



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

ALGORITMOS Y ESTRUCTURAS DE DATOS

Año 2014

Carrera/ Plan:

*Licenciatura en Informática
Plan 2003-07
Licenciatura en Sistemas
Plan 2003-07
Analista Programador Universitario
Plan 2007*

Año: 2°

Régimen de Cursada: *Semestral*

Carácter: *Obligatoria*

Correlativas: *Matemática 2 –
Algoritmos, Datos y Programas*

Coordinador: *Javier Díaz*

Profesores: *Schiavoni Alejandra –
Mostaccio Catalina -
Fava Laura*

Hs. semanales: 6 hs.

FUNDAMENTACIÓN

Esta materia es de gran importancia dentro de las carreras, ya que en ella se brindan los fundamentos de las estructuras de datos no lineales y del análisis de eficiencia de los algoritmos.

Los objetivos que se plantean en este curso consisten en lograr que los alumnos:

- adquieran un conocimiento exhaustivo de las principales estructuras de datos y aprendan a implementarlas en lenguaje Java, definiendo en forma eficiente sus clases y métodos;
- aprendan a analizar algoritmos y evaluar su eficiencia, utilizando un formalismo matemático para estimar el tiempo de ejecución requerido en función de la entrada de los mismos.

OBJETIVOS GENERALES:

Que los alumnos adquieran un conocimiento exhaustivo de las principales estructuras de datos y aprendan a implementarlas en forma eficiente: aprendan a analizar diferentes algoritmos de acceso y manejo a tales estructuras de datos, utilizando un formalismo matemático para estimar la eficiencia de los algoritmos.

CONTENIDOS MINIMOS:

- Estructuras de Datos no lineales
- Recursión.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

- Grafos.
- Algorítmica.
- Complejidad

PROGRAMA ANALÍTICO

1. Conceptos de la plataforma de ejecución y de desarrollo de JAVA. Conceptos básicos de Programación Orientada a Objetos: encapsulamiento, ocultamiento de información, clases e interfaces, objetos, herencia, polimorfismo, *binding* dinámico.

2.- Estructuras de datos recursivas: Listas, Árboles y Grafos. Distintas representaciones y estrategias de implementación de cada una. Resolución de problemas aplicando cada una de las estructuras. Repaso de Listas dinámicas, pilas y colas: representación, acceso y recorridos.

3.- Árboles binarios. Árboles de expresión. Árboles binarios de búsqueda. Árboles binarios de búsqueda balanceados: Árboles AVL. Recorridos ordenados (InOrden, PostOrden, PreOrden). Búsquedas. Actualización: inserción y borrado. Análisis de tiempo de ejecución de estas operaciones.

4.- Árboles generales. Distintas implementaciones. Recorridos ordenados (InOrden, PostOrden, PreOrden). Búsquedas. Actualización: inserción y borrado. Análisis de la eficiencia de cada algoritmo. Aplicaciones.

5.- Cola de prioridades. Heap binaria. Implementaciones y operaciones. Operaciones de inserción, borrado y construcción. Aplicaciones: Selección y Ordenación (Heapsort). Análisis de la eficiencia.

6.- Análisis de algoritmos. Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Modelo computacional. Concepto de tiempo de ejecución. Notación $O()$, Ω , Θ . Reglas generales para el cálculo del tiempo de ejecución. Cálculo de tiempo y orden de ejecución en algoritmos iterativos y recursivos. Comparación de distintas estrategias de diseño de algoritmos.

7.- Grafos orientados y no orientados. Grafos pesados. Distintas representaciones: Listas de Adyacencia y Matriz de Adyacencia. Definiciones básicas y conceptos fundamentales. Grafos acíclicos. Grafos conexos y dígrafos fuertemente conexos.

8.- Algoritmos de recorrido DFS y BFS. Árbol generador DFS: en grafos dirigidos y no dirigidos. Determinación de componentes conexas y fuertemente conexas. Análisis del tiempo de ejecución de los algoritmos mencionados.

9.- Ordenamiento topológico. Ejemplos de aplicación. Distintas implementaciones. Análisis de la eficiencia de cada uno.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

10.- Problema del camino mínimo: estudio de distintos casos. Su desarrollo para grafos pesados y no pesados; y grafos dirigidos y acíclicos. Algoritmos de Dijkstra y Floyd. Árbol generador mínimo. Algoritmos de Prim y Kruskal. Análisis del tiempo de ejecución de los algoritmos vistos.

METODOLOGÍA DE ENSEÑANZA

Las clases teóricas son dos veces por semana, una corresponde al dictado del contenido propiamente dicho de la materia y la otra aborda contenidos referidos a Programación Orientada a Objetos y al lenguaje Java, articulados con las estructuras de datos y los algoritmos vistos.

