

## INDICE

|   |     |
|---|-----|
| <b>1. Herramientas de Programación</b>        |     |
| La caja de herramientas de los programadores  | 2   |
| El objetivo                                   | 3   |
| ¿Por qué molestarse?                          | 4   |
| Al principio                                  | 4   |
| Compresión del problemas                      | 6   |
| Diseño descendente                            | 8   |
| Escritura de código fácilmente modificable    | 14  |
| Documentación                                 | 14  |
| Prueba de programas                           | 21  |
| Una advertencia                               | 24  |
| Resumen                                       | 24  |
| Aplicación: Problemas conducidos por órdenes  | 26  |
| Ejercicios                                    | 34  |
| Pre – Examen                                  | 36  |
| <b>2. Estructuras de Datos Incorporadas</b>   |     |
| Arrays uni – dimensionales                    | 44  |
| Arrays bi – dimensionales                     | 49  |
| Registros                                     | 56  |
| Estructuras empaquetadas                      | 58  |
| Comprobación de errores                       | 59  |
| Resumen                                       | 59  |
| Aplicación: Cadenas                           | 61  |
| Ejercicios                                    | 86  |
| Pre – Examen                                  | 88  |
| <b>3. Pilas</b>                               | 91  |
| Estructuras de datos definidas por el usuario | 92  |
| ¿Qué es una pila?                             | 93  |
| Operaciones sobre pilas                       | 94  |
| Implementación de una pila como un array      | 96  |
| Una implementación más general                | 101 |
| El nivel de aplicación                        | 104 |
| Resumen                                       | 104 |
| Aplicaciones: Evaluación de expresiones       | 106 |
| Laberinto                                     | 129 |
| Ejercicios                                    | 142 |
| Pre – Examen                                  | 146 |
| <b>4. Colas</b>                               | 149 |
| Nivel lógico: ¿Qué es una cola?               | 150 |
| Operaciones sobre colas                       | 151 |
| Implementación de la cola                     | 153 |
| Otro diseño de las colas                      | 155 |
| Aplicaciones de las colas                     | 161 |
| Resumen                                       | 162 |
| Aplicación: Simulación                        | 163 |
| Ejercicios                                    | 178 |
| Pre – Examen                                  | 182 |

|   |     |
|---|-----|
| <b>5. Listas Enlazadas</b>  | 185 |
| Niveles de la abstracción de daos   | 186 |
| Representaciones de listas  | 187 |
| Una razón para usar listas enlazadas  | 188 |
| Una perspectiva abstracta de una lista enlazada                               | 189 |
| Anatomía de un nodo   | 189 |
| Algunas notaciones  | 190 |
| Algunas operaciones sobre una lista enlazada                                  | 190 |
| Operaciones OBTENERNODO y LIBERARNODO   | 193 |
| Implementación de una lista enlazada como u array de registros                | 194 |
| Comenzando  | 197 |
| Implementación de las operaciones sobre listas                                | 198 |
| Inserción en una lista enlazada ordenada                                      | 200 |
| Implementación del algoritmo de insertar                                      | 204 |
| Supresión en una lista enlazada   | 207 |
| Implementación del algoritmo de suprimir                                      | 210 |
| Manteniendo una pila en una lista enlazada                                    | 212 |
| Listas enlazadas de la lista circular. La cola                                | 218 |
| Resumen   | 219 |
| Aplicación: Circulación de una revista  | 220 |
| Ejercicios  | 242 |
| Pre – Examen  | 245 |
| <b>6. Variables Punteros</b>  | 249 |
| La razón para la asignación dinámica  | 250 |
| Uso de las variables punteros en Pascal                                       | 251 |
| Sintaxis de la flecha hacia arriba  | 254 |
| Indicaciones para la depuración   | 261 |
| Más sobre listas  | 262 |
| Aplicaciones de las listas enlazadas  | 267 |
| Resumen   | 269 |
| Ejercicios  | 270 |
| Pre – Examen  | 274 |
| <b>7. Recursividad</b>  | 277 |
| Un ejemplo de recursividad  | 279 |
| Programación recursiva  | 281 |
| Vuelta a la función suma  | 284 |
| Una función Booleana  | 286 |
| Algunos otros aspectos  | 290 |
| Vuelta a IMPRIVERSO   | 290 |
| Cómo funciona la recursividad   | 292 |
| NFACT-Una vez más   | 296 |
| Resumen   | 299 |
| Aplicación: Ordenación rápida (Quicksort) – Un ejemplo de algoritmo recursivo | 300 |
| Ejercicios  | 310 |
| Pre – Examen  | 313 |
| <b>8. Árboles Binarios de Búsqueda</b>  |     |
| En el nivel lógico: Un árbol binario de búsqueda                              | 314 |

