

Introducción a la computación 65
(Una guía temática)

Ricardo Berlanga
1988



UNIVERSIDAD

INTRODUCCION A LA COMPUTACION (UNA GUÍA TEMÁTICA)

Ricardo Berlanga

Julio de 1988

"Un primer curso debe introducir a la programación como el arte o la técnica de construir y formular algoritmos de manera sistemática, reconociendo en ello una disciplina en su propio derecho."

"Los algoritmos, en el sentido más general, son recetas para clases de procesamiento de datos y procesos de control."

El estudiante debe ser educado a proceder metódica y sistemáticamente en el diseño de algoritmos a través de problemas y técnicas típicas de la programación, pero independientes del área de aplicación. En el mismo espíritu, las estructuras algorítmicas fundamentales deben recibir todo el énfasis y no las particularidades de notación o lenguaje; el lenguaje de programación es nuestra herramienta pero no un fin en sí mismo."

"Un curso en la construcción de algoritmos es parte del entrenamiento matemático básico del estudiante y no un instrumento para satisfacer las necesidades inmediatas de quienes desean estar capacitados para ocasionalmente codificar un problema y entregárselo a la computadora para su solución instantánea." Niklaus Wirth.

La guía temática propuesta a continuación es demasiado extensa para un curso y de ninguna manera su estructuración es pedagógicamente aceptable. Sin embargo ofrece un cuadro inicial y orgánico del mundo del procesamiento de la información.

I. Computadoras: Un Panorama. (Helms, Levine, Nanney, Pekelis, Pollack, Ulan)

- a) Resumen histórico.
- b) Las computadoras en la sociedad.
- c) La rapidez de la computadora.
- d) El costo decreciente de las computadoras.
- e) La computadora como herramienta.
 - i) Problemas científicos.
 - ii) Problemas matemáticos.
 - iii) Problemas en Ingeniería (CAD/CAM)
 - iv) Problemas alfanuméricos y manipulación simbólica.
 - v) Problemas administrativos y actuariales.
- f) Las ciencias de la computación.
 - i) Hardware.
 - ii) Programación y organización de sistemas.
 - iii) Software.
 - iv) Datos, información y códigos.
 - v) Metodologías.
 - vi) Matemáticas de la computación.
 - vii) Bases teóricas de la computación.
 - viii) Aplicaciones.
- g) El impacto de la computación en las matemáticas.

II). ¿Cómo Funciona una Computadora? (Helms, Levine, Nanney).

- a) Familiarización con el equipo existente en la Facultad.
 - i) Descripción, manejo físico y cuidado.
 - ii) Diskettes, teclado, monitor, disco duro e impresora.
- b) Descripción superficial de algunos paquetes y lenguajes.
 - i) Sistemas operativos (e.g. MS/DOS).
 - ii) Procesadores de texto (e.g. T/MAKER, CHI-WRITER).
 - iii) Lenguajes (e.g. Pascal, C, Fortran, Basic).
 - iv) Bases de datos (e.g. dBase).
 - v) Hojas electrónicas (e.g. Lotus 1-2-3).
 - vi) Paquetes de estadística (e.g. Stat-Graphics).
 - vii) Paquetes de análisis numérico (e.g. Phaser).
 - viii) Paquetes de graficación (e.g. los hechos por los alumnos de esta Facultad).
 - ix) Paquetes de CAD.
 - x) Otros (e.g. Windows).

- c) El modelo de von Neumann.
- d) Descripción funcional de un sistema de cómputo.
 - Arquitectura.
 - i) El procesador central. Registros.
 - ii) La memoria central. Direcciones.
 - iii) Estructura del "Bus". La transferencia de datos.
 - iv) Entrada/salida.
 - v) Memoria auxiliar.
 - vi) Control.
 - vii) Teleproceso.
- e) Tipos de computadoras.
 - i) Analógicas.
 - ii) Digitales (personales, minis, "main-frames", redes).
 - iii) Híbridas.

