

PROGRAMA DE LA ASIGNATURA: ANÁLISIS DISEÑO E IMPLANTACIÓN DE ALGORITMOS

PLAN:	98	CLAVE:	1132
LICENCIATURA:	INFORMÁTICA	CRÉDITOS:	8
ÁREA:	INFORMÁTICA	SEMESTRE:	1º.
REQUISITOS:	NINGUNO	HRS. CLASE:	2
TIPO DE ASIGNATURA:	OBLIGATORIA ( X )	HRS. POR SEMANA:	4
		OPTATIVA:	( )

OBJETIVO GENERAL:

AL FINALIZAR EL CURSO, EL ALUMNO CONOCERÁ LOS FUNDAMENTOS BÁSICOS DE LA COMPUTACIÓN Y PODRÁ UTILIZAR ESTRATEGIAS ALGORÍTMICAS PARA LA SOLUCIÓN DE PROBLEMAS.

TEMAS:	HORAS SUGERIDAS:
I. AUTÓMATAS Y LENGUAJES FORMALES.	12
II. COMPUTABILIDAD.	12
III. FUNCIONES RECURSIVAS.	6
IV. DISEÑO DE ALGORITMOS PARA LA SOLUCIÓN DE PROBLEMAS.	10
V. EVALUACIÓN DE ALGORITMOS.	6
VI. ESTRATEGIAS DE PROGRAMACIÓN PARA LA IMPLANTACIÓN DE ALGORITMOS.	16
EVALUACIÓN	6
	_____
	TOTAL: 68

TEMAS:

I. AUTÓMATAS Y LENGUAJES FORMALES.

- 1.-Definición de algoritmo.
  - 1.1.Clasificación de algoritmos
- 2.-Autómatas Finitos
- 3.-Máquina de Turing

II. COMPUTABILIDAD.

- 1.-Definición de un problema desde el punto de vista computable.
- 2.-Definición de computabilidad.
- 3.-Definición de complejidad.
- 4.-Problemas computables y no computables.
  - 4.1.Algoritmos de complejidad polinomial.
  - 4.2.Problemas intratables.

III. FUNCIONES RECURSIVAS.

- 1.-Definiciones Recursivas
- 2.-Análisis de programas recursivos
- 3.-Análisis de programas no recursivos
  - 3.1.Fibonacci
  - 3.2.Torres de Hanio
  - 3.3.Series y Sucesiones

IV. DISEÑO DE ALGORITMOS PARA LA SOLUCIÓN DE PROBLEMAS.

1. Análisis del problema.
  - 1.1. Identificación y abstracción de las constantes, variables, parámetros y sus relaciones.
  - 1.2. Especificaciones de entrada y salida.
  - 1.3. Procesamiento interno.
2. Diseño de algoritmos.
  - 2.1. Niveles de abstracción para la construcción de algoritmos.
    - 2.1.1. Lenguaje natural.
    - 2.1.2. Seudo-código.
    - 2.1.3. Diagramación.
  - 2.2. Estructuras básicas en un algoritmo.
    - 2.2.1. Ciclos.
    - 2.2.2. Contadores.
    - 2.2.3. Acumuladores.
    - 2.2.4. Condicional.
    - 2.2.5. "SI...ENTONCES...DE OTRA FORMA" (IF THEN-ELSE)
    - 2.2.6. Interruptores.
  - 2.3. Rutinas recursivas.
  - 2.4. Refinamiento progresivo.
  - 2.5. Procesamiento regresivo.
    - 2.5.1. Backtracking

## V. EVALUACIÓN DE ALGORITMOS.

1. Refinamiento progresivo.
2. Depuración y prueba.
  - 2.1. Eficiencia y confiabilidad.
  - 2.2. Verificación.
3. Documentación del programa.
  - 3.1. Propósito de la documentación.
  - 3.2. Herramientas para documentación.
4. Mantenimiento de programas.
  - 4.1. Tipos.
  - 4.2. Perfectivo.
  - 4.3. Correctivo.
  - 4.4. Adaptativo.
  - 4.5. Estrategias para mantenimiento.

