . . .                      | 333        |
| 5.5.1. Evitar excepciones . . . . .                              | 333        |
| 5.5.2. Empleo de excepciones . . . . .                           | 334        |
| Ejercicios . . . . .   | 339        |
| <b>6. La Biblioteca Estándar de E/S (IOStreams)</b>              | <b>341</b> |
| 6.1. La Biblioteca Estándar de C++ . . . . .                     | 341        |
| 6.2. La Biblioteca Estándar de E/S . . . . .                     | 344        |
| 6.3. Flujos de salida . . . . .                                  | 347        |
| 6.3.1. El operador de inserción <code>&lt;&lt;</code> . . . . .  | 349        |
| 6.3.2. Salida formateada . . . . .                               | 350        |
| 6.3.3. Salida de tipos definidos por el usuario . . . . .        | 356        |
| 6.4. Flujos de entrada . . . . .                                 | 357        |
| 6.4.1. El operador de extracción <code>&gt;&gt;</code> . . . . . | 359        |
| 6.4.2. Entrada de caracteres . . . . .                           | 361        |
| 6.4.3. Entrada de tipos definidos por el usuario . . . . .       | 362        |
| 6.5. Funciones virtuales de E/S . . . . .                        | 363        |
| 6.6. Flujos de fichero . . . . .                                 | 364        |
| 6.6.1. Apertura . . . . .  | 364        |
| 6.6.2. Cierre . . . . .  | 366        |
| 6.6.3. Acceso directo . . . . .                                  | 368        |
| 6.7. Flujos de cadena . . . . .                                  | 369        |

---

|   |            |
|---|------------|
| 6.8. Empleo de los estados de los flujos . . . . .                  | 371        |
| 6.9. Mezcla de las bibliotecas de C y de C++ . . . . .              | 373        |
| 6.10. Conclusión . . . . .  | 374        |
| Ejercicios . . . . .  | 377        |
| <b>7. La Biblioteca Estándar de Plantillas (STL)</b>                | <b>379</b> |
| 7.1. Introducción . . . . .   | 379        |
| 7.2. Contenedores e iteradores . . . . .                            | 380        |
| 7.2.1. Contenedores . . . . .                                       | 380        |
| 7.2.2. Iteradores . . . . .   | 381        |
| 7.3. Secuencias . . . . .   | 388        |
| 7.3.1. Vectores . . . . .   | 390        |
| 7.3.2. Colas dobles . . . . .                                       | 392        |
| 7.3.3. Listas . . . . .   | 393        |
| 7.4. Adaptadores de secuencia . . . . .                             | 394        |
| 7.5. Contenedores asociativos (ordenados) . . . . .                 | 398        |
| 7.5.1. Aplicaciones: <i>map</i> . . . . .                           | 405        |
| 7.5.2. Aplicaciones con claves repetidas: <i>multimap</i> . . . . . | 407        |
| 7.5.3. Conjuntos: <i>set</i> . . . . .                              | 411        |
| 7.5.4. Multiconjuntos: <i>multiset</i> . . . . .                    | 414        |
| 7.5.5. Resumen de contenedores asociativos ordenados . . . . .      | 415        |
| 7.6. Manejo de bits . . . . .                                       | 415        |
| 7.6.1. Conjunto de bits: <i>bitset&lt;N&gt;</i> . . . . .           | 418        |
| 7.6.2. Vector de booleanos: <i>vector&lt;bool&gt;</i> . . . . .     | 421        |
| 7.7. Objetos función . . . . .                                      | 423        |
| 7.7.1. Clasificación . . . . .                                      | 424        |
| 7.7.2. Objetos función predefinidos . . . . .                       | 428        |
| 7.7.3. Adaptadores de objetos función . . . . .                     | 430        |
| 7.8. Algoritmos . . . . .   | 432        |

---

|   |            |
|---|------------|
| 7.8.1. Algoritmos observadores elementales . . . . .          | 432        |
| 7.8.2. Algoritmos modificadores elementales . . . . .         | 436        |
| 7.8.3. Algoritmos numéricos generalizados . . . . .           | 443        |
| 7.8.4. Ordenación y operaciones en rangos ordenados . . . . . | 445        |
| 7.8.5. Montículos . . . . .                                   | 449        |
| Ejercicios . . . . .  | 453        |
| <b>A. El código ISO-8859-1</b>                                | <b>455</b> |
| <b>Colofón</b>  | <b>457</b> |
| <b>Bibliografía</b>   | <b>459</b> |

