

PROGRAMACIÓN II

| | | | |
|---------------------------|----|---|-------------|
| CLAVE: | | SECTOR : | BÁSICO |
| SEMESTRE: | 3 | ÁREA: | INFORMÁTICA |
| CRÉDITOS: | 10 | SERIACIÓN: | |
| | | ASIGNATURA PRECEDENTE INDICATIVA: Programación I. | |
| | | ASIGNATURA SUBSECUENTE INDICATIVA: Ninguna | |
| HORAS POR CLASE | | TEÓRICA: | 1 |
| CLASES POR SEMANA | | TEÓRICA: | 4 |
| HORAS POR SEMESTRE | | TEÓRICA: | 64 |
| | | PRÁCTICAS: | 2 |
| | | PRÁCTICAS: | 1 |
| | | PRÁCTICAS: | 32 |

Objetivos generales: Al final del curso el alumno:

- Conocerá y aplicará los métodos y técnicas para especificación, validación y verificación de *software*.
- Será introducido al concepto de tipo abstracto de dato (*abstract data type*).
- Estará capacitado para comprender los fundamentos relativos al concepto, la implantación y las aplicaciones de las estructuras de datos básicas - listas, arreglos, tablas, pilas o *stacks*, colas, árboles, gráficas - y los operadores asociados a ellas.
- Será introducido a la definición, implantación y aplicaciones de las estructuras de datos no lineales y los operadores asociados a ellas.
- Comparará distintos algoritmos para búsquedas y ordenamientos, con especial atención al balance entre complejidad y espacio contra tiempo.

Tema 1. Especificación, verificación y validación

8 horas teóricas
4 horas teórico-prácticas

Explicará los conceptos y aplicaciones de la especificación, verificación y validación.

- 1.1 Tipos de especificaciones: especificaciones operativas y descriptivas, construcción y uso.
- 1.2 Verificación. Metas, enfoques y pruebas; análisis; ejecución simbólica; depuración.
- 1.3 Lectura de código y diseño y recorridos estructurados.

Tema 2. Tipos abstractos de datos

8 horas teóricas
4 horas teórico-prácticas

Reconocerá el concepto de tipo abstracto de dato y sus implicaciones prácticas.

- 2.1 Conceptos involucrados.
- 2.2 Instrumentación de TAD (ADT) en un lenguaje de alto nivel, con ejemplos.

Tema 3. Estructuras de datos básicas

20 horas teóricas
10 horas teórico-prácticas

Comprenderá los principales conceptos y operadores relacionados con las estructuras de datos.

- 3.1 Definición de las estructuras de datos lineales. Estructuras secuenciales. Arreglos empacados.
- 3.2 Uso de las estructuras de datos básicas.
- 3.3 Implementaciones contiguas y ligadas, ajuste para que respondan al problema planteado, incluyendo contraposición entre tiempo y espacio.
- 3.4 Estructuras múltiples. Búsqueda, inserción y remoción de elementos. Listas circulares y bidireccionales. Listas múltiples.

Tema 4. Estructuras de datos no lineales

12 horas teóricas
6 horas teórico-prácticas

Explicará la naturaleza de las estructuras de datos no lineales y la importancia de éstas en la computación y las matemáticas.

- 4.1 Presentación de las estructuras no lineales. Árboles y estructuras arborescentes. Árboles binarios. Representación de árboles arbitrarios en base a árboles binarios.
- 4.2 Listas y recolección de basura. Asignación dinámica de espacio.

Tema 5. Búsqueda y ordenamientos

16 horas teóricas
8 horas teórico-prácticas

Comparará diferentes algoritmos para búsquedas y ordenamientos y sus implicaciones.

- 5.1 Algoritmos de ordenamiento de orden *n*long (*quicksort*, *heapsort*, *mergesort*), complejidad entre el tiempo y el espacio: mejores y peores casos.
- 5.2. Otros algoritmos de ordenamiento (algoritmo de Shell, de cubetas y de radicales).
- 5.3 Comparación de algoritmos.
- 5.4 Funciones de dispersión y resolución de colisiones.
- 5.5 Algoritmos de búsqueda y manejo de árboles balanceados; árboles B y AVL.
- 5.6 Algoritmo de ordenamientos externos.

Bibliografía básica:

- Tucker, B. *et al. Fundamental of Computing, I: Logic, Problems Solving, Programs and Computers*. 2nd edition. USA. McGraw-Hill. 1994.
- Magidin, M. *Estructuras de datos*. México. Editorial Trillas. 1991.

Bibliografía complementaria:

- Aho, A.V. *et al. Estructuras de datos y algoritmos*. México. Addison Wesley Publishing Company, 1988.
- Base, S. *Computer Algorithms. Introducing to Design and Analysis*. USA. Addison Wesley Publishing Company. 1990.
- Cormen, T.H. *et al. Introduction to Algorithms*. USA. McGraw-Hill Book Company. 1990.
- Biggerstaff, T.J., Perles A. J.; edited by Software Reusability, Volume I, II, ACM PRESS Addison Wesley Publishing Company, 1989.
- Ghezzy, C. *et al. Fundamentals of Software Engineering*. USA. Prentice-Hall Inc. 1991.
- Glass, R. L. *Software conflict. Essays on the Art and Science of Software Engineering*. USA. Yourdon Press Computing series. 1991.
- Gehani, N., McGettrick, A.D. editors. *Software Specification Techniques*. USA. Addison Wesley Publishing Company. 1980.
- Gries, D. editor. *Programming in the 1990s. An Introduction to the Calculation of Programs*. Springer Verlag, 1990.
- Dijkstra, W. E. editor. *Formal Development of Programs and Proofs*. USA. Addison Wesley Publishing Company. 1990.

Sugerencias didácticas:

Se recomiendan:

Tareas semanales en las cuales el alumno aplique el material visto en clase.

Trabajos de investigación bibliográfica para que el alumno amplíe sus conocimientos y conozca diferentes enfoques del material del curso.

Prácticas de cómputo para la experimentación con los algoritmos vistos en clase.

Análisis y resolución de casos prácticos.

Forma de evaluación:

Se recomiendan de 3 a 4 exámenes parciales y un examen final, así como la realización de tareas sobre los temas vistos en clase para reforzar los conocimientos teóricos adquiridos.

Perfil profesiográfico:

Egresado preferentemente de las licenciaturas en Ciencias de la Computación, Ingeniería en Sistemas, Matemáticas, Actuaría o alguna afín con experiencia docente y profesional en el área.