

**ALGORITMOS Y ESTRUCTURAS DE
DATOS**

Año 2019

Carrera/ Plan:*Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07**Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07**Analista Programador Universitario Plan 2015/Plan 2007**Analista en TIC Plan 2017***Año:** 2°**Régimen de Cursada:** *Semestral***Carácter (Obligatoria/Optativa):** *Obligatoria***Correlativas:** *Matemática 2 – Taller de Programación***Profesor/es:** *Javier Díaz - Alejandra Schiavoni –
Catalina Mostaccio - Laura Fava –
Pablo Iuliano***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;
- adquieran las herramientas necesarias para evaluar y aplicar las estructuras de datos adecuadas para la resolución de problemas concretos;
- 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.

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT10- Capacidad para realizar investigaciones bibliográficas y de diferentes fuentes de información a fin de obtener conocimiento actualizado en temas de la disciplina.
- CGT4- Conocer e interpretar los conceptos, teorías y métodos matemáticos relativos a la informática, para su aplicación en problemas concretos de la disciplina.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI-CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfaces humano computador y computador-computador.
- LS-CE1 Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.



Capacidad de análisis, diseño y evaluación de interfaces humano computador y computador-computador.

CONTENIDOS MÍNIMOS (de acuerdo al Plan de Estudios)

- Estructuras de Datos no lineales
- Recursión
- 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.
- 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.

BIBLIOGRAFÍA

Título	Autor(es)	Editorial	Año de edición
<i>Data Structures And Algorithm Analysis in Java; 3rd Edition</i>	Mark Allen Weiss	Addison-Wesley	2012
<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; 4th Edition.</i>	Mark Allen Weiss	Addison-Wesley	2009
<i>Introduction to algorithm; third edition</i>	Thomas H. Cormen	The MIT Press	2009

METODOLOGÍA DE ENSEÑANZA

Las clases teóricas son dos veces por semana, en las que se dicta el contenido referido a las estructuras de datos mencionadas, análisis de complejidad de los algoritmos y conceptos referidos a Programación Orientada a Objetos y al lenguaje Java, articulados con las estructuras de datos y los algoritmos vistos.

Durante el desarrollo de la materia, se estudian en detalle estructuras de datos avanzadas, analizando distintas formas de implementación y algoritmos de acceso a dichas estructuras. Respecto de cada una de ellas, se ven tanto aspectos teóricos de modelización como prácticos de implementación usando distintas estrategias y evaluando la eficiencia de cada una. La implementación de las estructuras de datos se realiza usando el lenguaje Java, que les permite tomar contacto con tecnologías y paradigmas de programación actuales.

El tema de análisis de la complejidad de los algoritmos, es transversal a toda la materia y tiene como objetivo que los alumnos aprendan a evaluar la eficiencia de todos los algoritmos. Para lograr este objetivo, se les explica el formalismo matemático subyacente, basado en el análisis de crecimiento de funciones trigonométricas y mecanismos para encontrar cotas asintóticas.

Se estimula el análisis, diseño e implementación de soluciones a problemas del mundo real, como paso inicial al desarrollo de proyectos. Se pone énfasis en el estudio de las estructuras de datos y su aplicabilidad en problemas reales concretos resolubles desde la informática, evaluando ventajas y desventajas del uso de cada una de ellas. A partir del análisis realizado, se implementan posibles soluciones, evaluando la eficiencia de cada una.

Las clases teóricas son de tipo expositivas, se utilizan presentaciones electrónicas que comprenden 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, utilizando la pizarra.

Las clases prácticas se llevan a cabo en aulas equipadas con computadoras donde los ejercicios de implementación se resuelven en lenguaje Java usando un ambiente de desarrollo apropiado. El objetivo es que los alumnos aprendan técnicas y herramientas de aplicación en Informática, siguiendo las tendencias marcadas por los cambios tecnológicos y la aplicación efectiva de las mismas.

Los auxiliares docentes, jefes de trabajos prácticos y ayudantes, responden las consultas de los alumnos y realizan explicaciones generales en la pizarra en caso de ser necesario.

