

Programa Regular de asignatura

- **Asignatura:** Complejidad Temporal, Estructura de Datos y Algoritmos.
- **Carrera/s:** Ingeniería en Informática
- **Ciclo lectivo:** 2018
- **Docente/s:**
 - Coordinador:** Lic. Pablo Iuliano
 - Plantel Docente:**
 - Dr. Ing. Leonardo Amet
 - Ing. Matías Suarez
 - Ing. Bernardo Pagola
- **Carga horaria semanal:** 4 hs.

Fundamentación:

Complejidad Temporal, Estructura de Datos y Algoritmos es una materia obligatoria correspondiente al segundo año de la carrera de Ingeniería en Informática.

En la materia los estudiantes completarán y profundizarán los conocimientos básicos de la informática vistos en la materia de primer año Fundamentos de Programación y la materia de segundo año Algoritmos y Programación con los conocimientos descriptos en las unidades temáticas del presente programa regular. Así mismo las técnicas de programación aprendidas serán la base para el dictado de materias más avanzadas dentro de la carrera como: Metodologías de Programación I, Metodologías de Programación II, Ingeniería de Software I, Lenguajes Formales y Autómatas, Programación en Tiempo Real etc.

Objetivos:

Los objetivos de Complejidad Temporal, Estructura de Datos y Algoritmos son brindar al estudiante las técnicas algorítmicas que le permitirán abordar soluciones computacionales correctas y eficientes a fin de resolver problemas no triviales junto con los conocimientos teóricos-prácticos, habilidades, experiencias y sentido crítico, fundamentados en teorías y técnicas sólidas, comprobadas y bien establecidas.

Contenidos mínimos:

Presentación del concepto de complejidad temporal. Expresar un algoritmo en término de $T(n)$. Cálculo del $T(n)$ para algoritmos iterativos y recursivos. Uso de tipos genéricos de datos para la resolución de problemas. Abstracción de datos mediante la especificación de interfaces comunes, herencia e implementaciones intercambiables entre sí. Presentación de las estructuras de datos: listas, pilas, colas, colas de prioridad, árboles y grafos. Implementaciones de las estructuras de datos: listas, colas, pilas, árboles y grafos. Algoritmos fundamentales: recorridos en arboles y grafos, caminos mínimos en grafos, árboles de expansión.

Contenidos Temáticos o Unidades: Unidad I

Análisis de Algoritmos. Eficiencia, introducción al concepto $T(n)$, casos, notación asintótica Big-Oh, regla de la suma y del producto, cálculo del $T(n)$ para algoritmos iterativos.

Unidad II

Análisis de Algoritmos. Eficiencia, cálculo del $T(n)$, casos, notación asintótica Big-Oh para algoritmos recursivos.

Unidad III

Estructuras Arbóreas. Definición, descripción, terminología, representaciones, recorridos y tiempo de ejecución de los mismos para Arboles Generales, Arboles Binarios, Arboles Binarios de Búsqueda, Arboles AVL.

Unidad IV

Colas de Prioridades. Definición, descripción, terminología, representaciones y tiempo de ejecución de las mismas. Implementación de colas de prioridades mediante Heap.

Unidad V

Hash. Definición, descripción, terminología, representaciones y tiempo de ejecución de las mismas. Implementación.

Unidad VI

Grafos. Terminología, representación con lista y matriz de adyacencias, recorridos DFS y BFS, sort topológico, caminos de costo mínimo y árbol de expansión mínimo.

Unidad VII

Encapsulamiento y Abstracción. Especificación de las interfaces e implementaciones intercambiables.

Bibliografía:

Chang, S. K. (2003). Data structures and algorithms (Vol. 13). World Scientific. Unidad I: Capítulo 2. Unidad II: Capítulo 3. Unidad III: Capítulo 11. Unidad IV: Apartado 9.3. Unidad V: Capítulo 10. Unidad VI: Capítulo 12. Unidad VII: Capítulo 4,5,6 y 7.

Preiss, B. R. (1999). Data Structures and Algorithms. John Wiley & Sons, Inc. Unidad I: Apartados 2.1.1-2.1.4 y Capítulo 3. Unidad II: Apartados 2.1.5. Unidad III: Capítulo 9 y 10. Unidad IV: Capítulo 11. Unidad V: Capítulo 8. Unidad VI: Capítulo 16. Unidad VII: Capítulo 5, 6 y 7.

Shaffer, Clifford A.: Data Structures and Algorithm Analysis in JAVA, Third Edition, Dover Publications, 2013. Unidad I: Apartados: 2.1-2.4 y 3.1-3.12. Unidad II: Apartados 2.4-2.6 y 3.1-3.12. Unidad III: Apartados 5.2-5.4, 6.1-6.3 y 13.1. Unidad IV: Apartado 5.5. Unidad V: Apartado 9.4. Unidad VI: Apartados 11.1-11.5. Unidad VII: Apartados 1.1 - 1.5.

