

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

ESTRUCTURAS DE DATOS

CÓDIGO: 7655

ÁREA N°: I

CARRERAS

Ingeniería en Sistemas de Información
Licenciatura en Ciencias de la Computación

PROFESOR RESPONSABLE:

Dr. Sergio Alejandro Gómez – Profesor Adjunto con Dedicación Exclusiva

CARGA HORARIA	Teoría 64hs	Práctica 50 hs	Laboratorio 14 hs	CANTIDAD DE SEMANAS	16
---------------	----------------	-------------------	----------------------	---------------------	----

CORRELATIVAS

PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA	
APROBADAS - Resolución de Problemas y Algoritmos	CURSADAS - Análisis Matemático I - Introducción a la Programación Orientada a Objetos	APROBADAS - Análisis Matemático I - Introducción a la Programación Orientada a Objetos	CURSADAS

DESCRIPCIÓN

Estructuras de Datos es el tercero de los cuatro cursos de programación de los planes de estudio de las carreras *Licenciatura en Ciencias de la Computación*, *Ingeniería en Computación* e *Ingeniería de Software*. El **objetivo general** de las materias de Programación es que los alumnos adquieran la capacidad para desarrollar aplicaciones de software de pequeña y mediana escala. El **objetivo específico** de Estructuras de Datos es que el alumno aprenda con detalle los conceptos de estructuración de datos y los algoritmos para el manejo de las diferentes estructuras de datos que se presentan. Este proceso se contextualiza en el marco de la programación orientado a objetos y atendiendo a los factores de calidad del software (es especial la eficiencia caracterizada por la noción de orden de tiempo de ejecución de un algoritmo).

COMPETENCIAS PREVIAS Y A DESARROLLAR

El curso asume que el alumno es capaz de realizar por si mismo una aplicación Java que involucre la definición de un conjunto de clases estructurada en una jerarquía de herencia con una interfaz de usuario gráfica y/o de línea de comando. El curso se inicia con una breve recapitulación de los conceptos fundamentales de la programación orientada a objetos y del lenguaje Java estudiados en el curso de programación anterior, los cuales son utilizados para introducir nuevos conceptos como las excepciones, las interfaces y la genericidad paramétrica. Luego se presentan las estructuras de datos en orden creciente de complejidad: lineales (listas, pilas y colas), colas con prioridad, asociativas (con tablas de hash), jerárquicas (árboles generales y binarios), asociativas (con árboles de búsqueda), y grafos. La etapa final está destinada a la aplicación de las estructuras de datos más

SAP

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		2 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ESTRUCTURAS DE DATOS	CÓDIGO: 7655	
	ÁREA N°: I	

adecuadas para la representación de distintos tipos de conjuntos, organización de archivos y para el reordenamiento de sucesiones.

Además de construir conocimiento relacionado con los contenidos conceptuales que se describen en el programa, se espera que los alumnos desarrollen competencias para:

- familiarizarse con un nuevo entorno de programación;
- estudiar en forma autónoma nuevos aspectos de un lenguaje de programación que conocen;
- buscar, analizar, entender y adaptar código que no es de su autoría;
- mejorar la expresión oral y escrita;
- redactar informes en forma correcta;
- generar documentación de código en formato Javadoc;
- estudiar en forma autónoma una nueva estructura de datos;
- elegir la mejor implementación de un tipo abstracto de datos en función de los requerimientos que se desean priorizar, y, poder justificar adecuadamente las elecciones realizadas.

METODOLOGÍA DE ENSEÑANZA

El enfoque adoptado en las cuatro materias de Programación es aprender *desde* la resolución de problemas, concibiendo a la programación como un proceso en el cual cada etapa tiene un rol relevante. Los **contenidos** de la materia incluyen **conceptos**, los **recursos** que brinda el lenguaje de programación para soportar estos conceptos, los **problemas** que permiten ilustrar la aplicación de los conceptos y los **criterios** que orientan la selección de conceptos y recursos y la comparación de soluciones. La enseñanza se divide en clases teóricas, prácticas y de laboratorio.

ACTIVIDADES TEORICAS Y PRÁCTICAS

La distinción entre las clases teóricas, prácticas y de laboratorio se refiere más a los **roles** que asumen en cada caso los docentes y los alumnos, que al tipo de contenido abordado.