Durante el desarrollo de las clases prácticas, se organizan actividades por equipos de trabajo, con un número variable de alumnos entre 2 y 4, que constan de la resolución de diferentes problemas planteados por el docente a cargo. En la evaluación del trabajo en equipo, hay presentaciones donde se exponen las explicaciones y fundamentaciones correspondientes que permiten calificar diferentes aptitudes de los miembros del equipo, tales como conocimientos adquiridos, interés en el tema y predisposición al trabajo colaborativo.

Los contenidos teóricos y prácticos están estrechamente vinculados, siendo éstos últimos una aplicación directa de los temas teóricos impartidos. A lo largo del desarrollo de la materia, los docentes proponen la búsqueda y consulta de bibliografía actualizada sobre los temas y el planteo y discusión de diferentes soluciones y alternativas que permiten resolver cada uno de los problemas.

Las clases teóricas y las clases prácticas se dictan en horarios de mañana y de 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.

Se utiliza 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 asignatura cuenta con un régimen que permite a los alumnos promocionar la materia durante el semestre de cursada sin rendir examen final.

Durante el desarrollo de la cursada se tomará asistencia en las clases prácticas, y el alumno debe contar con un porcentaje establecido por la cátedra para acceder al régimen de promoción. Además, se implementarán entregas y presentaciones de ejercicios en fechas preestablecidas. El docente responsable de la práctica registra los resultados de las presentaciones para ser tenidas en cuenta en las evaluaciones de los alumnos. También se considerarán aspectos relacionados con los aportes realizados por los alumnos, en base a búsquedas e investigaciones llevadas a cabo sobre los temas vistos.

La evaluación de la asignatura consiste en dos parciales teórico-prácticos con puntaje ponderado, donde se plantean ejercicios a resolver similares a los trabajados en los prácticos. Los ejercicios planteados en las evaluaciones comprenden el uso de las estructuras de datos y el análisis de complejidad de los algoritmos. Consisten en aplicar las estructuras de datos vistas en la materia e implementar algoritmos para plantear una solución a un problema concreto. En la evaluación se tiene en cuenta el análisis del problema y el diseño de una solución adecuada y eficiente como punto inicial para la futura implementación de un posible proyecto global. Se evalúa no sólo la correctitud de la solución, sino también la utilización de las técnicas y lenguaje de programación como herramienta de aplicación. Además, se plantean problemas en los que el alumno debe utilizar los métodos matemáticos formales vistos en las clases para estimar el tiempo de ejecución de los algoritmos en función del tamaño de la entrada. En las evaluaciones también se incluyen preguntas sobre conceptos teóricos relativos a los temas de la currícula.

La aprobación de ambos parciales con un promedio de 6 o superior, le otorga al alumno la promoción de la materia, y la obtención de una nota entre 4 y menor que 6 puntos, le otorga la aprobación de los trabajos prácticos, debiendo el alumno rendir un examen final.

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

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	• Contenidos/Actividades
Semana 1	11/03/2019	<ul style="list-style-type: none"> • La Plataforma Java. • Revisión de Fundamentos de Programación Orientada a Objetos. • Aplicación: Ejecución de programas java con y sin IDE (<i>IntegratedDevelopmentEnvironment</i>).
Semana 2	18/03/2019	<ul style="list-style-type: none"> • Clases y objetos en Java • 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. Introducción al lenguaje Java
Semana 3	25/03/2019	<ul style="list-style-type: none"> • Repaso de conceptos de Listas, definición de la estructura y operaciones. Pilas y Colas. Listas con objetos. • Noción intuitiva de análisis de algoritmos y complejidad. • Árboles binarios: representaciones, recorridos. • Constructores en java. Constructores y herencia.
Semana 4	01/04/2019	<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 2 – Encapsulamiento y Abstracción
Semana 5	08/04/2019	<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. • Resolución del TP 3 – Árboles Binarios
Semana 6	15/04/2019	<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.

