



Programa de Asignatura

I. IDENTIFICACIÓN				
Carrera o programa: Ingeniería Civil en Computación e Informática				
Unidad responsable: Escuela de Ingeniería				
Nombre de la asignatura: Diseño y Análisis de Algoritmos				
Código: ECIN-00608				
Semestre en la malla¹: 7				
Créditos SCT - Chile: 5				
Ciclo de Formación		Básico		Profesional X
Tipo de Asignatura		Obligatoria	X	Electiva
Clasificación de área de conocimiento²				
Área: Ingeniería y Tecnología			Sub área: Ingeniería Informática	
Requisitos:				
Pre-requisitos:			Requisito para:	
• ECIN-00506 Fundamentos de la Computación				

II. ORGANIZACIÓN SEMESTRAL							
Horas Dedicación Semanal (Cronológicas)		Docencia Directa	4.5	Trabajo Autónomo	3	Total	7.5
Detalle Horas	Cátedra	Ayudantía	Laboratorio	Taller	Terreno	Exp. Clínica	Supervisión
Directas	3		1.5				

¹Este campo

²Clasificación del curso de acuerdo a la OCDE



III. APORTE AL PERFIL DE EGRESO

La asignatura contribuye al dominio 1 del perfil de egreso, “Conocimiento científico y disciplinario”. Además, contribuye al dominio 2 “Habilidades y Actitudes Personales y Profesionales”. También contribuye al dominio 4 “Habilidades para la Práctica de la Ingeniería”. Al finalizar la asignatura los y las estudiantes serán capaces de aplicar principios y técnicas generales del diseño de algoritmos y analizar la complejidad de problemas y sus soluciones algorítmicas, dando énfasis en algoritmos de ordenamiento, búsqueda, selección y grafos, entre otros.

IV. HABILIDADES PERFIL DE EGRESO (RELACIÓN)

1.2 Aplicar conocimientos de ciencias de la ingeniería a la solución de problemas complejos de ingeniería.

2.1 Identificación, formulación, modelación y resolución de problemas complejos de ingeniería considerando las interacciones y la dinámica de las variables.

2.2 Aplicación del método científico para diseñar, conducir y realizar investigación en ingeniería.

4.4 Diseñar las soluciones TIC para la industria intensiva en procesamiento de la información. Estas soluciones consideran las arquitecturas TI junto a sus modelos de servicios y modelos operativos; los sistemas de software; y las plataformas de cómputo y comunicaciones junto a sus servicios asociados.



V. RESULTADOS DE APRENDIZAJE

1. Aplicar métodos de resolución de ecuaciones de recurrencia.
2. Medir empíricamente el rendimiento de un algoritmo.
3. Identificar órdenes de magnitud asociados a los algoritmos estudiados.
4. Diseñar soluciones algorítmicas y distinguir entre ellas según sus costos.
5. Desarrollar la capacidad para analizar soluciones y determinar el mejor algoritmo posible para resolver un problema dado.
6. Identificar si un problema es NP-completo.
7. Analizar las relaciones causa efecto de los procesos en estudio.
8. Preparar conclusiones apoyadas por datos, necesidades y valores.
9. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.

VI. ÁREAS TEMÁTICAS

1. Introducción (2,5 semanas)
 - 1.1 Algoritmos y problemas computacionales
 - 1.2 Análisis de ordenamiento por inserción
 - 1.3 Notaciones asintóticas
 - 1.4 Algoritmo de Strassen para la multiplicación de matrices
 - 1.5 Métodos de resolución de recurrencias
 - 1.6 Clases de complejidad (e.g., constante, lineal, logarítmica, cuadrática, exponencial)
 - 1.7 Medición empírica del rendimiento de un algoritmo
2. Análisis de técnicas para el diseño de algoritmos (3 semanas)
 - 2.1 Enfoque de fuerza bruta
 - 2.2 Dividir para reinar
 - 2.3 Programación dinámica
 - 2.4 Métodos Greedy
 - 2.5 Backtracking
3. Análisis de algoritmos de ordenamiento y selección (3 semanas)
 - 3.1 Definición de orden y árbol de decisión
 - 3.2 Análisis de Heapsort