# Índice de cuadros

|   |     |
|---|-----|
| 1.1. Palabras reservadas de C++ . . . . .   | 26  |
| 1.2. Palabras reservadas que son macros o tipos definidos en cabeceras de C . . . . . | 26  |
| 2.1. Secuencias de escape . . . . .   | 78  |
| 2.2. Resumen de operadores . . . . .  | 114 |
| 2.3. Palabras reservadas alternativas . . . . .                                       | 115 |
| 2.4. Dígrafos y trígrafos . . . . .   | 115 |
| 4.1. Cambio de los privilegios de acceso durante la herencia . . . . .                | 263 |
| 5.1. Excepciones estándares lanzadas por el lenguaje C++ . . . . .                    | 332 |
| 5.2. Excepciones estándares lanzadas por la biblioteca estándar . . . . .             | 332 |
| 6.1. Contenedores . . . . .   | 343 |
| 6.2. Utilidades generales . . . . .   | 343 |
| 6.3. Iteradores . . . . .   | 343 |
| 6.4. Algoritmos . . . . .   | 344 |
| 6.5. Diagnóstico . . . . .  | 344 |
| 6.6. Cadenas . . . . .  | 344 |
| 6.7. Entrada/Salida . . . . .   | 345 |
| 6.8. Localización . . . . .   | 345 |
| 6.9. Utilidades de apoyo . . . . .  | 346 |
| 6.10. Números . . . . .   | 346 |

---

|  |     |
|--|-----|
| 6.11. Manipuladores en <code>&lt;ios&gt;</code> , <code>&lt;ostream&gt;</code> e <code>&lt;iostream&gt;</code> . . . . . | 354 |
| 6.12. Manipuladores en <code>&lt;iomanip&gt;</code> . . . . .  | 354 |
| 6.13. Nombres de los <i>bits</i> de formato . . . . .  | 355 |
| 6.14. Modos de apertura de ficheros . . . . .  | 365 |
| 6.15. Valores de la enumeración <i>seek_dir</i> . . . . .  | 368 |
| 7.1. Contenedores de la STL . . . . .  | 381 |
| 7.2. Miembros de tipo . . . . .  | 382 |
| 7.3. Funciones miembro de tamaño . . . . .   | 382 |
| 7.4. Funciones miembro de construcción de iteradores . . . . .   | 383 |
| 7.5. Adaptadores de secuencia de la STL . . . . .  | 395 |
| 7.6. Tipos en contenedores asociativos . . . . .   | 415 |
| 7.7. Constructores de contenedores asociativos . . . . .   | 416 |
| 7.8. Inserción y borrado . . . . .   | 416 |
| 7.9. Otros métodos . . . . .   | 417 |
| 7.10. <i>bitset</i> : Operadores de bits con asignación . . . . .  | 421 |
| 7.11. Otros operadores de bits (miembros de <i>bitset</i> ) . . . . .  | 421 |
| 7.12. Otros operadores de bits (externos a la clase <i>bitset</i> ) . . . . .  | 421 |
| 7.13. Métodos de <i>bitset</i> para manipulación de bits . . . . .   | 422 |
| 7.14. Otras operaciones de <i>bitset</i> . . . . .   | 422 |
| 7.15. Clases de predicados de comparación estándar . . . . .   | 429 |
| 7.16. Clases de predicados lógicos estándar . . . . .  | 429 |
| 7.17. Clases de objetos función aritméticos estándar . . . . .   | 429 |
| 7.18. Adaptadores, negadores y ligadores estándar . . . . .  | 431 |

