



UNIVERSIDAD SIMÓN BOLÍVAR
Vicerrectorado Académico

1. Departamento: *Computación y Tecnología de la Información (6510)*

2. Asignatura: Algoritmos y Estructuras II

3. Código de la asignatura: CI2612

No. de unidades-crédito: 3

No. de horas semanales: Teoría 3 Práctica 1 Laboratorio 0

4. Fecha de entrada en vigencia de este programa: Abril 2012

5. OBJETIVO GENERAL: *Al final del curso, el estudiante está capacitado para diseñar un Tipo Abstracto de Datos (TAD), aplicar algoritmos y estructuras de datos clásicos en la solución de problemas computacionales y analizar la complejidad de algoritmos sencillos.*

6. OBJETIVOS ESPECÍFICOS: *El estudiante tendrá competencias para:*

1. *Interpretar el comportamiento de un TAD en el estilo de especificación basado en modelos*
2. *Especificar formalmente un TAD utilizando el estilo basado modelos descritos mediante la lógica de primer orden y estructuras matemáticas (conjuntos, multiconjuntos, secuencias, relaciones y funciones).*
3. *Aplicar técnicas de refinamiento de datos que permiten garantizar la consistencia de los modelos en la implementación de un TAD.*
4. *Seleccionar estructura de datos para representar un TAD utilizando criterios de eficiencia en tiempo y recursos.*
5. *Resolver problemas mediante el uso de algunos TADs y algoritmos clásicos (TAD Pila, TAD Cola, TAD Diccionario, algoritmos de Búsqueda y Ordenamiento, entre otros).*
6. *Representar información con tipos de datos estructurados recursivos*
7. *Concebir una solución recursiva para un problema computacional sencillo.*
8. *Verificar formalmente la validez de un algoritmo recursivo.*
9. *Diseñar programas estructurados que incluyan datos y funciones recursivos*
10. *Aplicar las nociones básicas del análisis de algoritmos en casos sencillos.*

7. CONTENIDOS:

TEORÍA

Tipos Abstractos de Datos (TAD): Concepto. Especificaciones formales con modelos. Ejemplo Diccionario. Modelos abstractos de representación: conjuntos, multiconjuntos, secuencias, relaciones y funciones. Implementación: refinamiento de datos, modelo concreto, invariante de acoplamiento. Correctitud de refinamiento de datos. Encapsulamiento y ocultamiento. Módulos y clases.

Especificaciones algebraicas. (8 horas)

Tipos Concretos de Datos: arreglos, registros, referencias o apuntadores. Implementación de tipos recursivos con estructuras de enlace simple y doble (listas y árboles) (4 horas)

Tipos Algebraicos Libres: ejemplos (expresiones, listas, árboles), definición de funciones, demostración de propiedades, primitivas de programación, programación de ejemplos iterativos y recursivos. (4 horas)

Árboles Binarios de Búsqueda: definición, propiedades, búsqueda, inserción y eliminación. Implementación. (2 horas)

Recursión. Introducción a las soluciones recursivas a problemas algorítmicos. Regla de corrección. (4 horas)

Introducción al Análisis de Complejidad de Algoritmos: notación O grande, Θ grande y Ω grande. Manejo de recurrencias. Aplicación en algoritmos clásicos (2 horas)

Algoritmos de Búsqueda y Ordenamiento: búsqueda secuencial, búsqueda binaria, ordenamiento por inserción, ordenamiento por selección, ordenamiento por intercambio (burbuja), ordenamiento por mezcla, ordenamiento rápido (Quicksort), ordenamiento de montículo (Heapsort). (6 horas)

PRÁCTICA

Ejercicios sobre especificación de TADs (Pila y Cola) (2 Horas).