## VI. ESTRATEGIAS DE PROGRAMACIÓN PARA LA IMPLANTACIÓN DE ALGORITMOS.

1. El programa como una expresión computable del algoritmo.
2. Realización de rutinas para:
  - 2.1. Ciclos.
  - 2.2. Contadores.
  - 2.3. Acumuladores.
  - 2.4. Condicional.
  - 2.5. "SI...ENTONCES...DE OTRA FORMA" (IF-THEN-ELSE).
  - 2.6. Interruptores.
  - 2.7. Recursión.
3. Programación no estructurada.
4. Programación estructurada.
  - 4.1. Concepto de modularidad.
  - 4.2. Flujo de información y control entre módulos.
  - 4.3. Enfoque arriba-abajo. (TOP-DOWN).- Enfoque abajo-arriba (BOTTOM-UP)
5. Divide y vencerás

#### BIBLIOGRAFÍA BÁSICA

1. ABELLANAS, M., *Análisis de Algoritmos y Teoría de Grafos*, México, Macrobit.
2. ACM Turing Award Lectures: *The First Twenty Years, 1966-1985*, Addison-Wesley.
3. AHO, Alfred, *Introducción a las Ciencias Computacionales*, México, Grupo Editorial Iberoamérica.
4. ANDERSON, John, *Foundations of Computers Technology*, Londres, Chapman & Hall.
5. BRONSON Gary J., *C++ para Ingeniería y ciencias*, México, Thomson-Learning, 1999, 884 p.p.
6. BROOKSHEAR, Glenn, *Introducción a las Ciencias de la Computación*, México, Addison-Wesley, 1995, (4ª. ed.)
7. CAIRÓ BATTISTUTTI, Oswaldo, *Metodología de la programación, Tomo I. Algoritmos, Diagramas de flujo y programas*, México, Alfaomega, 2002.
8. CEBALLOS, Francisco Javier, *Programación orientada a objetos con C++*,. España, Alfaomega-Rama, 2002.
9. CEBALLOS, Francisco Javier, *Curso de programación con C Microsoft*,. España, Alfaomega-Rama, 2002.
10. CEBALLOS, Francisco Javier, *Enciclopedia del lenguaje C*, España, Alfaomega-Rama, 2002.
11. COHEN, Daniel, *Introduction to computer Theory*, EEUU, Wiley,
12. HAREL, David, *Algorithmics, the Spirit of Computing*, EEUU, Addison-Wesley, 1987.
13. JAMSA, Kris, *Aprenda C++ Paso a paso*, España, Alfaomega, 2002.
14. JAMSA, Kris, *C++ Programación exitosa*, España, Alfaomega-Rama, 2002.
15. MATEOS, Alfonso, *Programación lineal y aplicaciones. Ejercicios resueltos*,. España, Alfaomega-Rama, 2001

#### BIBLIOGRAFÍA COMPLEMENTARIA

1. ALLEN, Weiss, *Estructuras de datos y Algoritmos*, EEUU, Addison-Weisley, 1995.
2. GRAHAM, Ronald, *Concrete Mathematics: A Foundation for Computer Science*, EEUU,.
3. KELLEY, Dean, *Teoría de Autómatas en Lenguajes Formales*, España, Prentice-Hall, 1995.

**TÉCNICAS DE ENSEÑANZA SUGERIDAS:**

Exposición oral	( X )
Exposición audiovisual	( )
Ejercicios dentro de la clase	( X )
Seminarios	( X )
Lecturas obligatorias	( X )
Trabajos de investigación	( X )
Prácticas de taller o laboratorio	( X )
Prácticas de campo	( )
otras	( )

**ELEMENTOS DE EVALUACIÓN:**

Exámenes parciales	( X )
Exámenes finales	( X )
Trabajos y tareas fuera del aula	( X )
Participación en clase	( X )
Asistencia a prácticas	( X )
otros	( X )