# Índice de figuras

|   |     |
|---|-----|
| 1.1. Los creadores de C y C++ . . . . .                           | 2   |
| 1.2. Representación de un objeto . . . . .                        | 9   |
| 1.3. Representación de objetos relacionados entre sí . . . . .    | 11  |
| 1.4. Categorías de objetos según su comportamiento . . . . .      | 12  |
| 1.5. Representación de los flujos de control y de datos . . . . . | 12  |
| 1.6. Representación de una clase . . . . .                        | 13  |
| 1.7. Relaciones entre clases . . . . .                            | 14  |
| 1.8. Pasos en la traducción de un programa . . . . .              | 19  |
|   |     |
| 4.1. Dependencias . . . . .                                       | 227 |
| 4.2. Asociaciones y enlaces . . . . .                             | 227 |
| 4.3. Asociación ternaria . . . . .                                | 228 |
| 4.4. Asociación ternaria como clase estereotipada . . . . .       | 229 |
| 4.5. Asociación uno a varios . . . . .                            | 230 |
| 4.6. Diagramas con multiplicidades . . . . .                      | 230 |
| 4.7. Funciones en una asociación . . . . .                        | 231 |
| 4.8. Funciones necesarias en una asociación . . . . .             | 232 |
| 4.9. Ejemplo de confusión entre asociación y mensajes . . . . .   | 232 |
| 4.10. Agregación . . . . .  | 232 |
| 4.11. Ejemplo de agregado múltiple . . . . .                      | 233 |
| 4.12. Agregación fija . . . . .                                   | 234 |
| 4.13. Agregación variable . . . . .                               | 234 |

---

|  |     |
|--|-----|
| 4.14. Agregación recursiva . . . . .                                     | 234 |
| 4.15. Representación gráfica de la composición . . . . .                 | 235 |
| 4.16. Representación de la navegabilidad de una asociación . . . . .     | 236 |
| 4.17. Atributos de enlace . . . . .                                      | 236 |
| 4.18. Sin atributos de enlace . . . . .                                  | 237 |
| 4.19. No todas las personas pueden dar clase . . . . .                   | 238 |
| 4.20. Atributos de enlace . . . . .                                      | 238 |
| 4.21. Clase de asociación . . . . .                                      | 239 |
| 4.22. Asociación calificada . . . . .                                    | 240 |
| 4.23. Ejemplo de generalización . . . . .                                | 241 |
| 4.24. Herencia múltiple . . . . .  | 242 |
| 4.25. Ejemplo de generalización según criterios independientes . . . . . | 242 |
| 4.26. ¿Heredar o asociar? . . . . .                                      | 243 |
| 4.27. Interfaces . . . . .   | 245 |
| 5.1. Jerarquía de excepciones estándares . . . . .                       | 333 |
| 6.1. Jerarquía de clases de flujos . . . . .                             | 347 |

# Prefacio

Este libro explica el lenguaje de programación C++ según la norma aprobada hace unos pocos años por el esfuerzo conjunto de los organismos de normalización ANSI e ISO. Lo hemos escrito pensando en nuestros alumnos de Programación Orientada a Objetos, asignatura de segundo curso de las Ingenierías Técnicas en Informática en la Universidad de Cádiz. Ellos ya han visto el lenguaje C en primer curso y por tanto sólo necesitan un breve repaso. Esto nos viene bien porque C++ es un lenguaje con muchas características y si ya se conocen las básicas incluidas en C se tiene algo ganado; por lo tanto nuestro libro parte de la base de que ya se poseen conocimientos de C, aunque también es posible utilizarlo sin tenerlos puesto que de todas formas no se omite la parte básica de C++, es decir la parte relativa a la programación estructurada que comparte con su antecesor.

A pesar de utilizarse en clases de prácticas de dos horas, el libro no está dividido en «prácticas» sino en capítulos extensos, y no hay que pensar que cada capítulo pueda estudiarse en dos horas. Si se usa para clases prácticas, es criterio del profesor hasta dónde llegar en cada capítulo, dependiendo de sus alumnos y de otras circunstancias. Nosotros estimamos que unas quince prácticas de dos horas cada una es el *mínimo* suficiente.

Aunque C++ admita varios paradigmas de programación, el empleo principal de este lenguaje es en la programación orientada a objetos (P.O.O.); se recomienda también por tanto para aprovechar al máximo la lectura, el conocimiento previo de alguna metodología orientada a objetos y de P.O.O. en general; si bien no es imprescindible este conocimiento pues se dan nociones generales básicas sobre la P.O.O.

El libro se divide en siete capítulos. A continuación se detallan sus contenidos.

1. Se empieza explicando cuándo, cómo y por qué surgió este lenguaje, lo que entronca con un breve resumen del paradigma de la P.O.O. A continuación, para pasar ya rápidamente a la práctica, se habla del proceso de la compilación de un programa; esta parte está ligeramente sesgada a entornos UNIX no gráficos porque pensamos que si no es la