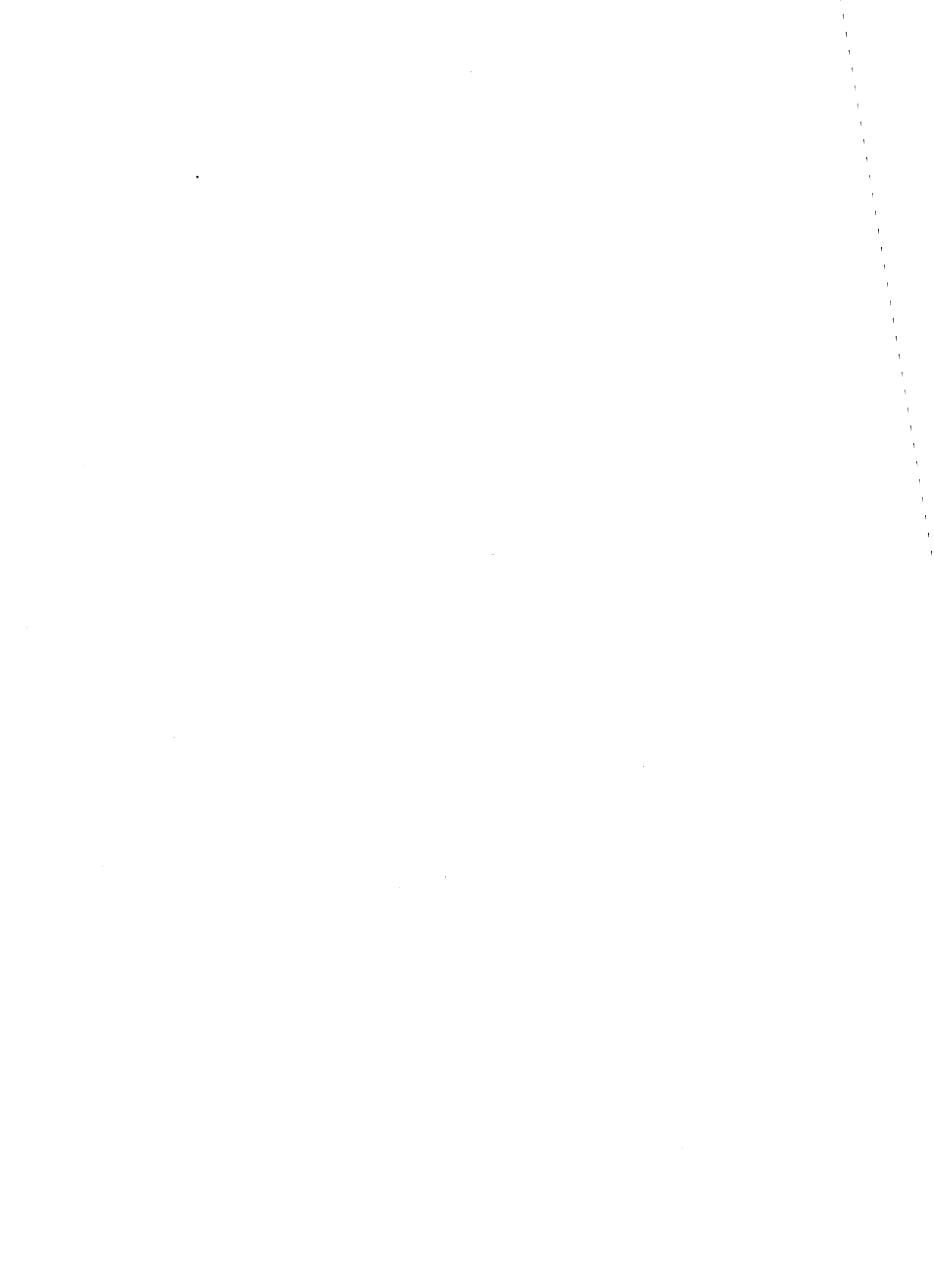
III). Comunicación, Información y Lenguaje (Ghezzi, Hamming, Helms, Hofstadter, Kline, Lewis, Malitz, Nanney, Pekelis, Wiener).

- a) Modelos de comunicación.
- b) Información. Datos.
 - i) Códigos.
 - ii) Canales y capacidad.
 - iii) Muestras.
 - iv) Entropía. Shannon
- c) Lenguajes (ver IX). Bases Teóricas de la Computación).
 - i) Naturales.
 - ii) Formales.
 - iii) De computación.

