

# Índice general



<b>1</b>	<b>Análisis de la eficiencia</b> .....	<b>1</b>
<b>1.1</b>	<b>Introducción</b> .....	<b>1</b>
1.1.1	Tamaño del problema .....	1
1.1.2	Algoritmos vs implementaciones .....	2
<b>1.2</b>	<b>Eficiencia de algoritmos</b> .....	<b>2</b>
1.2.1	Familias de órdenes de eficiencia .....	2
1.2.2	Notación asintótica .....	3
1.2.3	Eficiencia en tiempo y espacio .....	9
1.2.4	Elección del mejor algoritmo .....	9
<b>1.3</b>	<b>Análisis de algoritmos</b> .....	<b>10</b>
1.3.1	Operación elemental .....	10
1.3.2	Caso peor, caso promedio y análisis amortizado .....	11
1.3.3	Reglas para el cálculo de la eficiencia .....	14
<b>1.4</b>	<b>Ejemplos</b> .....	<b>17</b>
1.4.1	Algoritmo de multiplicación de matrices .....	18
1.4.2	Algoritmo de búsqueda binaria .....	18
1.4.3	Algoritmo de ordenación por selección .....	19
1.4.4	Mejor caso de ordenación por inserción .....	21
1.4.5	Algoritmo de mezcla de dos vectores .....	21
1.4.6	Algoritmo de ordenación por mezcla .....	22
<b>1.5</b>	<b>Problemas</b> .....	<b>24</b>

<b>2</b>	<b>Encapsulación</b>	<b>27</b>
<b>2.1</b>	<b>Abstracción</b>	<b>27</b>
2.1.1	Mecanismos de abstracción	28
<b>2.2</b>	<b>C++98 vs C++11/14</b>	<b>28</b>
2.2.1	Inicialización homogénea con llaves	29
2.2.2	Deducción del tipo con auto	32
2.2.3	Semántica de movimiento	33
<b>2.3</b>	<b>Clases en C++</b>	<b>40</b>
2.3.1	Abstracción e implementación de un vector dinámico	40
2.3.2	Ocultar o no ocultar	44
2.3.3	Funciones miembro especiales	46
2.3.4	Generación o eliminación explícita con <i>delete</i>	51
<b>2.4</b>	<b>Ejemplos</b>	<b>54</b>
2.4.1	El tipo <i>Vector</i>	55
2.4.2	El tipo <i>Conjunto</i>	58
2.4.3	El tipo <i>Polinomio</i>	61
<b>2.5</b>	<b>Problemas</b>	<b>63</b>
<b>3</b>	<b>Generalización: Plantillas</b>	<b>65</b>
<b>3.1</b>	<b>Introducción</b>	<b>65</b>
<b>3.2</b>	<b>Funciones plantilla en C++</b>	<b>66</b>
3.2.1	Tipo genérico de la plantilla: <i>typename</i> y <i>class</i>	67
3.2.2	Encadenar llamadas de funciones plantilla	67
3.2.3	Especificación explícita del tipo <i>T</i>	68
<b>3.3</b>	<b>Clases plantilla</b>	<b>69</b>
3.3.1	Definición de los métodos de la clase	70
3.3.2	Tipos asociados	72
3.3.3	Funciones miembro plantilla	75
3.3.4	Funciones y clases plantilla amigas de una clase plantilla	76
<b>3.4</b>	<b>Plantillas y compilación separada</b>	<b>77</b>
3.4.1	Inclusión de las definiciones	79
3.4.2	Instanciación explícita	83
3.4.3	Modelo mixto de inclusión e instanciación explícita	85
3.4.4	Compilación separada con <i>export</i>	89
<b>3.5</b>	<b>Compatibilidad del tipo base en la instanciación</b>	<b>89</b>
3.5.1	Conceptos	90
<b>3.6</b>	<b>Número y tipo de parámetros plantilla</b>	<b>92</b>
3.6.1	Número de parámetros	92
3.6.2	Tipo de parámetros	94
3.6.3	Valores por defecto	97
3.6.4	Especializaciones	98