En las clases **teóricas** el enfoque es expositivo a pesar de que el docente estimula la participación de los alumnos. Cada contenido es introducido a nivel conceptual, luego se analiza la implementación en el lenguaje de programación y luego se presentan problemas típicos de aplicación de dicho concepto seguido por un análisis de la eficiencia de los algoritmos presentados para resolver los problemas planteados. Así se tratan las diferentes estructuras, conjuntamente con los algoritmos para su creación, acceso y modificación se presentan como recursos para la implementación de tipos abstractos de datos. En todos los casos se hace el análisis del tiempo de ejecución para el peor caso. También se analiza la representación en memoria de las distintas alternativas.

El proceso constructivo es conducido por el docente con el objetivo principal de ilustrar conceptos; sin embargo, la interacción con los alumnos está siempre presente. Durante el desarrollo de la clase se intercala el uso de presentaciones de diapositivas para presentar

SAP

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

ESTRUCTURAS DE DATOS

CÓDIGO: 7655

ÁREA N°: I

conceptos y enunciados de problemas, el pizarrón para la construcción de las soluciones y la proyección de la ejecución las soluciones implementadas. Cuando es considerado necesario las clases pueden dictarse enteramente en el pizarrón.

En las clases prácticas se asume un rol activo por parte del alumno. La acción se orienta a hacer las actividades propuestas, pero también a solicitar ayuda cuando se produce un bloqueo en el desarrollo de estas actividades y consultar cuestiones que le permitan aplicar en la práctica los contenidos presentados en teoría. Aunque en algunas clases prácticas puede adoptarse en algún momento un enfoque expositivo, la acción docente también está orientada a atender las demandas de los alumnos. En general, la idea es mostrar la solución completa de algunos de los problemas típicos y brindar lineamientos que guíen el proceso de resolución o pautas que permitan salir del bloqueo para problemas medios o avanzados, partiendo siempre que sea posible, de las ideas o soluciones parciales propuestas por los alumnos.

La expectativa es que en las clases prácticas se trabaje fundamentalmente en las etapas de interpretación del enunciado y diseño de la solución. En las clases de laboratorio, el alumno tiene también un rol activo, pero las acciones se realizan con la computadora. Se trabaja fundamentalmente en las etapas de implementación y verificación.

Los problemas han sido seleccionados con diferentes criterios. En general se orientan hacia la solución de un problema partiendo generalmente de una estructura de datos propuesta con el objetivo de desarrollar un algoritmo eficiente implementado en Java. Finalmente se debe evaluar la solución propuesta en términos de su orden de tiempo de ejecución. Otros ejercicios apuntan a que el alumno lea una porción de código, detecte errores en la misma y brinde soluciones sobre cómo modificarlo para hacerlo correcto respecto de una especificación inicial. En la práctica se implementa también un proyecto de programación que sirve para reforzar los conceptos presentados en la teoría y en los trabajos prácticos.

El programa de la materia, los trabajos prácticos, el mecanismo y cronograma de evaluación así como las dispositivas de teoría están disponibles en la página web de la materia.

MECANISMOS DE EVALUACIÓN

Los alumnos son evaluados de la siguiente manera:

- Durante el cursado de la materia, a través de:
 - Dos instancias escritas (con un recuperatorio individual o general) en la que se valoran capacidades para definir diferentes estructuras de datos y construir algoritmos para su creación, consulta y modificación.
 - La realización de un proyecto de software en forma colaborativa a partir de lo que se evalúan habilidades para programar y elaborar informes.
- Posterior al cursado de la materia: Con un examen final que consta de una instancia escrita a través de la cual se busca detectar el nivel de comprensión de los contenidos del currículum.

