

**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 cualquier sistema de software depende del diseño de los algoritmos elegidos. Por otra parte, el análisis de algoritmos proporciona una visión de la naturaleza intrínseca del problema, así como posibles técnicas de solución independientemente del lenguaje de programación, hardware, etc. El curso entrega las herramientas fundamentales para el desarrollo de estas tareas.

**III. Resultados de aprendizaje**

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

1. Aplicar el análisis 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 procesen datos organizados en cadenas, árboles, grafos y otras estructuras de común uso.
6. Identificar la complejidad computacional de problemas y analizar las limitaciones que ella impone.

#### **IV. Unidades Temáticas**

- Estructuras de datos básicas (filas, pilas, mapas y conjuntos).
- Tipos de datos abstractos.
- Uso de “interfaces” y “generics” en Java.
- Análisis de complejidad de algoritmos.
- Problemas y algoritmos de ordenamiento.
- Enfoque divide y vencerás, teorema maestro.
- Estructuras enlazadas.
- Algoritmos sobre árboles.
- Hashing.
- Algoritmos sobre grafos.
- Algoritmos codiciosos.
- Programación dinámica.

#### **V. Metodología**

Se contemplan dos clases semanales de cátedra, basadas en presentaciones electrónicas, con apoyo adicional de pizarrón y/o contenido audiovisual. Esto será complementado con lectura de textos y material adicional, para la realización de actividades-tareas, permitiendo desarrollar habilidades relacionadas con el autoaprendizaje continuo. 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.

Durante el semestre se realizarán actividades-tareas prácticas. Cada actividad-tarea corresponderá a un pequeño ejercicio de diseño e implementación de un algoritmo que resuelva un problema dado (en niveles de conocimiento, comprensión y aplicación de la taxonomía de Bloom), con carga estimada en 1 hora semanal – ocupando el horario de ayudantía. Se esperará que cada entrega de actividad-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.

Por otro lado, durante el semestre se realizará dos solemnnes con ejercicios conceptuales (en niveles de análisis, síntesis y evaluación de la taxonomía de Bloom). Cada solemnne se basará en preguntas que apunten a analizar la complejidad y evaluar la correctitud de algoritmos descritos en pseudocódigo o de implementaciones, o de diseñar un algoritmo nuevo y describirlo en pseudocódigo.

Adicionalmente, se realizará a lo largo del semestre un módulo semanal de laboratorio. El laboratorio será dedicado al desarrollo de un proyecto semestral integrador de todos los conocimientos adquiridos en la asignatura. El proyecto debe estar enfocado en satisfacer alguna necesidad de una organización en particular (con potencial de escalamiento a otras organizaciones del mismo tipo) o del público general. La complejidad de la problemática abarcada debe justificar el uso de las técnicas presentadas en el curso - el curso de Estructura de datos y algoritmos debe proporcionar un importante valor agregado, con respecto a otras fuentes de conocimiento y habilidades, para el resultado final del proyecto. Un proyecto exitoso deberá incorporar también resultados de autoaprendizaje continuo. Las reglas específicas del proyecto semestral del semestre en curso serán publicadas durante

la primera semana de clases en un manual de proyecto semestral.

## VI. Evaluación

Se contempla la realización de una docena de tareas-actividades, dos pruebas solemnes, un proyecto semestral 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). Asimismo, para el cálculo de la NP participarán las notas de las pruebas solemnes (NS1 y NS2), la nota de actividades-tareas (NAT), y la nota del proyecto semestral (NPS). El proyecto semestral se evaluará a través de tres entregas: Propuesta de Proyecto (PP), Avance de Proyecto (AP) y Entrega Final de Proyecto (EFP). La nota del curso se calculará de la siguiente forma:

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

donde:

$$NP = 0,2*NS + 0,4*NAT + 0,4*NPS$$

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

$$NPS = 0.3*PP + 0.3*AP + 0.4*EFP$$

De acuerdo con la regla general, para aprobar el curso debe tenerse que  $NF \geq 4,0$  y para presentarse a Examen -  $NP \geq 3,5$ . La inasistencia a una prueba solemne implicará reemplazo de su nota con la NE.

El profesor podrá eximir del examen final a estudiantes con  $\min(NS, NAT, NPS) \geq 4,0$  y  $NP \geq 5,0$ . La inasistencia justificada a una prueba solemne implicará reemplazo de su nota con la NE.

## VII. Bibliografía Básica

1. R. Sedgewick, K. Wayne, Algorithms. Addison-Wesley, 2014.
2. J. Kleinberg and E. Tardos, Algorithm design. Boston, Mass: Pearson/Addison-Wesley, 2006.
3. R. Sedgewick, K. Wayne, Introduction to Programming in Java: An Interdisciplinary Approach. Addison-Wesley, 2017.

### **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.*

*Puedes consultar los reglamentos aquí: <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 Ud. 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.*

Elaborado por: Karol Suchan  
Revisado por:  
Fecha revisión: Enero 2022  
Fecha vigencia: Marzo de 2022