

Indice

UNIVERSIDAD NACIONAL DE ENTRE RÍOS
 FACULTAD DE INGENIERÍA
 CENTRO DE MEDIOS
 BIBLIOTECA

1095

| | | |
|-----------|--|--|
| ix | Prefacio | |
| 1 | Prólogo | |
| 4 | Capítulo 1 | |
| | El proceso de desarrollo de software. | |
| 4 | 1. ¿Ingeniería o artesanía de software? | |
| 5 | 2. El proceso de desarrollo de software. | |
| 13 | 3. Modelos del ciclo de vida del software. | |
| 13 | 3.1. El modelo clásico de Boehm. | |
| 15 | 3.2. Los nuevos paradigmas. | |
| 15 | 3.2.1. Prototipación. | |
| 18 | 3.2.2. Especificación operacional. | |
| 19 | 3.2.3. Implementación por transformación. | |
| 20 | 3.2.4. Desarrollo anidado. | |
| 22 | Capítulo 2 | |
| | El formalismo básico. | |
| 22 | 1. El concepto de problema. | |
| 30 | 2. Operaciones sobre problemas. | |
| 31 | 2.1. Suma de problemas. | |
| 33 | 2.2. El concepto de subproblema aditivo. | |
| 34 | 2.3. δ -soluciones de la suma de problemas. | |
| 38 | 2.4. Producto de problemas. | |
| 39 | 2.5. El concepto de subproblema multiplicativo. | |
| 40 | 2.6. Producto directo de problemas. | |
| 42 | 2.7. Objetivo de la teoría algebraica de problemas. | |
| 42 | 3. Términos y el concepto de solución en sentido amplio. | |
| 44 | 4. Sintaxis y semántica. | |
| 44 | 4.1 Lenguajes algorítmicos de programación. | |
| 45 | 4.2 Lenguajes declarativos de descripción de problemas. | |

| | | |
|-----------|--|--|
| 47 | Capítulo 3 | |
| | El problema de la programación | |
| 47 | 1. | Introducción. |
| 48 | 2. | La contrastación de teorías científicas. |
| 50 | 3. | La contrastación experimental de programas. |
| 50 | 4. | Clasificación de enunciados científicos. |
| 51 | 5. | Descomposición del problema de la programación. |
| 57 | 6. | Relación entre las aplicaciones, el problema de la programación y la meta-actividad de diseño y construcción de metodos de programación. |
| 60 | Capítulo 4 | |
| | Descomposición por abstracción. | |
| 60 | 1. | Introducción. |
| 62 | 2. | Descomposición. |
| 63 | 3. | Descomposición vía abstracción. |
| 65 | 4. | Abstracción. |
| 67 | 5. | Refinamientos sucesivos. |
| 67 | 5.1. | Preliminares. |
| 69 | 5.2. | El problema de ordenar una lista. |
| 70 | 5.2.1. | Especificación del dominio de las listas. |
| 71 | 5.2.2. | Especificación recursiva de Sort. |
| 73 | 5.2.3. | El programa abstracto. |
| 74 | 5.2.4. | Especificación de Split y Join. |
| 75 | 5.2.5. | Descomposición de Join. |
| 77 | 5.2.6. | Evaluación de la solución y discusión de alternativas. |
| 79 | 5.2.7. | Implementación del programa. |
| 81 | 5.3. | Conclusión. |
| 83 | 6. | El concepto de problema fácil. |
| 83 | 7. | Hacia la formalización de un esquema de descomposición. |
| 86 | 8. | Derivación del problema de búsqueda (Search) |

- 89** **Capítulo 5**
Hacia modelos generales del proceso de desarrollo de software.
- 89** 1. Por qué modelos generales descriptivos y prescriptivos.
90 2. El modelo PW.
92 3. El modelo LST del proceso de reificación.
100 4. La pierna izquierda del modelo PW.
104 5. Algunas consideraciones generales sobre el modelo PW.
105 6. Una visión más profunda del paso canónico del modelo LST.
109 6.1. Un ejemplo de aplicación del concepto de paso canónico: la implementación de un tipo abstracto de datos.
109 6.1.1. Implementación del tipo de datos lista de enteros.
110 6.1.2. Definición de la especificación Esp de lista de enteros.
112 6.1.3. Implementación del tipo: lista de enteros.
113 6.1.4. Interpretación de Esp en Impl.
116 7. La relación entre la especificación, el concepto de la aplicación y el problema final a la luz de la teoría de problemas.
119 8. Algunas consideraciones epistemológicas sobre el modelo general del proceso de desarrollo de software.
119 8.1. Los conceptos de teoría, interpretación y explicación en la ciencia empírica.
121 8.2. La pierna izquierda a la luz del concepto de interpretación en la ciencia empírica.
122 8.3. El proceso de reificación a la luz de la sistematización nomológico-deductiva.



Prefacio

- 164 Epílogo
- 167 Referencias Bibliográficas
- 173 Apéndice