



unab

**UNIVERSIDAD NACIONAL
GUILLERMO BROWN**

PROGRAMA

Estructuras de Datos

--00271--

[Plantel Docente]

1. Cantidad de horas semanales y totales

Horas semanales: 4 hs. (cuatro horas)

Horas totales: 64 hs. (sesenta y cuatro horas)

2. Nombres de las/los integrantes del equipo docente

Profesor Adjunto:

Jefe de Trabajos Prácticos (JTP):

3. Fundamentación

Esta materia es de carácter fundacional en ella se brindan los fundamentos de las estructuras de datos complejas, y el análisis de la eficiencia de los algoritmos.

Los objetivos de esta materia consisten en lograr que los alumnos:

- a) adquieran un conocimiento exhaustivo de las principales estructuras de datos y su implementación en el lenguaje [PYTHON], definiendo en forma eficiente sus clases y métodos;
- b) adquieran las herramientas necesarias para evaluar y aplicar las estructuras de datos adecuadas para la resolución de problemas concretos;
- c) aprendan a analizar algoritmos y evaluar su eficiencia, utilizando un formalismo matemático para estimar el tiempo de ejecución (complejidad computacional) requerido en función de la entrada de los mismos.

4. Programa sintético

La materia abordará aspectos relevantes de los distintos tipos de estructuras de datos, acompañado incrementalmente la introducción de la complejidad computacional intrínseca a cada una de ellas.

Los temas a abordar serán:

Unidad 1: Lenguaje de programación [PYTHON]. Encapsulamiento, clases e interfaces.

Unidad 2: Estructuras de datos recursivas,

Unidad 3: Árboles binarios.

Unidad 4: Árboles generales.

Unidad 5: Cola de prioridades. Heap binaria.

Unidad 6: Análisis de algoritmos.

Unidad 7: Grafos.

Unidad 8: Algoritmos de recorrido DFS y BFS.

Unidad 9: Ordenamiento topológico.

Unidad 10: Problemas NP. Problema del camino mínimo.

5. Objetivos

Que los alumnos/as adquieran un conocimiento exhaustivo de las principales estructuras de datos, 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.

Los objetivos a alcanzar propuestos son, que el alumno/a:

- Entienda la diferencia entre acceso aleatorio y acceso secuencial. Conozca la idea de interface de una estructura de datos, y sea capaz de utilizarla productivamente para la solución de problemas.
- Conozca la interface de distintas estructuras de datos básicas (pilas, colas, listas, arboles, hashing, etc.) y las utilice adecuadamente.
- Comprenda y utilice la noción de estructura contenedora, y la capacidad de realizar combinaciones complejas utilizándolas.
- Se familiarice con las nociones de ámbito y de pasaje de parámetros por valor o referencia.
- Maneje alguno de los principios básicos de diseño de interfaces de una estructura de datos (separación en constructores e inspectores de una interfaz, ecuaciones entre combinaciones de constructores, etc.), y pueda reconocerlos en situaciones prácticas junto con su utilidad.
- Comprenda el concepto de asignación dinámica de memoria, y pueda hacer programas que hagan un manejo dinámico explícito de memoria en forma adecuada.
- Entienda la noción de implementación de una estructura de datos, y de su eficiencia y sea capaz de implementar las interfaces vistas anteriormente con distintas alternativas variadas en eficiencia.
- Pueda resolver problemas mediante programas recursivos, y entienda la diferencia entre una resolución recursiva y otra iterativa.

6. Propósitos de la enseñanza

Nos proponemos mediante las distintas herramientas pedagógicas lograr que los estudiantes comprendan los aspectos fundamentales de las organizaciones y la administración de la información dentro de ellas.

7. Contenidos

Unidad 1: Encapsulamiento e Interfaces.

Revisión de los conceptos del lenguaje de programación [PYTHON]. Conceptos básicos de Programación Orientada a Objetos: encapsulamiento, clases e interfaces, y objetos.

Unidad 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, implementación, acceso y recorridos.

Unidad 3: Árboles Binaros.

Árboles binarios. Árboles de expresión. Recorridos ordenados (InOrden, PostOrden, PreOrden). Construcción de árboles. Búsquedas. Actualización: inserción y borrado. Análisis de tiempo de ejecución de estas operaciones.

Unidad 4: Árboles Generales.

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

Unidad 5: Cola de Prioridades (Heap).

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

Unidad 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.

Unidad 7: Grafos.

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.

Unidad 8: Algoritmos de Recorrido.

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.

Unidad 9: Ordenamiento.

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

Unidad 10: Problemas NP.

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.