Weiss, M. A. (2000). Estructuras de datos en Java: compatible con Java 2. Unidad I: Capítulo 5. Unidad II: Capítulo 7. Unidad III: Capítulo 17 y 18. Unidad IV: Capítulo 20. Unidad V: Capítulo 19. Unidad VI: Capítulo 14. Unidad VII: Capítulo 4, 15 y 16.

Weiss, M.A. Data Structures and Algorithm Analysis in JAVA,
4th Edition, Pearson/Addison Wesley, 2014. Unidad I: Apartados:
1.2 y Capitulo 2. Unidad II: Apartados 1.3 y Capitulo 2. Unidad III: Capitulo 4. Unidad IV:
Capitulo 6. Unidad V: Capitulo 5. Unidad VI: Capitulo 9. Unidad VII: Capitulo 3.

Bibliografía optativa

Granville Barnett, and Luca Del Tongo (2008). Data Structures and Algorithms: Annotated Reference with Examples First Edition. Unidad I: Capitulo 1. Unidad II: Apendice A y C. Unidad III: Capitulo 3, 7. Unidad IV: Capitulo 4 y Apartado 6.2. Unidad VII: Capitulo 2 y Apartado 6.1.

Hernández, Z.J. y otros: Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++, Thomson, 2005. Unidad III: Capitulo 5, 6 y 7. Unidad VI: Capitulo 10. Unidad VII: Capitulo 1.

Propuesta Pedagógico-Didáctica:

Las clases se desarrollarán en el Laboratorio de Informática. Se organizarán en modalidades teórico-prácticas con soporte de presentaciones digitales y prácticas en función de cada clase.

En las clases se presentan los contenidos teóricos y se van resolviendo en forma conjunta ejemplos que ayuden a comprender los nuevos conceptos introducidos.

La formación práctica está basada en la resolución de problemas tipo y de problemas abiertos de ingeniería, cuyas resoluciones se realizan principalmente en las computadoras, utilizando aplicaciones de uso en la industria que permitan un contacto directo con las tecnologías actuales.

En cuanto a los problemas abiertos de ingeniería, se realizarán trabajos relacionados con la implementación de una aplicación determinada en base a la programación orientada a objetos. Cada trabajo debe incluir el diagrama UML de las clases desarrolladas, un detalle de los problemas encontrados,

las formas de solucionarlos, las condiciones de ejecución, formato de los datos de entrada e ideas o sugerencias para realizar una versión mejorada del mismo. La realización de los trabajos implica la aplicación de los conocimientos adquiridos hasta entonces, lo cual constituye la base formativa para que el estudiante adquiera las habilidades que le permitan encarar proyectos y diseños de ingeniería.

Los estudiantes deben realizar entregas de al menos un trabajo integrador durante el desarrollo de la cursada el cual será revisado por los docentes y devuelto a los estudiantes para su corrección.

El material correspondiente a las clases teóricas, así como los documentos de la práctica se encontrarán disponibles a través de un sitio Web al cual los estudiantes tendrán acceso. Este mecanismo también provee la posibilidad de ser utilizado para realizar consultas simples.

En las actividades extra-áulicas se establecerán guías de actividades prácticas para que el estudiante pueda ejercitar, a fin de consolidar los conceptos aprendidos en clase.

Régimen de aprobación:

La asignatura tiene un régimen mediante el cual los alumnos que cumplan los requisitos podrán acceder a la aprobación de la materia por promoción directa. Alternativamente la materia se puede aprobar habiendo aprobado la cursada junto con el examen final teórico-práctico calificado con nota igual a 4 o superior.

La evaluación integradora de las instancias teórico-prácticas se realiza a través de un parcial teórico-práctico. Además, los estudiantes deberán desarrollar un trabajo final donde se integren los temas vistos en la materia. Las clases son obligatorias, exigiéndose una asistencia del 75 por ciento, ya que implican participación y debate que forman parte de la evaluación.

A continuación se detalla la forma mediante la cual se calcula la nota final del estudiante y por la cual se obtiene, materia promocionada/ cursada aprobada (rinde final)/ materia desaprobada, ajustándose con la reglamentación vigente de la de la Universidad que estipula promoción para notas finales de materia de 7 a 10, cursada aprobada con notas de 4 a 6.99 y cursada desaprobada con notas de 0 a 3.99 (Art. 45):

Al finalizar la cursada cada estudiante tendrá una calificación de la parte teórica-práctica (NTP) obtenida de la evaluación integradora teórico práctica, una calificación conceptual (NC) obtenida a partir de la participación en clase y una calificación del Trabajo Práctico Final (NTF). Tomando las calificaciones detalladas anteriormente, la nota final de la asignatura se obtiene de la siguiente manera:

$$\text{NOTA FINAL} = 0,4\text{NTP} + 0,2\text{NC} + 0,4\text{NTF}$$