Semana 7	22/04/2019	<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 4 – Árboles Generales
Semana 8	29/04/2019	<ul style="list-style-type: none"> • Cálculo en algoritmos iterativos y recursivos. Ejemplo: Subsecuencia de suma máxima
Semana 9	06/05/2019	<ul style="list-style-type: none"> • Repaso y consultas para la evaluación
Semana 10	13/05/2019	<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 dígrafos fuertemente conexos. • Definición en java de Grafos con listas y con Matriz de adyacencia. Operaciones básicas. • Resolución del TP 5 – Análisis de algoritmos
Semana 11	20/05/2019	<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. • Recorridos de grafos básicos, modificaciones a dichos algoritmos en java.
Semana 12	27/05/2019	<ul style="list-style-type: none"> • Repaso y consultas para la evaluación
Semana 13	03/06/2019	<ul style="list-style-type: none"> • Ordenamiento topológico. Definición. Ejemplos. Distintas implementaciones. • Análisis del tiempo de ejecución de los algoritmos mencionados. • Resolución del TP 6 – Grafos 1era Parte
Semana 14	10/06/2019	<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 15	17/06/2019	<ul style="list-style-type: none"> • Problema del camino mínimo (continuación): <ul style="list-style-type: none"> ◦ Algoritmo de Dijkstra (implementado con heap)

		<ul style="list-style-type: none"> ◦ 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 7 – Grafos 2da Parte
Semana 16	24/06/2019	<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 17	01/07/2019	<ul style="list-style-type: none"> • Consultas
Semana 18	08/07/2019	<ul style="list-style-type: none"> • Muestra y Consultas
Semana 19	15/07/2019	<ul style="list-style-type: none"> • Consultas
Receso de invierno		
Semana 20	05/08/2019	<ul style="list-style-type: none"> • Consultas

Evaluaciones previstas	Fecha
Evaluación del Módulo 1	Semana 06/05/2019
Recuperatorio de la Evaluación del Módulo 1	Semana 27/05/2019
Evaluación del Módulo 2	Semana 17/06/2019
Recuperatorio de la Evaluación del Módulo 2	Semana 01/07/2019
Recuperatorio flotante	Semana 05/08/2019

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

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

Firma del/los profesor/es

**ALGORITMOS Y ESTRUCTURAS DE
DATOS****Redictado****Año 2019****Carrera/ Plan:**

Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07
Analista Programador Universitario Plan 2015/Plan 2007
Analista en TIC Plan 2017

Año: 2°**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:** Matemática 2 – Taller de Programación**Profesor/es:** Javier Díaz - Alejandra Schiavoni –

Catalina Mostaccio - Claudia Queiruga -
Pablo Iuliano

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;
- adquieran las herramientas necesarias para evaluar y aplicar las estructuras de datos adecuadas para la resolución de problemas concretos;
- 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.

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGT1- Identificar, formular y resolver problemas de Informática.
- CGT10- Capacidad para realizar investigaciones bibliográficas y de diferentes fuentes de información a fin de obtener conocimiento actualizado en temas de la disciplina.
- CGT4- Conocer e interpretar los conceptos, teorías y métodos matemáticos relativos a la informática, para su aplicación en problemas concretos de la disciplina.
- CGT5- Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Informática.
- LI-CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS-CE1 Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.



Capacidad de análisis, diseño y evaluación de interfaces humano computador y computador-computador.

CONTENIDOS MÍNIMOS (de acuerdo al Plan de Estudios)

- Estructuras de Datos no lineales
- Recursión.
- 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.
- 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.

BIBLIOGRAFÍA

Título	Autor(es)	Editorial	Año de edición
<i>Data Structures And Algorithm Analysis in Java; 3rd Edition</i>	Mark Allen Weiss	Addison-Wesley	2012
<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; 4th Edition.</i>	Mark Allen Weiss	Addison-Wesley	2009
<i>Introduction to algorithm; third edition</i>	Thomas H. Cormen	The MIT Press	2009

METODOLOGÍA DE ENSEÑANZA

Las clases teóricas son dos veces por semana, en las que se dictan el contenido referido a las estructuras de datos mencionadas y el análisis de complejidad de los algoritmos y conceptos referidos a Programación Orientada a Objetos y al lenguaje Java, articulados con las estructuras de datos y los algoritmos vistos.