8. Bibliografía y recursos audiovisuales

Bibliografía Obligatoria:

- [Joyanes, 2003] Joyanes Aguilar, L. (2003). Fundamentos de Programación; Algoritmos, Estructuras de Datos y Objetos (3a. ed.). Madrid: McGraw-Hill Interamericana.
- [Cormen et al., 2017] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2017). Introduction to algorithms.
- [Aho et al., 1998] Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1998). Estructuras de datos y algoritmos.
- [Goodrich et al., 2018] Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2018). Data structures and algorithms in Python.
- [Burin et al., 2017] Ing. Daniel Burin y Docentes de la cátedra 75.01/ 95.01 Computación (ed. 2017). Departamento de Computación, Facultad de Ingeniería, Universidad de Buenos Aires. "Algoritmos y Programación I Aprendiendo a programar usando Python como herramienta" <http://materias.fi.uba.ar/7501/apunte%20PYTHON.pdf>
- [Lee et al., 2015] Lee, K. D., & Hubbard, S. (2015). Data Structures and Algorithms with Python. Cham: Springer International Publishing.

Bibliografía Complementaria:

- [Weiss, 2010] Weiss, M. A. (2010). Data structures & problem solving using Java. Boston: Pearson Education.
- [Drozdek, 2007] Drozdek, A. (2007). Estructura de Datos Y Algoritmos en Java. Cengage Learning Latin America.
- [Brassard et al., 2008] Brassard, G., & Bratley, P. (2008). Fundamentos de algoritmia. Madrid: Pearson Prentice Hall.
- [Heilman, 2010] Heileman, Gregory, L. (2010). Estructura de datos, algoritmos y programación orientada a objetos. La Habana: Editorial Félix Varela.
- [Gamma et al., 1995] Gamma, E., Helm, R., Johnson, R. E., & Vlissides, J. (1995). Design patterns: Elements of reusable object-oriented software. Reading, Mass: Addison-Wesley.

9. Metodología

Los contenidos se abordarán utilizando los siguientes mecanismos pedagógicos:

- Distribución de material bibliográfico para las distintas unidades presentadas.
- Encuentros sincrónicos donde el docente presentará y abordará los contenidos.
- Dinámicas disparadoras dentro de los encuentros sincrónicos para evaluar conocimientos previos de los estudiantes.
- Trabajos prácticos a desarrollar en equipos de trabajo previamente conformados.
- Puesta en común de los trabajos prácticos entregados.
- Se acompañará el proceso de enseñanza-aprendizaje con videos relacionados a los contenidos dados y uso de casos de organizaciones reales.

10. Uso del campus virtual e integración de TIC en la propuesta pedagógica

La UNaB comenzó a implementar su propio campus virtual, dentro de el se subirá la bibliografía de cada unidad, se habilitará un foro para consultas generales, y se subirá material adicional de cada tema. También, será el medio para la visualización y entrega de los trabajos prácticos y el medio oficial de comunicación entre el cuerpo docente y los estudiantes.

11. Evaluación

A. Requisitos de aprobación

Situación de regularidad

Asistencia mínima a los encuentros virtuales del 70%

Aprobación de TP requeridos

Aprobación de los 2 parciales o sus recuperatorios con 4 (cuatro o más)

Situación de promoción

Asistencia mínima a los encuentros virtuales del 70%

Aprobación de TP requeridos

Aprobación de los 2 parciales o sus recuperatorios con 6 (seis o más). No se promedia la nota.

B. Criterios de evaluación

Para los Trabajos prácticos los criterios serán:

- Completitud de las consignas solicitadas
- Adecuación de las respuestas a lo desarrollado en clases
- Adecuación de las respuestas a la bibliografía de la materia
- Trabajo entregado dentro de las fechas establecidas

Para los exámenes los criterios serán:

- Completitud de las consignas solicitadas
- Adecuación de las respuestas a la bibliografía de la materia
- Creación propia en las respuestas
- Ejemplificación de los conceptos teóricos

C. Formatos de la evaluación de las distintas instancias

Para los trabajos prácticos se utilizarán los diferentes formatos disponibles en el Aula virtual.

Para el caso de los exámenes serán 4 consignas a desarrollar por examen.

12. Cronograma de actividades / Planificación de clases

Las clases se desarrollarán los días Jueves de 19 a 23 hs. En encuentros sincrónicos vía Google Meet/Zoom y posterior aplicación práctica a desarrollar en la semana siguiente. Cada jueves se hará una puesta en común de los trabajos presentados y se dará introducción al nuevo Trabajo Práctico.

Cronograma		
Semana		Tema
1	TEORIA	
2	CLASE TEORICO-PRACTICA	
3	CLASE TEORICO-PRACTICA	
4	CLASE TEORICO-PRACTICA	

5	CLASE TEORICO-PRACTICA	
6	CLASE TEORICO-PRACTICA	
7	PARCIAL	1er PARCIAL
8	CLASE TEORICO-PRACTICA	
9	CLASE TEORICO-PRACTICA	
10	CLASE TEORICO-PRACTICA	
11	CLASE TEORICO-PRACTICA	
12	CLASE TEORICO-PRACTICA	
13	PRACTICA	
14	PARCIAL	2do EXAMEN PARCIAL
15	RECUPERATORIO	Recuperatorios 1er y 2do Parcial
16		CIERRE DE NOTAS