Las clases teóricas son explicaciones detalladas de cada tema realizadas por el profesor responsable y los alumnos intervienen realizando consultas y preguntas. Para reforzar algunos temas, se dan ejercicios que se resuelven en clase en conjunto entre el docente y los alumnos.

Las clases prácticas consisten en una explicación a cargo de un Jefe de Trabajos Prácticos, en la que se desarrollan ejercicios troncales de cada trabajo práctico a modo de ejemplo. Luego los alumnos realizan consultas a los auxiliares docentes.

Teóricos y prácticos están estrechamente vinculados, siendo éstos últimos una aplicación directa de los temas teóricos impartidos. Los trabajos prácticos se llevan a cabo en aulas equipadas con computadoras donde los ejercicios se resuelven en lenguaje Java usando un ambiente de desarrollo apropiado.

Las clases teóricas y las clases prácticas se dictan en los horarios de la mañana y de la tarde. Los alumnos eligen libremente al comienzo de la cursada el horario de teoría y de práctica al que van a asistir, sin embargo, tienen la posibilidad de asistir eventualmente a otro turno de práctica ante la imposibilidad de hacerlo en su horario.

Con anterioridad a cada una de las instancias de evaluación se ofrecen clases de consulta en función de las problemáticas más relevantes.

Las clases teóricas serán a través de presentaciones electrónicas, que contienen el desarrollo de los temas. Además, se utiliza la pizarra para realizar una explicación más detallada en caso de ser necesario, o para resolver algún ejercicio o ejemplo en forma grupal.

Se utiliza también Moodle (plataforma de aprendizaje virtual), que ofrece una funcionalidad muy útil para la organización del curso. A través de ella, se publican las clases teóricas, los trabajos prácticos y las explicaciones de los mismos, material adicional de consulta y la bibliografía. Además, se usan los *Foros* para realizar consultas, anuncios, discusiones, etc, las *Wikis* para que los alumnos trabajen en grupo, las *Tareas* para que los alumnos realicen entregas, las *Encuestas* cuando es necesario consultar a los alumnos, por ejemplo para que opten por un turno de práctica.

EVALUACIÓN

La evaluación de los trabajos prácticos consiste en una prueba práctica escrita que comprende los tres temas centrales del currículo. Es una prueba de desarrollo, donde se



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

plantean ejercicios a resolver similares a los trabajados en los prácticos. Hay dos instancias de recuperatorio, sobre el o los tema/s que no fueron aprobados en instancias anteriores.

La división en temas permite al alumno poder organizarse para estudiar cada uno de ellos y de esta manera decidir cuando está en condiciones de presentarse a la evaluación considerando las fechas propuestas y así alcanzar la aprobación de los tres temas.

El final es teórico-práctico, una parte con enunciados de opción múltiple y otra con preguntas de respuesta breve.

BIBLIOGRAFÍA OBLIGATORIA

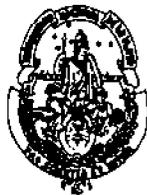
Título	Autor(es)	Editorial	Año de edición
<i>Data Structures And Algorithm Analysis in Java; 2nd Edition</i>	Mark Allen Weiss	Addison-Wesley	2007
<i>Data Structures and Algorithms</i>	A. Aho, J. Hopcroft, J. D. Ullman	Addison-Wesley	1983
<i>Thinking in Java, fourth edition.</i>	Bruce Eckel,	Prentice Hall,	2006

BIBLIOGRAFÍA COMPLEMENTARIA

Título	Autor(es)	Editorial	Año de edición
<i>Data Structures in Java; 1st Edition</i>	Thomas A. Standish	Addison-Wesley	1997
<i>Data Structures and Problem Solving using Java; 3rd Edition.</i>	Mark Allen Weiss	Addison-Wesley	2006
<i>Introduction to algorithm; third edition</i>	Thomas H. Cormen	The MIT Press	2009

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Contenidos/Actividades	Evaluaciones previstas
Semana 1	<ul style="list-style-type: none">• La Plataforma Java.• Introducción a Fundamentos de Programación Orientada a Objetos.• Aplicación: Ejecución de programas java con y sin IDE (<i>Integrated</i>	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