Durante el desarrollo de la materia, se estudian en detalle estructuras de datos avanzadas, analizando distintas formas de implementación y algoritmos de acceso a dichas estructuras. Respecto de cada una de ellas, se ven tanto aspectos teóricos de modelización como prácticos de implementación usando distintas estrategias y evaluando la eficiencia de cada una. La implementación de las estructuras de datos se realiza usando el lenguaje Java, que les permite tomar contacto con tecnologías y paradigmas de programación actuales.

El tema de análisis de la complejidad de los algoritmos, es transversal a toda la materia y tiene como objetivo que los alumnos aprendan a evaluar la eficiencia de todos los algoritmos. Para lograr este objetivo, se les explica el formalismo matemático subyacente, basado en el análisis de crecimiento de funciones trigonométricas y mecanismos para encontrar cotas asintóticas.

Se estimula el análisis, diseño e implementación de soluciones a problemas del mundo real, como paso inicial al desarrollo de proyectos. Se pone énfasis en el estudio de las estructuras de datos y su aplicabilidad en problemas reales concretos resolubles desde la informática, evaluando ventajas y desventajas del uso de cada una de ellas. A partir del análisis realizado, se implementan posibles soluciones, evaluando la eficiencia de cada una.

Las clases teóricas son de tipo expositivas, se utilizan presentaciones electrónicas que comprenden 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, utilizando la pizarra.

Las clases prácticas se llevan a cabo en aulas equipadas con computadoras donde los ejercicios se resuelven en lenguaje Java usando un ambiente de desarrollo apropiado. El objetivo es que los alumnos aprendan técnicas y herramientas de aplicación en Informática, siguiendo las tendencias marcadas por los cambios tecnológicos y la aplicación efectiva de las mismas.

Los auxiliares docentes, jefes de trabajos prácticos y ayudantes, responden las consultas de los alumnos y realizan explicaciones generales en la pizarra en caso de ser necesario.

Durante el desarrollo de las clases prácticas, se organizan actividades por equipos de trabajo, con un número variable de alumnos entre 2 y 4, que constan de la resolución de diferentes problemas planteados por el docente a cargo. En la evaluación del trabajo en equipo, hay presentaciones donde se exponen las explicaciones y fundamentaciones correspondientes que permiten calificar diferentes aptitudes de los miembros del equipo, tales como conocimientos adquiridos, interés en el tema y predisposición al trabajo colaborativo.

Los contenidos teóricos y prácticos están estrechamente vinculados, siendo éstos últimos una aplicación directa de los temas teóricos impartidos. A lo largo del desarrollo de la materia, los docentes proponen la búsqueda y consulta de bibliografía actualizada sobre los temas y el planteo y discusión de diferentes soluciones y alternativas que permiten resolver cada uno de los problemas.

Las clases prácticas se dictan en diferentes turnos. Los alumnos eligen libremente al comienzo de la cursada el turno 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.

Se utiliza 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 asignatura cuenta con un régimen de promoción que permite a los alumnos promocionar la materia durante el semestre de cursada sin rendir examen final.

Durante el desarrollo de la cursada se tomará asistencia en las clases prácticas, y el alumno debe contar con un porcentaje establecido por la cátedra para acceder al régimen de promoción. Además, se implementarán entregas de ejercicios en fechas preestablecidas. El docente responsable de la práctica registra los resultados de las presentaciones para ser tenidas en cuenta en las evaluaciones de los alumnos. También se considerarán aspectos relacionados con los aportes realizados por los alumnos, en base a búsquedas e investigaciones llevadas a cabo sobre los temas vistos.