|   |     |
|---|-----|
| Algo de vocabulario sobre los árboles binarios                      | 316 |
| Búsqueda de un árbol  | 317 |
| Inserción en un árbol binario de búsqueda                           | 318 |
| Implementación del algoritmo de insertar                            | 322 |
| Supresión en un árbol binario de búsqueda                           | 323 |
| Implementación del algoritmo de suprimir                            | 326 |
| Algoritmo de supresión alternativo                                  | 330 |
| Recorridos de un árbol  | 335 |
| Aplicaciones de los árboles binarios de búsqueda                    | 342 |
| Resumen   | 342 |
| Aplicación: índice  | 343 |
| Ejercicios  | 366 |
| Pre – Examen  | 368 |
| <b>9. Más sobre Árboles</b>   | 371 |
| Árboles binarios de expresiones                                     | 372 |
| Una representación no enlazada de los árboles binarios              | 384 |
| Ordenación por montículo  | 387 |
| Grafos  | 395 |
| Resumen   | 404 |
| Ejercicios  | 404 |
| Pre – Examen  | 407 |
| <b>10. Verificación</b>   | 409 |
| Las técnicas de verificación  | 410 |
| El caso de la sentencia GOTO  | 415 |
| Aplicación: Ordenación por inserción                                | 418 |
| Ejercicio   | 425 |
| Pre – Examen  | 425 |
| <b>11. Algoritmos de Ordenación y Consideraciones de Eficiencia</b> | 427 |
| ¿Qué es “bueno”?  | 428 |
| Ordenación por selección directa                                    | 430 |
| Ordenación por burbuja  | 434 |
| La O – Grande   | 438 |
| Comparación en O – Grande de las ordenaciones                       | 439 |
| Análisis de la ordenación rápida                                    | 440 |
| Análisis de la ordenación por montículo                             | 441 |
| Otras consideraciones de eficiencia                                 | 441 |
| Más sobre la ordenación en general                                  | 443 |
| Resumen   | 446 |
| Ejercicios  | 446 |
| Pre – Examen  | 448 |
| <b>12. Búsqueda</b>   | 451 |
| Búsqueda secuencial   | 452 |
| Ordenación de claves  | 454 |
| Búsqueda binaria  | 455 |
| Transformación de claves  | 459 |
| Resumen   | 468 |
| Ejercicios  | 469 |
| Pre – Examen  | 470 |

|   |     |
|---|-----|
| <b>13. Bienvenido al Mundo Real</b>           | 473 |
| La compañía de automatización Acme (ficticio) | 475 |
| La situación del software                     | 475 |
| Objetivos de los buenos programas             | 478 |
| Principios de la creación de buenos programas | 479 |
| Realización de nuestros objetivos             | 488 |
| Elección de una estructura de datos           | 489 |
| Resumen                                       | 490 |
| Apéndices                                     | A1  |
| Respuestas a los Ejercicios                   | A25 |
| Glosario                                      |     |
| Asignaciones de Programación por Capítulo     | B1  |
| Índice  |     |