



## INDICE

<i>Prólogo</i> .....	VII
<i>Prefacio</i> .....	IX
1. Principios básicos .....	1
¿Por qué LCP? Objetivos del diseño. ¿Qué es un programa? Estructuras lógicas. El lenguaje de programación. Teoría de conjuntos. Desarrollo de ideas.	
2. Estructuras repetitivas simples .....	10
Diagrama de estructura. Datos de entrada. Estructura del programa.	
3. Las instrucciones del programa .....	21
El procedimiento. Las categorías. La lista ordenada. <i>Advertencia</i> . Codificación en lenguaje de programación. <i>Codificación en párrafos sucesivamente consecutivos</i> . <i>Codificación en párrafos sucesivamente internos (anidados)</i> . Consideraciones prácticas.	
4. Estructuras alternativas simples .....	46
Subconjuntos alternativos. Valor de los elementos. Codificación generalizada. Numeración de párrafos. Subconjuntos alternativos-repetitivos.	
5. Estructuras complejas .....	65
Estructuras complejas repetitivas. Estructura compleja mixta. Estructuras complejas de programa. Codificación generalizada. Descomposición de los elementos del ALS. Numeración de la estructura de programa.	
6. Datos derivados .....	82
Uso de los datos de salida. Datos intermedios. Conflicto de secuencia. <i>Subrutinas</i> . Datos derivados considerados como salida. Programas basados en procedimientos.	
7. Estructuras alternativas complejas .....	104
Estructuras equivalentes. Derivación de la estructura de programa. Objetivos del diseño. <i>Facilidades de mantenimientos</i> . Eficiencia. Comparación. Sumario.	
8. Resolución de estructuras alternativas complejas .....	121
Optimización de la estructura. Algebra de Boole. Diagramas de Veitch. Aplicación a la programación. Estructuras mixtas. Raíz no evidente. Subrutinas. Casos de no combinación de subconjuntos. El conjunto nulo. Sumario.	
9. Aplicaciones normales. ....	138
Archivos múltiples de entrada. <i>Intercalación lógica</i> . Procedimientos de clasificación en	

	COBOL. Datos sin aplicación. <i>Salida lógica. Entrada lógica. Validación de entrada. Subrutinas e indicadores ("flags"). Búsqueda. Estructuras recurrentes. Codificación generalizada.</i>	
10.	Datos multivaluados y variables múltiples ..... Datos multivaluados. Estructuras simples. <i>Agrupamiento de valores. Eficiencia. Estructuras complejas. Modificación de los datos. Modificación de la técnica. Variables múltiples. Simplificación de las variables.</i>	181
11.	Programas en línea (on line) ..... Bases de datos. Interrupciones de entrada-salida.	224
12.	Pruebas de programación ..... Prueba de programa. Diseño del test de datos. Determinación de las rutas. <i>Estructuras repetitivas simples. Estructuras alternativas simples. Estructuras repetitivas complejas y complejas mixtas. Estructuras alternativas complejas. Predeterminación de la salida. Archivos de múltiples entradas. Subrutinas. Prueba de máquina.</i>	252
13.	Mantenimiento del programa ..... El procedimiento. <i>Caso A: no hay cambio en la estructura. Caso B: cambio de salida que requiere cambio de estructura. Caso C: cambios en los archivos físicos de entrada. Caso D: archivos físicos sin cambio, pero considerados en forma distinta desde el punto de vista lógico. Sumario. Corrección de programa no-LCP. Diagrama de flujo.</i>	270
14.	Conclusión ..... <i>1. Construir el ALS. 2. Construir el ALE. 3. Comprobar el ALE contra el ALS. 4. Construir los ALD necesarios. 5. Resolver las alternativas complejas. 6. Producir la estructura del programa. 7. Verificar la estructura del programa contra el ALS. 8. Construcción de todos los AFE y subrutinas asociadas. 9. Listar las instrucciones por tipo. 10. Considerar degeneraciones y subrutinas. 11. Poner las instrucciones en orden. 12. Comprobar contra el ALS. 13. Codificar. 14. Compilar. 15. Preparar datos de prueba. 16. Probar en máquina. Puntos prácticos. Desarrollo de sistemas. Conclusión.</i>	292
	<i>Bibliografía</i> .....	299