Nota: La revolución que hoy vivimos finalmente queda enmarcada en este capítulo. Lo demás es antecedente, herramienta, implementación y estructura, soporte o consecuencia.

IV). Algebra Booleana y Circuitos Lógicos. (Helms, Mendelson).

- a) El cálculo proposicional.
 - i) Tablas de verdad. Tautologías.
 - ii) Axiomatización y demostración.
- b) Algebra Booleana.
 - i) Funciones booleanas.
 - ii) Formas canónicas y normales.
 - iii) Un ejemplo: Teoría de conjuntos.



- c) Circuitos lógicos.
 - i) Compuertas.
- d) Sintaxis .vs. Semántica.
- e) Hardware.
 - i) El elemento "flip-flop".
 - ii) Estados.
 - iii) Conversión de tablas de estados en diagramas lógicos y viceversa.
 - iv) Microprocesadores. Chips y semiconductores.
 - v) Organización de la memoria.

V). Programación de Sistemas (Levine, Ullman).

- a) Lenguaje de máquina y ensambladores.
- b) Cargadores, encadenadores y preprocesadores.
- c) Compiladores y traductores. Intérpretes.
- d) Sistemas operativos.

VI). Datos y su Representación. (Scott).

- a) Números, bits y bases.
 - i) Codificación binaria de números naturales, enteros y reales.
- b) Operaciones en \mathbb{N} , \mathbb{Z} y \mathbb{R} .
 - i) Suma y resta.
 - ii) Multiplicación.
 - iii) División.
 - iv) Corrimientos.
 - v) Operaciones trascendentes. Exponencial y logaritmo.
- c) Precisión. Redondeo y truncamiento.
- d) Errores aritméticos. "Underflow" y "overflow".
- e) Representación de cuerdas y caracteres.
 - i) Operaciones (e.g. corrimientos, máscaras y concatenación).
- f) Representación de otras clases de datos.

VII). Introducción a las Estructuras de Datos. (Wirth [1]).

- a) El concepto de tipo ("Data type").
- b) Tipos primitivos.
- c) Arreglos.
- d) Registros.
- e) Conjuntos.

- f) Archivos secuenciales.
- g) Estructuras dinámicas de información.
 - i) Listas encadenadas.
 - ii) Árboles.
 - iii) Gráficas y trayectorias.
- h) Operaciones (ver VIII). Algoritmos).
 - i) Ordenamientos, encadenamientos, particiones, inserciones y mezclas.
 - ii) Recursión.
 - iii) Recorridos y búsquedas.

VIIID. Algoritmos. (Knuth, Malitz, Nanney, Wirth [1], [2]).

- a) Procesos
- b) Procedimientos efectivos. Finitud.
- c) Algoritmos.
 - i) Numéricos
 - ii) Pertinentes a las estructuras de datos mismas.
 - iii) Combinatorios.
 - iv) Recursivos.
 - v) Sintácticos.
- d) Decisiones.

IX). Bases Teóricas de la Computación. (Hamming, Hofstadter, Kleene, Lewis, Malitz).

- a) Automatas.
- b) Máquinas de Turing. Algoritmos abstractos.
- c) Gramáticas
 - i) Generación y reconocimiento.
 - ii) Expresividad y complejidad.
- d) Indecidibilidad.
- e) Aritmetización de las Metamatemáticas. Gödel.
- f) Teoría de modelos.
- g) La tesis de Church.

X). Elementos de Programación. (Levine, Nanney, Wirth [2]).

- a) El concepto de programa de computación.
- b) Fases de la creación de un programa.
 - i) Comprensión.
 - ii) Análisis y diseño.
 - iii) Programación.

- iv) Codificación e implementación.
- v) Ejecución y ajuste.
- vi) Mantenimiento y mejoramiento.
- c) Elementos sintácticos de un programa.
 - i) Identificadores. Palabras reservadas.
 - ii) Constantes.
 - iii) Operadores.
 - iv) Expresiones.
 - v) Declaraciones y tipos.
 - vi) Archivos y entrada/salida.
 - vii) Afirmaciones y control de flujo
 - viii) Apuntadores y estructuras.
 - ix) Funciones y procedimientos. Parámetros.
- d) Semántica. Atributos.
 - i) Alojamiento estático y dinámico.
 - ii) Valor y tipo.
 - iii) Alcance: vida y visibilidad.
 - iv) Paso de parámetros.

XI). Programación Estructurada (Levine, Nanney, Wirth [1], [2]).

- a) Pseudocódigo.
- b) Diagramas de flujo. Control.
- c) Módulos, procedimientos y subprogramas.
- d) Técnicas de diseño descendiente ("Top-down")
- e) Documentación.

XII). El Sistema Operativo MS/DOS. (Ver el manual).

- a) Discos y archivos. Sus nombres y atributos.
- b) Directorios y trayectorias. Sus nombres.
- c) Comandos: el mantenimiento de archivos.
- d) Configuración de un sistema.
- e) Manejadores de dispositivos.
- f) Errores y mensajes.
- g) Utilerías.
 - i) XTree y Norton Commander.
 - ii) The Norton Utilities.
 - iii) Otras

XIIID). Un Procesador de Texto (e.g. T/MAKER).

- a) Algunos conceptos básicos.
 - i) Pantalla, teclado y el ratón. La impresora.
 - ii) Modo de edición y modo de comando.
 - iii) Area de trabajo.
 - iv) El marco y la ventana.
 - v) El "pronto" y el cursor.
 - vi) El menú. Autoayuda.
 - vii) Posición, carácter, línea, columna, cuerda, párrafo, hoja y documento. Márgenes, espaciado y justificación.
 - viii) El tabulador y el carro.
 - ix) El "buffer" y macros.
 - x) Estilos de caracteres.
 - xi) Símbolos especiales
 - xii) Procesadores "ASCII" vs. procesadores en modo gráfico.
- b) Movimientos del cursor.
- c) Caracteres, líneas y columnas.
 - i) Inserción, supresión y borrado.
 - ii) Concatenación y rompimiento.
 - iii) Otros movimientos (sin uso del "buffer").
- d) El "buffer".
 - i) Limpiar el "buffer".
 - ii) Copiar el y al "buffer".
 - iii) Mover al "buffer".
- e) Búsqueda y reemplazamiento de cuerdas.
- f) El tabulador.
- g) Macros.
- h) Márgenes y alineación.
- i) Diccionario, índices y conteos.
- j) Manejo de archivos.
 - i) Generar, suprimir y salvar. Renombrar.
 - ii) Cortar, insertar, combinar, arreglar y cotejar.
- k) Programación de comandos.
- l) Impresión.
 - i) Diseño y acabado.
- m) Errores.

XIV). Un Lenguaje Específico.

- a) Instalación en el equipo existente.
 - i) Requerimientos.
- b) Estructura general de un programa en el lenguaje escogido.
- c) "Defaults" y precedencias.
 - i) Tamaño de los tipos estándar.
 - ii) Conversión automática de tipos.
 - iii) Inicialización automática.
 - iv) Declaraciones automáticas.
 - v) Precedencia en las operaciones. Precisión.
- d) El lenguaje estándar .vs. la versión escogida.
- e) Intérpretes y entornos integrados.
- f) Estilo.
- g) Edición de código fuente.
- h) El compilador
 - i) Directivos del preprocesador.
 - ii) Directivos del compilador.
- i) El ensamblador.
- j) El encadenador.
 - i) Unidades.
- k) Ejecución.
- l) Bibliotecas.
- m) Manejo de archivos.
- n) Entrada/salida
- o) Memoria.
 - i) Modelos de memoria.
 - ii) Segmentación. Cálculo de direcciones.
 - iii) Apuntadores cercanos, lejanos y enormes.
 - iv) Acceso directo a memoria.
 - v) El "Stack".
- p) Interacción con el sistema.
 - i) Interrupciones. Registros.
 - ii) Llamadas al sistema operativo.
 - iii) Acceso a puertos.

- q) Interface con otros lenguajes.
 - i) Convenciones de llamado.
 - ii) Parámetros de variables.
 - iii) Parámetros de valores.
 - iv) Resultados de funciones.
 - v) Llamadas cercanas y lejanas.
 - vi) Guardado de registros.
- r) Errores y expurgado ("debug")
 - i) De compilación y encadenamiento.
 - ii) De ejecución y entrada/salida.
 - iii) En el uso de apuntadores.
 - iv) Métodos de rastreo.

