

Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

Descriptor de asignatura

Estructuras de datos y algoritmos

1. Identificación de la asignatura:

Nombre de la Asignatura: Estructura de datos y algoritmos	
Códigos: CDI-1100	Créditos: 6
Duración: Semestral	Ubicación en el plan de estudios: Semestre 3
Requisitos: CIT-1110 Programación avanzada	
Sesiones cátedras semanales: 2 cátedra, 1 laboratorio	
Sesiones de Ayudantía: 1	

2. Descripción de la asignatura:

El curso aborda los fundamentos teóricos y prácticos del análisis y diseño de algoritmos, destacando la importancia de la complejidad computacional para evaluar su eficiencia. Se enseña a aplicar notación asintótica para comparar algoritmos abstractos y sus implementaciones concretas, considerando limitaciones y costos computacionales. Los estudiantes realizarán estudios empíricos sobre tiempo de ejecución, uso de memoria y escalabilidad, utilizando métricas relevantes para evaluar desempeño en distintos contextos.

Se analizan las estructuras de datos básicas y avanzadas, seleccionando las más adecuadas según las características del problema. El curso enfatiza la implementación de soluciones algorítmicas eficientes, utilizando estructuras como listas, pilas, colas, árboles, tablas de hash y grafos. Además, se fomenta la documentación clara del desempeño de las soluciones desarrolladas, integrando los resultados en informes técnicos que respalden los procesos y análisis realizados.

3. Resultados de Aprendizaje:

1. Aplica el análisis teórico de la complejidad computacional a algoritmos abstractos e implementaciones concretas, utilizando notación asintótica, para compararlos y evaluarlos en el contexto del desarrollo de sistemas y aplicaciones específicas.
2. Realiza estudios empíricos del desempeño de implementaciones concretas, mediante la recolección y análisis de métricas como tiempo de ejecución, uso de memoria y escalabilidad, en contextos de pruebas controladas o escenarios reales de uso.
3. Analiza el desempeño de estructuras de datos básicas, seleccionando las más adecuadas para el modelamiento de soluciones e implementación de algoritmos eficientes.
4. Implementar soluciones a problemas complejos utilizando estrategias algorítmicas adecuadas y documentando su desempeño.
5. Implementa algoritmos de procesamiento de datos, organizados en cadenas, árboles, grafos y otras estructuras de común uso.
6. Identifica la complejidad computacional de problemas, analizando las limitaciones que ella impone.
7. Participa en equipos de trabajo, planificando, coordinando y ejecutando tareas con liderazgo y responsabilidad, comunicándose efectivamente y elaborando informes técnicos que reflejen procedimientos, resultados y análisis del trabajo realizado.ç

4. Unidades Temáticas:

Unidad 1: Fundamentos de algoritmos y análisis de complejidad

- Algorítmica de números enteros
- Análisis empírico de complejidad
- Análisis teórico de complejidad

Unidad 2: Algoritmos de ordenamiento y búsqueda

- Algoritmos de ordenamiento: Quick sort, Merge sort, Insertion sort, Heap sort.
- Búsqueda binaria

Unidad 3: Tipos de datos abstractos y estructuras fundamentales

- Tipos de datos abstractos y estructuras que los implementan
- Tipos básicos: filas, pilas, colas de prioridad, mapas
- Estructuras enlazadas

Unidad 4: Estructuras avanzadas para optimización y relaciones

- Árboles
- Tablas de hash
- Grafos

5. Descripción general del método de enseñanza:

Se contemplan dos clases semanales de cátedra, basadas en presentaciones electrónicas, con apoyo adicional de pizarrón y/o contenido audiovisual. El trabajo conceptual de las cátedras será complementado con una sesión semanal de ayudantía. Durante estas sesiones, la o el ayudante servirá de guía para el trabajo personal de las y los estudiantes enfocado en la resolución de actividades-tareas.