	<i>Development Environment).</i>	
Semana 2	<ul style="list-style-type: none">• Definición de clases en Java• Instanciación de objetos y comunicación a través de mensajes• Repaso de conceptos de Listas, pilas y colas. Listas con objetos.• Herencia y clases abstractas. Ejemplo de listas con Herencia en java.• Tipos de datos genéricos. Ejemplos.• Resolución del TP 1 – JDK y uso de Eclipse• Resolución del TP 2 – Introducción al lenguaje Java	
Semana 3	<ul style="list-style-type: none">• Repaso de Listas, definición de la estructura y operaciones• Noción intuitiva de análisis de algoritmos y complejidad.• Árboles binarios: representaciones, recorridos.• Constructores en java. Constructores y herencia.• Resolución del TP 3 – Encapsulamiento y Abstracción	
Semana 4	<ul style="list-style-type: none">• Aplicaciones de árboles binarios: árboles de expresión.• Árboles generales: ejemplos y terminología. Distintas representaciones e implementaciones. Recorridos. Aplicaciones.• Interfaces en java, la interface comparable. Ejemplos de uso de esta interface en estructuras ordenadas.• Resolución del TP 4 – Árboles Binarios	
Semana 5	<ul style="list-style-type: none">• Cola de prioridades: concepto y características. Heap binaria: propiedades e implementación. Operaciones de acceso y construcción.• Aplicaciones de Heap: Selección y Ordenación (HeapSort). Análisis de la eficiencia.• Repaso de Interfaces. Ejemplos con Heap.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Semana 6	<ul style="list-style-type: none">• Árboles binarios de búsqueda. Repaso de las operaciones. Árboles AVL: definición y representación. Concepto de Balanceo.• Árboles AVL: implementación de las operaciones. Rotaciones simples y dobles.• Paquetes y especificadores de acceso. Ejemplos de especificadores de acceso con árboles generales y árboles binarios de búsqueda.• Resolución del TP 5 – Árboles Generales	
Semana 7	<ul style="list-style-type: none">• Análisis de algoritmos. Modelo computacional. Concepto de tiempo de ejecución. Notación "Big-Oh". Reglas generales para el cálculo del tiempo de ejecución.• Polimorfismo y binding dinámico en java.• Resolución del TP 6 – Cola de Prioridades	
Semana 8	<ul style="list-style-type: none">• Cálculo en algoritmos iterativos y recursivos. Ejemplo: Subsecuencia de suma máxima• Resolución del TP 7 – ABB y AVL	
Semana 9	<ul style="list-style-type: none">• Grafos: ejemplos y terminología. Grafos orientados y no orientados. Grafos pesados. Definiciones básicas y conceptos fundamentales. Distintas representaciones: Listas y Matriz de adyacencia. Grafos acíclicos. Grafos conexos y digrafos fuertemente conexos.• Definición en java de Grafos con listas y con Matriz de adyacencia. Operaciones básicas.• Resolución del TP 8 – Análisis de algoritmos	
Semana 10	<ul style="list-style-type: none">• Algoritmos de recorrido DFS y BFS. Árbol generador DFS: en grafos dirigidos y no dirigidos. Determinación de componentes conexas y fuertemente conexas. Análisis del tiempo de ejecución de los algoritmos mencionados.	



8

UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

	<ul style="list-style-type: none">• Recorridos de grafos básicos, modificaciones a dichos algoritmos en java.	
Semana 11	<ul style="list-style-type: none">• Ordenamiento topológico. Definición. Ejemplos. Distintas implementaciones.• Análisis del tiempo de ejecución de los algoritmos mencionados.• Resolución del TP 9 – Grafos 1era Parte	
Semana 12	<ul style="list-style-type: none">• Problema del camino mínimo: introducción y estudio de distintos casos. Aplicaciones. Algoritmos para cada caso. Análisis del tiempo de ejecución. Caminos mínimos desde un origen para: grafos no pesados y grafos con pesos positivos<ul style="list-style-type: none">- Algoritmo de Dijkstra (versión original)	
Semana 13	<ul style="list-style-type: none">• Problema del camino mínimo (continuación):<ul style="list-style-type: none">- Algoritmo de Dijkstra (implementado con heap)- Caminos mínimos desde un origen para: grafos con pesos positivos y negativos y grafos dirigidos y acíclicos- Caminos mínimos entre cada par de vértices. Algoritmo de Floyd.• Resolución del TP 10 – Grafos 2da Parte	
Semana 14	<ul style="list-style-type: none">• Árbol generador mínimo. Definición. Algoritmos de Prim y Kruskal. Análisis del tiempo de ejecución de los algoritmos vistos.	
Semana 15	<ul style="list-style-type: none">• Consultas	Parcial Semana del 16/06/2014
Semana 16	<ul style="list-style-type: none">• Muestra y Consultas	
Semana 17	<ul style="list-style-type: none">• Consultas	1er Recuperatorio Semana del 30/06/2014
Semana 18	<ul style="list-style-type: none">• Muestra y Consultas	
Receso Invernal		
Semana 19	<ul style="list-style-type: none">• Consultas	2do Recuperatorio



9

UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

		Semana del 04/08/2014
--	--	-----------------------

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

- Plataforma virtual: <https://catedras.info.unlp.edu.ar/>

Firmas del/los profesores responsables: