**Facultad de Ingeniería y Ciencias  
Escuela de Informática y Telecomunicaciones**

**PROGRAMA DE ASIGNATURA**

Estructura de datos y algoritmos

**1.** **Identificación de la asignatura:**

| Nombre de la Asignatura: Estructura de datos y algoritmos | |
| --- | --- |
| Códigos: CIT-2306 | 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.

**4.** **Unidades Temáticas:**

Unidad I: Fundamentos de algoritmos y análisis de complejidad

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

### Unidad II: Algoritmos de ordenamiento y búsqueda

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

### Unidad III: 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 IV: Estructuras avanzadas para optimización y relaciones

* Árboles
* Tablas de hash
* Grafos

**Metodología**

El curso contempla dos clases semanales de cátedra apoyadas en presentaciones electrónicas, el uso de pizarrón y recursos audiovisuales, enfocadas en la exposición de los conceptos teóricos y su aplicación. Este trabajo se complementa con una sesión semanal de ayudantía, donde el o la ayudante guía a las y los estudiantes en la resolución de ejercicios prácticos.

Además, el curso incluye un módulo semanal de laboratorio, en el que se desarrollarán cinco proyectos de mayor complejidad, que requieren tanto la entrega del código funcional como de un informe descriptivo de la solución. Junto con las actividades presenciales, se espera que las y los estudiantes dediquen tiempo de trabajo autónomo al estudio del material, la práctica de ejercicios y el desarrollo de laboratorios. El proceso de aprendizaje se complementa con dos pruebas solemnes durante el semestre y un examen final, los cuales integran ejercicios representativos de las actividades realizadas en cátedras, ayudantías y laboratorios.

| Descripción de la metodología | Rol de estudiante en el aula | Actividades autónomas del estudiante |
| --- | --- | --- |
| **Clases expositivas**:  Presentación de contenidos teóricos mediante presentaciones electrónicas, pizarrón y recursos audiovisuales, orientadas a la comprensión conceptual y la introducción de técnicas. | - Participa activamente, atiende, toma apuntes y realiza preguntas. | - Revisa y estudia previamente el material recibido.  - Repasa y consolida los contenidos vistos en clase.  - Lee y estudia textos guía y complementarios. |
| **Trabajo colaborativo**:  Desarrollo de actividades de laboratorio en grupos, que requieren el diseño, implementación y documentación de soluciones más complejas. | - Colabora activamente en la coordinación del grupo.  - Desarrolla las tareas asignadas, tanto en código como en documentación.  - Incorpora la retroalimentación recibida desde el/la docente, a sus tareas.  - Elabora colaborativamente un informe técnico, acorde a lo solicitado. | - Repasa y resume los contenidos vistos en aula (cátedra y laboratorio).  - Investiga sobre herramientas para la resolución de las tareas asignadas.  - Desarrolla las tareas asignadas, tanto en código como en documentación.  - Coordina y revisa junto a su equipo las distintas entregas según calendario de actividades. |
| **Resolución de problemas**:  Desarrollo de tareas individuales en ayudantías, que plantean problemas específicos que deben resolverse mediante diseño e implementación de algoritmos funcionales. | - Analiza los problemas asignados al curso.  - Soluciona de forma autónoma los problemas asignados.  - Debate sobre la correctitud de su solución junto a su curso. | - Incorpora la retroalimentación recibida a sus prácticas de desarrollo y documentación. |

**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 ≥ 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 ≥ 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) ≥ 4,0 y NP ≥ 5,0.

| Evaluación | Descripción | Tipo de evaluación e instrumentos | Modalidad de retroalimentación | Ponderación |
| --- | --- | --- | --- | --- |
| Solemnes | Evaluaciones individuales de carácter formal que integran y aplican los contenidos teóricos y prácticos del curso. Se realizan en fechas definidas durante el semestre. | Prueba escrita con ejercicios de diseño, análisis, implementación y uso de estructuras y algoritmos. | Retroalimentación grupal mediante revisión de soluciones y discusión de errores frecuentes en clases. | El promedio corresponderá al 40% de la nota de presentación. |
| Controles | Evaluaciones prácticas en algunas sesiones de ayudantía. | Tarea teórica y/o práctica individual de programación con entrega inmediata. | Retroalimentación inmediata del ayudante en la sesión y/o retroalimentación escrita. | El promedio corresponderá al 30% de la nota de presentación. |
| Laboratorios | Actividades prácticas grupales que requieren la aplicación de los contenidos en problemas de mayor complejidad, con entrega de código funcional y un informe técnico. | Desarrollo de código y entrega de informe descriptivo. | Retroalimentación escrita sobre el informe y el código entregado, con revisión de errores, sugerencias y evaluación del diseño y la implementación. | El promedio corresponderá al 30% de la nota de presentación. |
| Examen final | Evaluación individual que integra la totalidad de los contenidos teóricos y prácticos del curso. Busca medir el dominio global de las competencias desarrolladas durante el semestre. | Prueba escrita con ejercicios de análisis, diseño e implementación de estructuras de datos y algoritmos. | Retroalimentación individual a través de la revisión de la prueba. | Corresponderá al 30% de la nota final. |

**7.** **Planificación de Sesiones:**

La planificación de las sesiones dependerá del calendario académico vigente al momento de dictar

la asignatura. En cualquier caso, cada docente que la imparta deberá informar a la Escuela (antes del comienzo del semestre) la planificación de sus actividades, evaluaciones, y mecanismos de diversificación de las mismas.

Esta información será informada al estudiantado durante la primera semana de clases.

**8.** **Bibliografía**

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 éticas básicas**

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 estudiantes y docentes, deben abordarse desde este marco de respeto.

La universidad cuenta con dos reglamentos importantes de conocer:

* Reglamento de Convivencia Estudiantil
* 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/>

Así también, 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.

El plagio comprobado es una actitud que puede resultar en severas sanciones disciplinarias y/o en la exclusión de la Universidad (Artículo 44, Reglamento del Estudiante de Pregrado).

Elaborado por: Karol Suchan

Fecha revisión: Mayo de 2025

Fecha vigencia: Marzo de 2026