En cada sesión de ayudantía, además del tiempo dedicado a la resolución de dudas, se realizará una tarea. Cinco de ellas serán evaluadas con una nota, figurando en la planificación semana a semana como “controles”. En las demás, el ayudante va a entregar retroalimentación para guiar el aprendizaje de cada estudiante. Cada tarea corresponderá a un pequeño ejercicio de diseño e implementación de un algoritmo que resuelva un problema dado, a entregar al final de la ayudantía. Se esperará que cada entrega de tarea corresponda a un código que se compile y ejecute correctamente, entregando las soluciones correctas de las instancias de prueba dentro del límite de tiempo establecido.

Adicionalmente, se contempla un módulo semanal de laboratorio. Durante el semestre, se trabajará en cinco proyectos. Cada uno corresponderá a un desafío de mayor envergadura que los vistos en las tareas. Para cada laboratorio, además del código de la solución, se deberá entregar un informe con una descripción de la solución entregada, acorde a lo solicitado. El plazo de entrega de laboratorio va a variar entre una y dos semanas, según lo anunciado para cada caso.

Acorde al creditaje de la asignatura, además de las 5,5 horas cronológicas de actividades presenciales, el curso contempla 3,5 horas de trabajo autónomo. Este tiempo debe ser dedicado al estudio del material de clases, la lectura de textos y material adicional, la ejercitación en la resolución de pequeñas tareas como las de las ayudantías, y el desarrollo de los trabajos de laboratorio. Un buen uso del tiempo de trabajo autónomo es fundamental para el desarrollo de autoaprendizaje continuo.

Durante el semestre se realizarán dos pruebas solemnes que reunirán los tipos de ejercicios practicados en las cátedras, laboratorios y ayudantías. La materia de cada solemne corresponderá a la totalidad de los contenidos vistos en el curso hasta el momento respectivo. Las fechas de las solemnes serán anunciadas oportunamente dentro de cada una de las dos semanas de solemnes establecidas por la Escuela. Al final del semestre, durante el periodo de exámenes establecido por

la Escuela, se realizará un examen con ejercicios de los tipos practicados durante el semestre, sobre contenidos que contemplan la totalidad de la materia del curso.

6. Descripción general de la modalidad de evaluación:

Se contempla la realización de cinco controles, cinco laboratorios, dos pruebas solemnes y un examen. La nota final (NF) del curso se calculará a partir de la nota de presentación (NP) y la nota del examen (NE). Para el cálculo de la NP, participarán las notas de las pruebas solemnes (NS1 y NS2), el promedio simple de los controles (NC) y el promedio simple de los laboratorios (NL). De acuerdo con la regla general, para aprobar el curso debe tenerse que $NF \geq 4,0$.

De acuerdo con lo estipulado en la oferta académica, la asistencia a las sesiones de cátedra y laboratorio es obligatoria. Para controlar la tasa de asistencia, se realizará un registro tanto al inicio como al final de cada sesión, y el estudiante deberá estar presente en ambos momentos. Para cada uno de estos tipos de actividades, se requiere una tasa de asistencia mínima del 70%. En caso de no cumplir con este requisito, el estudiante será considerado como no apto para aprobar, con una nota final (NF) igual a la menor entre la nota de presentación (NP) y 3,9.

Para poder presentarse al examen, se debe tener $NP \geq 3,5$. De lo contrario, se reprueba con $NF = NP$.

La inasistencia a una prueba solemne, en caso de ser justificada ante la Secretaría Académica, implicará el reemplazo de su nota con la NE.

El profesor podrá eximir del examen final a estudiantes con $\min(NS; NC; NL) \geq 4,0$ y $NP \geq 5,0$.

7. Bibliografía Básica Obligatoria (Opcional):

1. R. Sedgewick, K. Wayne, Introduction to Programming in Java: An Interdisciplinary Approach. Addison-Wesley, 2017.
2. R. Sedgewick, K. Wayne, Algorithms. Addison-Wesley, 2014.
3. J. Kleinberg and E. Tardos, Algorithm Design. Pearson/Addison-Wesley, 2006.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms. MIT Press, 2009.