- 3.3 Análisis de Quicksort
- 3.4 Análisis de Bucketsort
- 3.5 Tipos de problema de selección
- 3.6 Análisis de variante de selección basada en Quicksort
- 4. Análisis de algoritmos de Hashing (2 semanas)
 - 4.1 Definición de Hashing y sus alternativas
 - 4.2 Análisis de Hashing con encadenamiento
 - 4.3 Análisis de Hashing con direccionamiento abierto
 - 4.4 Determinación de la mejor estrategia de aplicación de Hashing
- 5. Análisis de algoritmos de búsqueda en texto (2 semanas)
 - 5.1 Definición del problema de búsqueda en texto
 - 5.2 Análisis del algoritmo de fuerza bruta
 - 5.3 Análisis del algoritmo de Knuth-Morris-Pratt
 - 5.4 Análisis de algoritmo de Rabin-Karp
- 6. Análisis de algoritmos en grafos (2 semanas)
 - 6.1 Representaciones de grafos. Isomorfismo entre grafos.
 - 6.2 Análisis del algoritmo de Kruskal
 - 6.3 Análisis del algoritmo de búsqueda en profundidad para grafos no dirigidos
 - 6.4 Análisis de algoritmo de determinar distancias mínimas en grafos dirigidos: Dijkstra, Floyd
- 7. Problemas NP-Complejos (1,5 semanas)
 - 7.1 Definición de problemas NP-Complejos
 - 7.2 Determinación de complejidad de problemas NP-Complejos



VII. ORIENTACIONES METODOLÓGICAS

1. La metodología a desarrollar en esta asignatura debe favorecer la interacción entre las y los estudiantes a través de trabajos prácticos colaborativos que permitan la solución a problemas específicos contextualizados a la asignatura.
 - Se sugiere el uso de clases expositivas y participativas con método combinado, es decir, clases expositivas con alternancia de trabajos en grupo de corta duración para responder preguntas.
 - Se sugiere la utilización de la metodología activa de análisis de casos para desarrollar experiencias que permitan incorporar los elementos teórico-prácticos asociados a los resultados de aprendizaje de la asignatura.
2. Las experiencias de cátedra/laboratorio/taller deben ser realizadas por medio de la utilización de software moderno aplicable a la asignatura.
3. Se recomienda que las y los estudiantes realicen presentaciones periódicas sobre el trabajo realizado que incluya: contextualización, desarrollo y conclusiones.
4. Actividades prácticas recomendadas: cápsulas teóricas, reuniones de trabajo, taller de trabajo en equipo y liderazgo, presentaciones e informes escritos de avance en español, revisión del estado del arte asociado al problema, lluvia de ideas, análisis de alternativas y descripción detallada de la solución.



VII. ORIENTACIONES Y CRITERIOS PARA LA EVALUACIÓN

1. Se recomienda la aplicación de una evaluación diagnóstica al inicio de la asignatura.
2. La asignatura podría contemplar dos instancias de evaluación de los resultados de aprendizaje: cátedra y taller/laboratorio.
 - En el caso de existir, ambas debieran ser aprobadas por separado: el porcentaje de cada una de ellas deberá ser de 60% para cátedra y 40% para taller/laboratorio.
 - En el caso que la asignatura tenga actividades de taller/laboratorio, éstas deben ser realizadas en grupos de estudiantes y se recomienda la elaboración por parte de los estudiantes de un informe sobre la actividad desarrollada.
3. Se evaluará el conocimiento conceptual y procedimental mediante el desarrollo de al menos dos pruebas sumativas de carácter presencial.
 - Se recomienda además la aplicación de una evaluación mediante la entrega de un trabajo desarrollado en las horas indirectas asociadas a la asignatura.
 - Se recomienda que las y los estudiantes realicen una o más presentaciones de los trabajos realizados, la evaluación de la misma debe ser por medio de la aplicación de una rúbrica.
4. Se recomienda realizar evaluaciones de carácter formativo. Esto permite al docente introducir correcciones, añadir alternativas y reforzar los aspectos para ayudar al estudiantado en el logro de sus habilidades.
5. La asistencia y condiciones de aprobación de la asignatura debe ser acorde a la aplicación del Reglamento de Docencia de Pregrado.



IX. RECURSOS BIBLIOGRÁFICOS

Bibliografía Mínima

- Levitin, A. (2011). Introduction to the Design and Analysis of Algorithms (3rd ed.). Addison Wesley.
- Cormen, Leiserson, Rivest, Stein. (2009) Introduction to Algorithms (3rd ed.). The MIT Press.

Bibliografía Complementaria

- Sedgewick, R., Flajolet P. (2013). An Introduction to the Analysis of Algorithms (2nd ed.). Addison Wesley.
- Knuth, D.E. (2011). The Art of Computer Programming. Addison Wesley.
- Soza, H. (1999). Apuntes de Análisis de Algoritmos. Universidad Católica del Norte.