Ejercicios sobre implementación de TADs (2 Horas)

Ejercicios sobre algoritmos recursivos (2 Horas)

Ejercicios sobre tipos algebraicos libres (2 Horas)

Ejercicios sobre implementación de algoritmos iterativos de búsqueda y ordenamiento (2 Horas)

Ejercicios sobre implementación de algoritmos recursivos de búsqueda y ordenamiento (1 Horas)

Ejercicios sobre árboles binarios de búsqueda (1 Hora)

8. ESTRATEGIAS METODOLÓGICAS, DIDÁCTICAS O DE DESARROLLO DE LA ASIGNATURA:

1. *Sesiones de teoría impartidas mediante clases magistrales.*
2. *Sesiones de ejercicios y/o problemas con discusión grupal.*
3. *Trabajos en grupo con ejercicios a resolver fuera del aula. Las dudas sobre estas tareas se aclaran en horas de consulta.*

A lo largo del curso se presenta la noción de TADs. En la primera parte se hace énfasis en la especificación formal con modelos. Luego se dan varias implementaciones de los TADs con tipos

concretos de datos, mostrando las bondades de unas respecto a las otras en base a criterios de eficiencia y uso de los recursos. Posteriormente, se presentan las estructuras de datos dinámicas como implementaciones alternativas. Luego se introducen los algoritmos clásicos de búsqueda y ordenamiento, su utilidad en la solución de un amplia variedad de problemas computacionales, así como su aplicación en muchos TADs. Se realiza el análisis de complejidad de estos algoritmos.

9. ESTRATEGIAS DE EVALUACIÓN:

1. Tres pruebas escritas con un porcentaje de 30% cada una. Estas pruebas son parciales. Se hacen ejercicios de: completación de la especificación formal de un TAD, completación de implementación de un TAD, corrección de refinamiento de datos, diseño de algoritmos recursivos, diseño de tipos algebraicos libres, uso e implementación de estructuras de datos dinámicas, completación de la implementación de algoritmos de búsqueda y ordenamiento, análisis de complejidad .
2. Ejercicios, tareas o asignaciones para fuera del aula con un porcentaje del 10%. Cada semana se manda una tarea con un número entre 5-10 ejercicios. De éstas se seleccionan dos y de cada una de ellas sólo se corrigen dos ejercicios.

10. FUENTES DE INFORMACIÓN

1. T.H. Cormen, C.E. Leiserson, R.L. Rivest & C. Stein, *Introduction to Algorithms*, The MIT Press, 3ra. edición, 2009. Capítulos 1, 2, 3 y 4; 6 y 7; 10, 11, 12 y 13; Apéndice A.
2. J.N. Ravelo, *Especificación e Implementación de Tipos Abstractos de Datos*, Universidad Simón Bolívar, 2012, disponible en <http://www ldc.usb.ve/~jravelo/docencia/algoritmos/material>.
3. J.N. Ravelo & K. Fernández, *Recursión. Reglas de Correctitud*, Universidad Simón Bolívar, 2012, disponible en <http://www ldc.usb.ve/~jravelo/docencia/algoritmos/material>.
4. J.N. Ravelo & K. Fernández, *Tipos Algebraico-Libres*, Universidad Simón Bolívar, 2012, disponible en <http://www ldc.usb.ve/~jravelo/docencia/algoritmos/material>.
5. J.N. Ravelo & K. Fernández, *Ordenamiento sobre Arreglos*, Universidad Simón Bolívar, 2012, disponible en <http://www ldc.usb.ve/~jravelo/docencia/algoritmos/material>.
6. N. Wirth, *Algorithms + Data Structures = Programs*, Prentice Hall, 1976. Capítulos 1 y 4.
7. B. Liskov con J. Guttag, *Program Development in Java – Abstraction, Specification, and Object-Oriented Design*, Addison-Wesley, 2001. Capítulos 1 al 10.
8. R. Mitchell, *Abstract Data Types and Modula-2*, Prentice Hall, 1992. Capítulos 1, 2 y 4.
9. C. Morgan, *Programming from Specifications*, Prentice Hall, 2da. edición, 1994. Capítulo 9.
10. S. Thompson, *Haskell – The Craft of Functional Programming*, Addison-Wesley, 1996. Capítulo 10.
11. A.V. Aho, J.E. Hopcroft & J.D. Ullman, *Data Structures and Algorithms*, Addison- Wesley, 1983.
12. X. Franch Gutiérrez, *Estructuras de Datos – Especificación, Diseño e Implementación*, Alfaomega, 2002.
13. L. Joyanes Aguilar, *Fundamentos de Programación – Algoritmos y Estructuras de Datos*,

McGraw-Hill, 2da. edición, 1996.