La evaluación de la asignatura consiste en dos parciales teórico-prácticos con puntaje ponderado, donde se plantean ejercicios a resolver similares a los trabajados en los prácticos. Los ejercicios planteados en las evaluaciones comprenden el uso de las estructuras de datos y el análisis de complejidad de los algoritmos. Consisten en aplicar las estructuras de datos vistas en la materia e implementar algoritmos para plantear una solución a un problema concreto). En la evaluación se tiene en cuenta el análisis del problema y el diseño de una solución adecuada y eficiente como punto inicial para la futura implementación de un posible proyecto global. Se evalúa no sólo la correctitud de la solución, sino también la utilización de las técnicas y lenguaje de programación como herramienta de aplicación. Además, se plantean problemas en los que el alumno debe utilizar los métodos matemáticos formales vistos en las clases para estimar el tiempo de ejecución de los algoritmos en función del tamaño de la entrada. En las evaluaciones también se incluyen preguntas sobre conceptos teóricos relativos a los temas de la currícula.

La aprobación de ambos parciales con un promedio de 6 o superior, le otorga al alumno la promoción de la materia, y la obtención de una nota entre 4 y 6 puntos, le otorga la aprobación de los trabajos prácticos, debiendo el alumno rendir un examen final.

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

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
Semana 1	19/08/2019	<ul style="list-style-type: none"> ● La Plataforma Java. ● Revisión de Fundamentos de Programación Orientada a Objetos. ● Aplicación: Ejecución de programas java con y sin IDE (<i>Integrated Development Environment</i>).
Semana 2	26/08/2019	<ul style="list-style-type: none"> ● Clases y objetos en Java ● 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. Introducción al lenguaje Java
Semana 3	02/09/2019	<ul style="list-style-type: none"> ● Repaso de conceptos de Listas, definición de la estructura y operaciones. Pilas y Colas. Listas con objetos. ● Noción intuitiva de análisis de algoritmos y complejidad. ● Árboles binarios: representaciones, recorridos. ● Constructores en java. Constructores y herencia.
Semana 4	09/09/2019	<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 2 – Encapsulamiento y Abstracción
Semana 5	16/09/2019	<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. ● Resolución del TP 3 – Árboles Binarios
Semana 6	23/09/2019	<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.
Semana 7	30/09/2019	<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 4 – Árboles Generales

Semana 8	07/10/2019	<ul style="list-style-type: none"> ● Cálculo en algoritmos iterativos y recursivos. Ejemplo: Subsecuencia de suma máxima
Semana 9	14/10/2019	<ul style="list-style-type: none"> ● Repaso y consultas para la evaluación
Semana 10	21/10/2019	<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 dígrafos fuertemente conexos. ● Definición en java de Grafos con listas y con Matriz de adyacencia. Operaciones básicas. ● Resolución del TP 5 – Análisis de algoritmos
Semana 11	28/10/2019	<ul style="list-style-type: none"> ● Repaso y consultas para la evaluación
Semana 12	04/11/2019	<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. ● Recorridos de grafos básicos, modificaciones a dichos algoritmos en java.
Semana 13	11/11/2019	<ul style="list-style-type: none"> ● Ordenamiento topológico. Definición. Ejemplos. Distintas implementaciones. ● Análisis del tiempo de ejecución de los algoritmos mencionados. ● Resolución del TP 6 – Grafos 1era Parte
Semana 14	18/11/2019	<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 15	25/11/2019	<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 7 – Grafos 2da Parte
Semana 16	02/12/2019	<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 17	09/12/2019	• Consultas
Semana 18	16/12/2019	• Muestra y Consultas
Receso		
Semana 20	03/02/2020	• Consultas

Evaluaciones previstas	Fecha
Evaluación del Módulo 1	Semana 14/10/2019
Recuperatorio de la Evaluación del Módulo 1	Semana 28/10/2019
Evaluación del Módulo 2	Semana 25/11/2019
Recuperatorio de la Evaluación del Módulo 2	Semana 09/12/2019
Recuperatorio flotante	Semana 03/02/2020

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

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

Firma del/los profesor/es