<b>3.7</b>	<b>Otros aspectos relacionados con C++14</b>	<b>100</b>
3.7.1	Intercambiar y máximo en la biblioteca estándar	100
3.7.2	Constructores y miembros <i>constexpr</i>	102
3.7.3	Alias con <i>using</i>	103
3.7.4	Eliminando alternativas con <i>delete</i>	104
3.7.5	Listas de inicialización	105
3.7.6	Constructor delegado	107
3.7.7	Literales definidos por el usuario	108
<b>3.8</b>	<b>Problemas</b>	<b>109</b>
<b>4</b>	<b>Contenedores, iteradores y algoritmos genéricos</b>	<b>111</b>
<b>4.1</b>	<b>Introducción</b>	<b>111</b>
<b>4.2</b>	<b>Posición: Un primer diseño</b>	<b>112</b>
4.2.1	Acceso con índices	113
4.2.2	Abstracción de la posición	117
<b>4.3</b>	<b>Iteradores: conceptos básicos con vectores-C</b>	<b>123</b>
4.3.1	Punteros e iteración	123
4.3.2	Iteración de sólo lectura: <i>const</i>	125
4.3.3	Conversión de iteradores	127
<b>4.4</b>	<b>Diseño de iteradores para contenedores</b>	<b>128</b>
4.4.1	Iteración sobre <i>Vector</i> y <i>Conjunto</i>	128
4.4.2	Contenedores no modificables con iteradores	130
4.4.3	Invalidación de iteradores	131
4.4.4	Tipos de iteradores	132
<b>4.5</b>	<b>Algoritmos genéricos</b>	<b>135</b>
4.5.1	Rangos con iteradores	137
4.5.2	Características de tipos: <i>Type-traits</i>	141
4.5.3	Conceptos e iteradores	148
<b>4.6</b>	<b>Algunos aspectos relacionados con C++14</b>	<b>148</b>
4.6.1	Simplificar con <i>auto</i>	148
4.6.2	Bucles basados en rango	149
<b>4.7</b>	<b>Problemas</b>	<b>150</b>
<b>5</b>	<b>Objetos llamables</b>	<b>153</b>
<b>5.1</b>	<b>Introducción</b>	<b>153</b>
5.1.1	Punteros a función	154
5.1.2	Punteros a miembro	154
<b>5.2</b>	<b>Objetos función</b>	<b>156</b>
5.2.1	Objetos función con estado	157
5.2.2	Parámetros de plantilla	160
5.2.3	Paso y devolución de objetos función	166
5.2.4	Conceptos y objetos función	168
5.2.5	Objetos función de la biblioteca estándar	170

<b>5.3</b>	<b>Expresiones lambda</b>	<b>171</b>
5.3.1	Sintaxis de una expresión lambda . . . . .	172
5.3.2	Captura del entorno . . . . .	174
<b>5.4</b>	<b>Nuevas formas de definir funciones</b>	<b>179</b>
5.4.1	Uso de <i>decltype</i> . . . . .	180
5.4.2	Deducción automática del tipo devuelto . . . . .	182
5.4.3	Lambdas genéricos . . . . .	182
5.4.4	Deducción <i>auto</i> vs <i>decltype</i> . . . . .	183
<b>5.5</b>	<b>Utilidades de la biblioteca estándar</b>	<b>185</b>
5.5.1	Empaquetar un puntero a miembro en un <i>functor</i> . . . . .	185
5.5.2	Empaquetar funciones y argumentos: <i>std::bind</i> . . . . .	187
5.5.3	Un tipo para <i>gobernarlos</i> a todos: <i>std::function</i> . . . . .	190
<b>5.6</b>	<b>Problemas</b>	<b>193</b>
<b>6</b>	<b>Contenedores y algoritmos del estándar . . . . .</b>	<b>195</b>
<b>6.1</b>	<b>Introducción</b>	<b>195</b>
6.1.1	Claves de la referencia . . . . .	195
<b>6.2</b>	<b>Contenedores</b>	<b>198</b>
6.2.1	Tipos de contenedores . . . . .	199
6.2.2	Contenedores secuenciales . . . . .	201
6.2.3	Contenedores asociativos con orden . . . . .	205
6.2.4	Contenedores asociativos sin orden . . . . .	211
<b>6.3</b>	<b>Iteradores: de contenedores a algoritmos</b>	<b>217</b>
6.3.1	Contenedores y tipos de iteradores . . . . .	217
6.3.2	Iteradores: inserciones, borrados, búsquedas... . . . .	221
6.3.3	Funciones útiles para iteradores . . . . .	223
<b>6.4</b>	<b>Comparación de contenedores: operaciones</b>	<b>224</b>
6.4.1	Contenedor como concepto . . . . .	226
6.4.2	Contenedores secuenciales . . . . .	229
6.4.3	Contenedores asociativos . . . . .	231
6.4.4	Otras operaciones . . . . .	234
<b>6.5</b>	<b>Algoritmos</b>	<b>237</b>
6.5.1	Algoritmos de la biblioteca estándar de C++ . . . . .	238
<b>6.6</b>	<b>Problemas</b>	<b>250</b>
<b>7</b>	<b>Adaptadores . . . . .</b>	<b>253</b>
<b>7.1</b>	<b>Introducción</b>	<b>253</b>
<b>7.2</b>	<b>Adaptadores de contenedores</b>	<b>254</b>
7.2.1	El adaptador <i>stack</i> . . . . .	254
7.2.2	El adaptador <i>queue</i> . . . . .	256
7.2.3	El adaptador <i>priority_queue</i> . . . . .	258