14. A. Kaldewaij, *Programming: The Derivation of Algorithms*, Prentice Hall, 1990.

15. E. Arráiz, E. Pasarella & C. Zoltan, *Tipos Abstractos de Datos y Algoritmos*, Universidad Simón Bolívar, 2001.

11. CRONOGRAMA DE ACTIVIDADES

Esta sección es un apéndice a ser desarrollado por el profesor al inicio de cada ejecución del programa, y que debe informarse a los estudiantes). Éste orienta al estudiante y al docente sobre el desarrollo de la asignatura en el tiempo. El cronograma puede ser flexible y depende entre otros factores, del período de actividades docentes.

Se presenta el siguiente cronograma como modelo, el cual pretende enfatizar las horas de práctica, ya que no se tiene un día específico para las mismas.

Semana	Día 1	Día 2
1	TEORÍA: Tipos Abstractos de Datos (TADs): introducción, especificaciones con modelos, ejemplo Diccionario.	TEORÍA: TADs: Modelos Abstractos de representación (conjuntos, multiconjuntos, secuencias, relaciones y funciones), especificación, refinamiento de datos
2	TEORÍA: TADs: invariante de acoplamiento, ejemplo Diccionario (continuación), correctitud de refinamiento de datos.	PRÁCTICA: especificaciones con modelos de los TADs Pila y Cola
3	TEORÍA: Tipos Concretos de Datos: arreglos y registros, referencias o apuntadores. Referencias: definición, operaciones, estructuras de enlace simple y doble. Ejemplo TAD Diccionario.	TEORÍA: TADs: encapsulamiento y ocultamiento, módulos y clases. Especificaciones algebraicas. Ejemplo TAD Diccionario.
4	PRÁCTICA: Implementaciones del TAD Pila y TAD Cola. Especificaciones algebraicas del TAD Pila y TAD Cola	PRUEBA ESCRITA 1
5	TEORÍA: Introducción a los algoritmos recursivos. Regla de corrección de la llamada recursiva.	TEORÍA: Tipos Algebraicos Libres: ejemplos (expresiones, listas, árboles), definición de funciones, demostración de propiedades.
6	TEORÍA: Tipos Algebraicos Libres: primitivas de programación, programación de ejemplos iterativos y recursivos	PRÁCTICA: Algoritmos recursivos y Tipos algebraicos libres
7	TEORÍA: Referencias: implementación de tipos recursivos, programación, ejemplos (listas y árboles). Diagramas de memoria.	PRUEBA ESCRITA 2
8	TEORÍA: Ejercicios sobre análisis descendente, subprogramas y pasaje de parámetros	TEORÍA: Algoritmos de Ordenamiento: inserción, análisis de complejidad.
9	TEORÍA: Algoritmos de Ordenamiento: selección, análisis de complejidad	PRÁCTICA: algoritmos iterativos de búsqueda y ordenamiento (burbuja). Analisis de complejidad.

10	TEORÍA: Algoritmos de Ordenamiento: Mezcla, Quicksort, Heapsort. Análisis de complejidad..	TEORÍA: Árboles de búsqueda: definición, propiedades, búsqueda.
11	TEORÍA: Árboles de búsqueda: inserción, eliminación	PRÁCTICA: Algoritmos de ordenamiento, Árboles de búsqueda.
12	TEORÍA: Repaso	PRUEBA ESCRITA 3