

46



Índice General

| | |
|--|------|
| Prefacio e Introducción | XI |
| Introducción | XI |
| ¿Por qué aparece ahora la orientación a objetos? | XIV |
| ¿Quién debe leer este libro? | XVI |
| ¿Cómo se debe utilizar este libro? | XVII |
| Agradecimientos | XX |

Primera parte

Los fundamentos de la orientación a objetos

| | |
|--|----|
| 1. Una visión del futuro | 3 |
| Introducción | 4 |
| Las fuerzas conductoras | 5 |
| Información multimedia | 6 |
| Computación de usuario final | 9 |
| Procesamiento distribuido | 10 |
| El futuro | 11 |
| El progreso reciente | 17 |
| El Macintosh de Apple | 17 |
| El computador NeXT | 19 |
| La arquitectura orientada a objetos en los años 90 | 21 |
| Resumen | 25 |
| 2. Los conceptos básicos | 29 |
| Mecanismos básicos | 31 |
| Objetos | 31 |
| Mensajes y métodos | 32 |
| Clases, subclases y objetos | 34 |



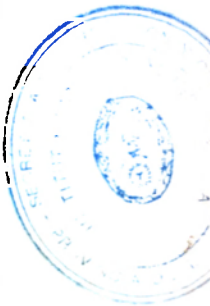
| | |
|---|-----------|
| Herencia | 36 |
| Conceptos clave | 38 |
| Encapsulación | 38 |
| Abstracción | 38 |
| Polimorfismo | 39 |
| Persistencia | 39 |
| Términos técnicos relacionados | 40 |
| Ligadura dinámica | 40 |
| Programación visual | 40 |
| Blob (<i>Binary Large Object</i>) | 41 |
| El método tradicional frente a la orientación a objetos | 41 |
| Resumen | 43 |
| 3. Las ventajas | 45 |
| Gestión de la complejidad | 47 |
| Flexibilidad en el desarrollo del software | 48 |
| Reutilización | 49 |
| Aumento de la productividad | 51 |
| Extensibilidad y mantenibilidad | 51 |
| Programación por el usuario | 52 |
| Resumen | 52 |

Segunda Parte

El panorama del software orientado a objetos

| | |
|---|-----------|
| 4. Lenguajes | 57 |
| Historia de los lenguajes orientados a objetos | 60 |
| Ventajas de los lenguajes orientados a objetos | 63 |
| Funcionalidad de los lenguajes orientados a objetos | 67 |
| Objetos y clases | 69 |
| Métodos y mensajes | 70 |
| Herencia | 71 |
| Polimorfismo | 73 |
| Ligadura dinámica | 74 |
| Herencia múltiple | 75 |
| Bibliotecas de clases | 78 |
| Herramientas de desarrollo | 80 |
| Estudio de los lenguajes orientados a objetos | 82 |

| | |
|---|------------|
| C++ | 82 |
| Objective-C | 86 |
| Pascal orientado a objetos | 89 |
| Smalltalk | 91 |
| Eiffel | 95 |
| Common Lisp Object System | 99 |
| Desarrollo de temas técnicos | 101 |
| Ligadura dinámica frente a estática | 101 |
| Clases como objetos | 102 |
| Concurrencia | 103 |
| Estándares o normas | 104 |
| Resumen | 104 |
| 5. Bases de datos | 109 |
| Historia de las bases de datos orientadas a objetos | 110 |
| Ventajas de las bases de datos orientadas a objetos | 118 |
| Funcionalidad de las bases de datos orientadas a objetos | 121 |
| Visión general | 122 |
| Objetos | 124 |
| Métodos | 126 |
| Herencia | 127 |
| Bibliotecas de clases | 128 |
| Persistencia | 129 |
| Consulta | 130 |
| Integridad | 131 |
| Temas técnicos en evolución | 132 |
| Optimización de la consulta | 133 |
| Base de datos distribuida | 134 |
| Concurrencia | 135 |
| Rendimiento | 137 |
| Modificación de esquema | 137 |
| Soporte del lenguaje | 138 |
| Estándares | 138 |
| Aplicaciones | 139 |
| Resumen | 140 |
| 6. Interfaces de usuario | 143 |
| Historia de los interfaces de usuario orientados a objetos | 144 |
| Ventajas de los interfaces de usuario orientados a objetos | 146 |
| Funcionalidad de los interfaces de usuario orientados a objetos | 152 |
| Windows y Presentation Manager | 153 |
| X Window | 159 |



| | |
|---|-----|
| Constructor de interfaces de NeXTStep | 165 |
| Macintosh de Apple | 167 |
| Herramientas de desarrollo para interfaces de usuario orienta- dos a objetos | 171 |
| Smalltalk | 172 |
| NewWave | 173 |
| Caseworks | 176 |
| CommonView | 177 |
| NeWS | 179 |
| Metaphor | 180 |
| Resumen | 180 |

Tercera parte

Desarrollo de aplicaciones orientadas a objetos 185

| | |
|--|-----|
| 7. Análisis y diseño | 187 |
| Historia del análisis y diseño orientados a objetos | 188 |
| Análisis estructurado de sistemas | 188 |
| Análisis orientado a objetos | 190 |
| Ventajas del análisis y diseño orientados a objetos | 191 |
| Proceso del análisis y diseño orientados a objetos | 193 |
| Identificación y definición de objetos | 194 |
| Definición y organización de clases | 199 |
| Metodologías emergentes | 202 |
| Resumen | 218 |
| 8. Programación y mantenimiento | 221 |
| Programación y mantenimiento procedimental (procedural) .. | 225 |
| Programación y mantenimiento orientados a objetos | 228 |
| Gestión de proyectos orientada a objetos | 237 |
| Herramientas y entornos de desarrollo | 240 |
| Examinadores ("Hojeadores") | 240 |
| Inspectores | 243 |
| Comprobadores (<i>Profilers</i>) | 244 |
| Depuradores de errores | 244 |
| Entornos de desarrollo | 247 |
| Resumen | 250 |

| | |
|---|-----|
| 9. Aplicaciones emergentes | 253 |
| Ventajas de las aplicaciones orientadas a objetos | 254 |
| Mayor flexibilidad | 254 |
| Integración transparente | 255 |
| Empleo más fácil | 256 |
| Funcionalidad de las aplicaciones orientadas a objetos | 257 |
| Desde el procesamiento a la simulación para la realización | 257 |
| Desde la computación centralizada a la distribuida | 258 |
| Desde el texto y los gráficos al multimedia | 260 |
| Desde el "casi objeto" a la orientación a objetos | 261 |
| Ejemplos de aplicaciones orientadas a objetos | 265 |
| Ingeniería de software asistida por computador | 266 |
| Enseñanza asistida por computador | 267 |
| Fabricación integrada por computador | 269 |
| Edición asistida por computador | 269 |
| Entornos de programación visual | 271 |
| Tendencias futuras en las aplicaciones orientadas a objetos | 273 |
| Resumen | 273 |

Cuarta parte

| | |
|--|-----|
| Apéndices | 277 |
| A. Glosario | 279 |
| Términos | 279 |
| Siglas | 294 |
| B. Vocabulario técnico bilingüe | 301 |
| C. Referencias | 305 |
| Índice de materias | 311 |