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		4 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ESTRUCTURAS DE DATOS	CÓDIGO: 7655	
	ÁREA N°: I	

PROGRAMA SINTÉTICO

- Programación Orientada a Objetos. El lenguaje de programación Java.
- Tipos de datos. Tipos abstractos de datos. Estructuras de Datos. Algoritmos y Estructuras de Datos. Tipos de datos recursivos. Estructuras enlazadas. Estructuras dinámicas. Representación de datos en memoria. Estrategias de implementación. Manejo de memoria en ejecución.
- Problemas, modelos, algoritmo y programas. Diseño y análisis de algoritmos. Tiempo de ejecución de un programa. Notación $O()$.
- Colecciones con modelo secuencial. Listas, Pilas, Colas. Definición de los correspondientes tipos abstractos de datos. Aplicaciones e implementación. Listas. Diferentes estructuras de datos para representar listas. Un Tipo Abstracto de Datos lista. Algoritmos fundamentales: Recorrido, búsqueda y actualización. Aplicaciones. Iteradores. Definición de un tipo abstracto de datos iterador.
- Estructuras de Datos jerárquicas. Árboles. Árboles binarios. Conceptos y aplicaciones. Tipos abstractos de datos para los modelos árbol y árbol binario. Implementaciones.
- Colecciones con modelo conjuntista. Conjuntos. Tipos abstractos de datos con modelo conjuntista. Diccionarios, colas con prioridad y mapeos. Representaciones.
- Estructuras avanzadas para representar conjuntos. Tablas Hash. Árboles de búsqueda. Árboles balanceados. Árboles de recuperación. Árboles m-arios. Árboles parcialmente ordenados.
- Estructuras de Datos no lineales y no jerárquicas. Grafos dirigidos y no dirigidos: Conceptos básicos. Implementación. Aplicaciones. Definición de tipos abstractos de datos Grafo.
- Métodos de ordenamiento de conjuntos. El modelo de ordenamiento interno. Clasificaciones. Algoritmos de ordenamiento. Análisis de los algoritmos.
- Medios de almacenamiento externo. Archivos. Clasificación. Organización. Acceso. Aplicaciones: Implementación de archivos.

PROGRAMA ANALÍTICO

- 1) Programación Orientada a Objetos. Revisión de los conceptos fundamentales y su relación con los diferentes factores en la calidad del software. Definición de un lenguaje algorítmico. El lenguaje de implementación Java: aspectos sintácticos, semánticos y pragmáticos. Genericidad paramétrica. Tipos Envoltorio. Interfaces. Excepciones, captura y manejadores.
- 2) Conceptos básicos de estructuración de datos. Tipos de datos, estructuras de datos y tipos abstractos de datos. Tipos de datos recursivos. Aplicaciones. Estructuras dinámicas. Estructuras enlazadas. Los conceptos básicos de estructuración de datos en el contexto del paradigma Orientado a Objetos. Descripción mediante un lenguaje de diseño. Implementación. Representación de datos en memoria. Estrategias de implementación.
- 3) Problemas, modelos, algoritmos y programas. Diseño y análisis de algoritmos. Tiempo de

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

ESTRUCTURAS DE DATOS

CÓDIGO: 7655

ÁREA N°: I

ejecución de un algoritmo y de un programa. Evaluación del tiempo de ejecución. La tasa de crecimiento. Evaluación asintótica. Notaciones $O()$, $\Omega()$ y $\Theta()$. Cálculo del tiempo de ejecución de un programa para el caso más desfavorable. Manejo de memoria en ejecución.

4) Colecciones con modelo secuencial: Pilas y colas. Definición de los correspondientes tipos abstractos de datos. Aplicaciones. Implementaciones de pilas: con arreglos y con listas enlazadas. Implementaciones de colas: con arreglos, cola circular y con listas enlazadas. Estructuras de datos que combinan los comportamientos de las pilas y las colas.

5) Colecciones con modelo secuencial: Listas. Estructuras lineales básicas. Definición. Ejemplos de aplicación. Un tipo abstracto de datos lista. Implementaciones con arreglos y con enlaces. Listas doblemente enlazadas y listas circulares. Análisis comparativo de las distintas representaciones. Algoritmos fundamentales: recorrido, búsqueda y actualización. Aplicaciones.

6) Recorrido de una estructura de datos. Mecanismos de abstracción para recorrer estructuras de datos. Iteradores. Definición de un tipo abstracto de datos Iterador. Recursos que provee Java para iterar. Iteración sobre estructuras de datos lineales.