XV). Corrección y Eficiencia. Análisis de Programas. (Wirth [2]).

- a) Legibilidad.
- b) Portabilidad y adaptabilidad.
- c) Estructura y diseño.
- d) Robustez.
- e) Verificación
 - i) Verificaciones estáticas.
 - ii) Verificaciones al tiempo de ejecución.
 - iii) Ensayos.
 - iv) Ejecución simbólica.
- f) Detección de errores.
- g) Demostraciones de corrección.
- h) Complejidad algorítmica.

XVI). Diseño de Experimentos Matemáticos. (Grenander).

- a) Elección de las estructuras de datos.
- b) Búsqueda de lo inesperado.
- c) Repetición de los experimentos en forma modificada.
 - i) Rediseño a partir de la experiencia obtenida.
- d) Interpretación de los resultados.

XVII). Ejemplos. (Churchhouse, Dahlquist, Grenander, Knuth, Wirth [1], [2]).

- a) Del análisis.
 - i) Series.
 - ii) Aproximación de funciones.
 - iii) Ecuaciones no lineales.
 - iv) Diferenciación e integración.
 - v) Ecuaciones diferenciales.
 - vi) Ecuaciones integrales.
 - vii) Interpolación y extrapolación.
 - viii) Análisis de Fourier.
- b) De la Teoría de Números.
- c) De la Teoría de Gráficas y Combinatoria.
- d) De la Teoría de Juegos.
- e) De la Probabilidad (números aleatorios).
- f) De la Estadística.
- g) Del Algebra Lineal.
- h) De la Optimización y el Control.
- i) De la Modelación.
- j) De la Simulación. Método de Monte Carlo.
- k) De la Teoría de Grupos.
- l) De la Geometría Moderna y Analítica.
- m) De los Sistemas Dinámicos.
- n) Algoritmos recursivos.
- o) De la Inteligencia Artificial.
- p) De la Computación Gráfica.
- q) Del Procesamiento de Imágenes.
- r) Del Reconocimiento de Patrones.
- s) Visión computacional
- t) Sistemas expertos.
- u) Compiladores.
- v) Otros.

Nota: Cada renglón se debe desglosar. Esto corresponde a los interesados. (c.f. I).f).

Bibliografía:

- Churchhouse (Editor): Numerical Methods. Wiley (1981).
- Dahlquist, Björck: Numerical Methods. Prentice-Hall (1974).
- Ghezzi, Jazayeri: Programming Language Concepts. Wiley (1982).
- Grenander: Mathematical Experiments on the Computer. Academic Press (1982).
- Hamming: Coding and Information Theory. Prentice-Hall (1980).
- Helms. Computer Handbook. McGraw-Hill (1983).
- Hofstadter: Gödel, Escher, Bach. Harvester Press (1979).
- Kleene: Introduction to Metamathematics. Wolters-Noordhoff and North-Holland (1971).
- Kline (Editor): Mathematics in the Modern World. Freeman (1968).
- Knuth: Fundamental Algorithms. Addison-Wesley (1968).
- Levine: Introducción a la Computación. McGraw-Hill (1985).
- Lewis, Papadimitriov: Elements of the Theory of Computation. Prentice-Hall (1981).
- Malitz: Introduction to Mathematical Logic. Springer-Verlag (1979).
- Nanney: Pascal. Prentice-Hall (1985).
- Pekelis: Cybernetics A to Z. MIR (1974).
- Pollack (Editor): Studies in Computer Science. MAA (1982).
- Scott: Computer Number Systems and Arithmetic. Prentice-Hall (1985).
- Ulam: Science, Computers and People. Birkhauser (1986).
- Ullman: Fundamental Concepts of Programming Systems. Addison-Wesley (1976).
- Wiener: The Human Use of Human Beings. Avon (1979).
- Wirth [1]: Algorithms + Data Structures = Programs. Prentice-Hall (1976).
- Wirth [2]: Systematic Programming: An Introduction. Prentice-Hall (1973).

