

ESTRUCTURAS DE DATOS

2016-II

Profesores:

Germán Hernández, Gr 1 y 2
Andrés Arévalo, Gr 3 y 4

e-mail: gjhernandezp@unal.edu.co

e-mail: ararevalom@unal.edu.co

Estudiante auxiliar:

Andrés Felipe Cruz Salinas

e-mail: afcruzs@unal.edu.co

Pre-requisito: 2016375 - Programación Orientada a Objetos

1. OBJETIVOS

- Entender las principales estructuras de datos, desde un punto de vista abstracto, como colecciones de datos sobre las cuales se define un conjunto de operaciones.
- Implementar, de diversas formas y en al menos un lenguaje de programación, las principales estructuras de datos, así como las operaciones que se puedan realizar sobre ellas.
- Solucionar problemas de manejo de información mediante abstracción, diseño detallado e implementación, adecuada a la situación particular que se modela.
- Producir implementaciones de las estructuras de datos y sus operaciones, contemplando aspectos de calidad tales como corrección, encapsulamiento, modularidad, eficiencia y simplicidad.
- Utilizar otros paradigmas de pensamiento para diseñar algoritmos que le permitan ampliar sus capacidades para resolver problemas.

2. METODOLOGÍA

- Presentación magistral de conceptos y métodos por parte del profesor.
- Talleres prácticos que se evalúan usando un juez en línea para poner en práctica los conceptos estudiados en la clase magistral.

3. CONTENIDO

Tema	Descripción
1	Introducción a la asignatura <ul style="list-style-type: none">• Motivación• Generalidades de Java
2	Estructuras de datos lineales <ul style="list-style-type: none">• Arreglos• Listas simplemente encadenadas• Listas doblemente encadenadas• Pilas• Colas• Arreglos dinámicos y análisis amortizado

3	Colas de Prioridad, Montículos y Conjuntos <ul style="list-style-type: none"> • Árbol binario • Árbol binario completo • Montículo (<i>Heap</i>) • Aplicación: colas de prioridad • Aplicación: heap Sort • Conjuntos (<i>Sets</i>) • Conjuntos disjuntos – implementación ingenua • Conjuntos disjuntos – implementación eficiente
4	Hashing <ul style="list-style-type: none"> • Direccionamiento directo y encadenamiento • Funciones (<i>Maps</i>) • Tablas hash: funciones hash, colisiones • Aplicación: Búsqueda de protones • Tablas hash distribuidas
5	Árboles binarios de búsqueda <ul style="list-style-type: none"> • Grafos y su representación computacional • Árboles • Recorrido de los árboles • Árboles binarios de búsqueda (<i>Binary Search Tree</i>, BST) • Árboles AVL • Árboles 2-3, Árbol 2-3-4, • Árboles Rojo-Negro

4. EVALUACION

Evaluación	Porcentaje
Parcial 1	30%
Parcial 2	30%
Talleres	40%

5. BIBLIOGRAFIA BÁSICA

- Data Structures, University of California, San Diego -Coursera
<https://www.coursera.org/learn/data-structures>
- M. A. Weiss, Data Structures and Algorithm Analysis in Java, 3rd Ed., Pearson, 2012.
- L. Joyanes and I. Zahonero. Estructuras de datos en Java, 1st Ed., McGraw-Hill, 2008.
- C. A. Shaffer, Data Structures and Algorithm Analysis, Edition 3.2 (Java Version), Virginia Tech, 2012.
- H. M. Deitel and P. J. Deitel, Java How to Program, 9th Ed., Prentice Hall, 2012.
- R. Sedgewick and K. Wayne, Algorithms, 4th Edition, Princeton University, Addison-Wesley, 2011.

6. OTROS RECURSOS:

- Udemy - Java Tutorial for Complete Beginners: <https://www.udemy.com/java-tutorial/learn/#/>
- Open Data Structures (in pseudocode) Edition 0.1Gβ Pat Morin: <http://opendatastructures.org/ods-python-screen.pdf>
- Data Structures and Algorithms in C++. Michael T. Goodrich, Roberto Tamassia, David M. Mount. <http://cpp.datastructures.net/>
- MIT Introduction to Algorithms lecture videos. Srinivas Devasadas. <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/>
- Montículos Heapsort [DISI Heapsort](#)
[Lecture 4: Heaps and Heap Sort](#) MIT Introduction to Algorithms(6.006) Fall 2011
VisuAlgo - visualising data structures and algorithms through animation
[Sorting Algorithms Demo - Java](#)
Udacity CS 215 Lesson 4: It's Who You Know Keeping track of your best friends using heaps
- [DISI HeapSort](#)
- [IPython notebook: montículos](#)
- [Sorting Comparison Demos](#)