

Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

PROGRAMA DE ASIGNATURA
Estructuras de datos y algoritmos

I. Identificación

- Nombre : Estructuras de datos y algoritmos
- Código : CIT-2006
- Créditos : 6
- Duración : Semestral
- Ubicación en plan de estudio: Semestre 3
- Requisitos : CIT-1010 Programación avanzada
- Sesiones semanales : 2 cátedras, 1 ayudantía, 1 laboratorio

II. Descripción del curso

El rendimiento de un sistema de software está estrechamente ligado al diseño de los algoritmos que lo sustentan. El análisis de algoritmos, por su parte, ofrece una comprensión profunda de la naturaleza intrínseca de los problemas a resolver y de las posibles técnicas para abordarlos, independientemente del lenguaje de programación, el hardware u otros factores. Este curso entrega las herramientas fundamentales para desarrollar dichas capacidades.

III. Resultados de aprendizaje

Al finalizar el curso el el/la estudiante será capaz de:

1. Aplicar el análisis teórico de complejidad computacional a algoritmos abstractos e implementaciones concretas, usando la notación asintótica, para su comparación y análisis en el contexto de su uso para el desarrollo de sistemas y aplicaciones específicas.
2. Realizar estudios empíricos del desempeño de implementaciones concretas.
3. Analizar el desempeño de estructuras de datos básicas y seleccionar 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. Crear algoritmos que procesan datos organizados en cadenas, árboles, grafos y otras estructuras de uso común.
6. Identificar la complejidad computacional de problemas y analizar las limitaciones que ella impone.

IV. Unidades Temáticas

- Algoritmos iterativos.
- Recursión.
- Análisis empírico y teórico de complejidad.
- Tipos de datos básicos: listas, filas, pilas, colas de prioridad, conjuntos y mapas.
- Algoritmos de ordenamiento y búsqueda binaria.
- Listas enlazadas.

- Árboles.
- Tablas de hash.
- Grafos.

V. Metodología

Se contemplan dos sesiones 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 ejercicios.

En cada sesión de ayudantía, además del tiempo dedicado a la resolución de dudas, se trabajará en la resolución de ejercicios. En cinco sesiones de ayudantía, la resolución de ejercicios será evaluada con una nota, figurando en la planificación semana a semana como “control”. En las demás, el ayudante va a entregar retroalimentación para guiar el aprendizaje de cada estudiante. Cada ejercicio de ayudantía corresponderá a un pequeño problema computacional, con desafíos de diseño, implementación o análisis de un algoritmo que lo resuelva. Se esperará que cada entrega 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, el curso contempla una sesión semanal de laboratorio. A lo largo del semestre se desarrollarán cuatro proyectos, cada uno correspondiente a un desafío de mayor complejidad que los abordados en las ayudantías. Para cada proyecto, los estudiantes deberán entregar tanto el código de la solución como un informe escrito en el que se describa y fundamente la propuesta presentada, de acuerdo con las especificaciones indicadas. Posteriormente, algunos equipos —seleccionados al azar— deberán realizar una presentación oral de sus resultados. En estos casos, la evaluación de la presentación podrá ajustar la calificación del informe. Los equipos que no sean seleccionados mantendrán la nota originalmente obtenida en el informe. La calendarización de los desafíos se encuentra disponible en la planificación semana a semana del curso. Los proyectos se basarán en adaptaciones y extensiones de las implementaciones que acompañan el texto guía *Algorithms*, de Sedgewick y Wayne. De este modo, se busca acercar a los estudiantes a la práctica profesional, en la que una parte significativa del trabajo consiste no en desarrollar software desde cero, sino en comprender, modificar y extender código preexistente.

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.

VI. Evaluación

Se contempla la realización de cinco controles, cuatro laboratorios, dos pruebas solemnes y un examen.

La inasistencia a una prueba solemne, debidamente justificada ante la Secretaría Académica, implicará que la evaluación sea calificada con nota NE. Las inasistencias no justificadas serán sancionadas con nota 1,0. En el caso de las presentaciones de laboratorio, la inasistencia debidamente justificada ante la Secretaría Académica implicará que la nota del laboratorio corresponda a la nota obtenida en el informe escrito. Las inasistencias no justificadas serán calificadas con nota 1,0 en la evaluación respectiva. Asimismo, la no entrega de un laboratorio será calificada con nota 1,0, sin excepciones, considerando la amplitud del plazo asignado para su desarrollo. Finalmente, la inasistencia a un control, debidamente justificada ante la Secretaría Académica, implicará que dicha evaluación no sea considerada en el cálculo del promedio de controles. Las inasistencias no justificadas serán calificadas con nota 1,0.

La nota final (NF) del curso se calculará a partir de la nota de presentación (NP) y la nota de 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). La nota final se calculará de la siguiente forma:

$$NF = 0,7*NP + 0,3*NE$$

donde:

$$NP = 0,4*NS + 0,3*NC + 0,3*NL$$

$$NS = (NS1 + NS2)/2$$

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 $NF = NP$.

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

De acuerdo con la regla general, para aprobar el curso debe tenerse que $NF \geq 4,0$.

VII. Bibliografía Básica

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.

PAUTAS ETICAS BASICAS

El aula es un espacio donde los intercambios buscan generar un clima que potencie el aprendizaje, basado en el respeto y el buen trato. Las diferencias, tanto entre estudiantes, como entre estudiante y docentes, deben abordarse desde este marco de respeto.

La universidad cuenta con dos reglamentos importantes de conocer:

Reglamento de Convivencia

Normativa de Prevención y Sanción de Acciones de Discriminación, Violencia Sexual y/o de Género.

Puede consultar los reglamentos en el siguiente sitio web: <https://www.udp.cl/universidad/reglamentos-y-politicas/>

El plagio es el uso de las ideas o trabajo de otra persona sin el adecuado consentimiento. El plagio puede ser intencional o no. El plagio intencional es el claro intento de hacer pasar el trabajo o ideas ajenas como el suyo propio para su beneficio. El plagio no intencional puede ocurrir si Usted no conoce el mecanismo adecuado de referenciar la fuente de sus ideas e información. Si no está seguro de los métodos aceptados para referenciar, debería consultar con su profesor, tutor o personal de biblioteca.

"Cualquier conducta de un o una estudiante que vicie o tienda a viciar una actividad o evaluación académica, ya sea que se ejecute antes, durante o luego de su realización, dará origen a una o más de las siguientes sanciones, según la gravedad de la falta cometida: nota mínima (1,0) en la respectiva evaluación; reprobación del curso respectivo; suspensión por un semestre o año académico, o expulsión de la Universidad." (Artículo 50, Reglamento del Estudiante de Pregrado).

Elaborado por: Karol Suchan

Revisado por: Víctor Reyes

Fecha revisión: Marzo 2026

Fecha vigencia: Marzo 2026