<b>7.3</b>	<b>Adaptar a la interfaz de iterador</b>	<b>261</b>
7.3.1	Iteradores <i>istream/ostream</i> . . . . .	261
7.3.2	Iteradores <i>reverse</i> . . . . .	265
7.3.3	Iteradores de inserción . . . . .	267
7.3.4	Iteradores de movimiento . . . . .	270
7.3.5	Funciones para generación de adaptadores . . . . .	270
<b>7.4</b>	<b>Problemas</b>	<b>271</b>
<b>8</b>	<b>Metaprogramación</b> . . . . .	<b>273</b>
<b>8.1</b>	<b>Introducción</b>	<b>273</b>
<b>8.2</b>	<b>Metafunciones</b>	<b>274</b>
8.2.1	Metaprogramación y <i>plantillas</i> . . . . .	274
8.2.2	Metaprogramación y <i>constexpr</i> . . . . .	275
8.2.3	Devolución con plantillas: <i>value</i> y <i>type</i> . . . . .	276
<b>8.3</b>	<b>Plantillas... revisitado</b>	<b>279</b>
8.3.1	Deduciendo el tipo . . . . .	279
8.3.2	Referencias universales . . . . .	282
8.3.3	Paquetes de parámetros . . . . .	283
8.3.4	SFINAE . . . . .	287
<b>8.4</b>	<b>Condicionales y bucles</b>	<b>288</b>
8.4.1	Condicionales . . . . .	289
8.4.2	Bucles . . . . .	295
<b>A</b>	<b>Solución a los ejercicios</b> . . . . .	<b>299</b>
<b>A.1</b>	<b>Eficiencia</b>	<b>299</b>
<b>A.2</b>	<b>Encapsulación</b>	<b>302</b>
<b>A.3</b>	<b>Plantillas</b>	<b>306</b>
<b>A.4</b>	<b>Contenedores</b>	<b>309</b>
<b>A.5</b>	<b>Objetos llamables</b>	<b>314</b>
<b>A.6</b>	<b>Contenedores y algoritmos</b>	<b>320</b>
<b>A.7</b>	<b>Adaptadores</b>	<b>324</b>
<b>B</b>	<b>Tablas</b> . . . . .	<b>329</b>
<b>B.1</b>	<b>Tabla ASCII</b>	<b>329</b>
<b>B.2</b>	<b>Operadores C++</b>	<b>330</b>
<b>B.3</b>	<b>Palabras reservadas de C89, C99, C11, C++ y C++11</b>	<b>332</b>
<b>B.4</b>	<b>Referencia de contenedores</b>	<b>332</b>
B.4.1	Contenedores vs métodos . . . . .	333
B.4.2	Operaciones de los contenedores . . . . .	336

<b>B.5</b>	<b>Referencia de algoritmos</b>	<b>351</b>
B.5.1	Algoritmos clasificados . . . . .	351
B.5.2	Algoritmos por orden alfabético . . . . .	358
<b>B.6</b>	<b>Referencia de objetos función</b>	<b>365</b>
	<b>Bibliografía . . . . .</b>	<b>367</b>
	<b>Referencias electrónicas</b>	<b>370</b>
	<b>Índice alfabético . . . . .</b>	<b>371</b>