7) Estructuras de datos jerárquicas. Árboles. Conceptos y terminología básica. Aplicaciones. Árboles ordenados. Un tipo de datos abstracto Árbol. Implementación con una estructura de datos enlazada. Recorridos. Árboles binarios: conceptos y recorridos. Implementaciones con arreglos y con enlaces. Aplicaciones. Árboles de expresión. Iteradores sobre árboles.

8) Colecciones con modelo conjuntista. Conjuntos. El tipo abstracto de datos conjunto. Estructuras de datos para representar conjuntos. Tipos de datos abstractos con modelos conjuntistas: Colas con prioridad, diccionarios y mapeos. Aplicaciones e implementaciones. Análisis comparativo de las diferentes representaciones.

9) Estructuras de Datos avanzadas. Estructuras avanzadas para representar conjuntos: Tablas Hash. Árboles completos y heaps. Árboles binarios de búsqueda. Árboles balanceados por altura, AVL, 2-3. Árboles de recuperación (TRIE). Árboles m-arios, B-árboles. Estructuras compuestas: composición de estructuras de datos a partir de estructuras básicas.

10) Estructuras de Datos no lineales y no jerárquicas. Grafos dirigido y no dirigidos. Conceptos y terminología básica. Clasificaciones: dirigidos y no dirigidos, rotulados y no rotulados, con y sin aristas paralelas. Recorridos en profundidad y en anchura. Representaciones con matrices y con listas. Análisis comparativo de las diferentes implementaciones. Un tipo abstracto de datos grafo. Aplicaciones.

11) Métodos de ordenamiento de conjuntos. Algoritmos de ordenamiento. Análisis de los algoritmos: selección, inserción, burbuja (intercambio), merge-sort, quick-sort y heap-sort.

12) Medios de almacenamiento externo. Clasificación. Organización. Acceso. Archivos. Diferentes formas de organización: secuenciales, con índices y de acceso directo. Estructuras de datos aptas para su organización. Aplicaciones: Implementación de archivos.

UNIVERSIDAD NACIONAL DEL SUR BAHÍA BLANCA		6 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
ESTRUCTURAS DE DATOS	CÓDIGO: 7655 ÁREA N°: I	

BIBLIOGRAFÍA

Bibliografía Básica

- Goodrich, Michael and Tamassia, Roberto. *Data Structures and algorithms in Java, 4th edition*. John Wiley Sons, Inc. 2006.
- Sergio Alejandro Gómez. *Estructuras de Datos: Notas del curso*, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, 2013-2015.

Bibliografía Adicional

- Aho, A. V., Hopcroft, J. E. and Ullman, J. D. *Data Structures and Algorithms*. Addison Wesley, 1983.
- Dale, N., Joyce, D. and Weems, Ch. *Object-oriented Data Structures Using Java*. Jones & Bartlett Publishers, 2006.
- Drake, Peter. *Data structures and algorithms in Java*. Prentice Hall, 2006.
- Ford, William and Topp William. *Data Structures with Java*. Prentice Hall, 2004.
- Goodrich, M., Tamassia, R. and Goldwasser, M. *Data Structures & Algorithms in Java*. Sixth Edition. Wiley, 2014.
- Lewis, John and Chase, Joseph. *Estructuras de datos con Java: Diseño de estructuras y algoritmos*. Ed. Pearson, 2006.
- Weiss, Mark Allen. *Data Structures and Algorithm Analysis in Java*. 2nd edition, 2006.
- Weiss, Mark Allen. *Data Structures and Problem Solving Using Java*. 3rd edition, 2005.
- Weiss, Mark Allen. *Data Structures and Algorithm Analysis in Java*. Third Edition. Addison-Wesley Pearson Education, Inc. 2012.
- Weiss, Mark Allen. *Data Structures and Algorithm Analysis in Java*. Third Edition. Pearson, 2012.

AÑO 2019		FIRMA PROFESOR RESPONSABLE 	
VISADO			
COORDINADOR ÁREA	SECRETARIO ACADÉMICO		DIRECTOR DEPARTAMENTO
	 Dr. DIEGO E. MARTINEZ SECRETARIO ACADÉMICO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR		 Dr. MARCELO A. FALAPPA DIRECTOR DECANO DPTO. DE CS. E ING. DE LA COMP. UNIVERSIDAD NACIONAL DEL SUR