PROGRAMA DE CURSO
DISEÑO Y ANÁLISIS DE ALGORITMOS

A. Antecedentes generales del curso:

Departamento	Ciencias de la Computación							
Nombre del curso	Diseño y Análisis de Algoritmos		Código	CC4102		Créditos	6	
Nombre del curso en inglés	*Design and Analysis of Algorithms*							
Horas semanales	Docencia	3	Auxiliares	1,5		Trabajo personal	5,5	
Carácter del curso	Obligatorio	X		Electivo				
Requisitos	(MA3402: Estadística/MA3403: Probabilidades y Estadística), CC3001: Algoritmos y estructuras de datos, CC3102: Teoría de la computación							

B. Propósito del curso:

El curso tiene como propósito que los y las estudiantes diseñen, analicen e implementen algoritmos y estructuras de datos, considerando técnicas avanzadas (cotas inferiores, análisis amortizado, dominios discretos, y algoritmos en línea y competitividad) y el uso de modelos avanzados de computación (memoria secundaria, algoritmos probabilísticos y aleatorizados, aproximados y paralelos) para la resolución eficiente de problemas computacionales.

El curso tributa a las siguientes competencias específicas (CE) y genéricas (CG):

CE1: Analizar problemas computacionales, construir modelos, expresándolos en representaciones y lenguajes formales adecuados. CE2: Analizar, diseñar y/o adoptar, algoritmos y estructuras de datos que cumplan con las garantías requeridas de correctitud y eficiencia. CE3: Gestionar bases de datos utilizando modelos, lenguaje de consulta asociados, técnicas eficientes de acceso a datos y aplicación de políticas de seguridad, con la finalidad de obtener información relevante.	CG1: Comunicación académica y profesional Comunicar en español de forma estratégica, clara y eficaz, tanto en modalidad oral como escrita, puntos de vista, propuestas de proyectos y resultados de investigación fundamentados, en situaciones de comunicación compleja, en ambientes sociales, académicos y profesionales.

C. Resultados de aprendizaje:

Competencias específicas	Resultados de aprendizaje
CE1	RA1: Demuestra matemáticamente las cotas inferiores en problemas para establecer la optimalidad de un algoritmo, utilizando técnicas como estrategia del adversario, teoría de la información y reducciones.
CE2	RA2: Diseña y analiza algoritmos y estructuras de datos utilizando técnicas avanzadas como análisis amortizado, uso de dominios discretos, y algoritmos en línea y competitividad.
CE2, CE3	RA3: Diseña y analiza algoritmos y estructuras de datos en modelos avanzados de computación, incluyendo memoria secundaria, algoritmos aleatorizados y probabilísticos, aproximados y paralelos.
CE1, CE2, CE3	RA4: Determina y analiza un conjunto de ejemplos emblemáticos de soluciones de algoritmos y estructuras de datos de mediana complejidad aplicables a problemas que se le presenten.
CE2, CE3	RA5: Implementa algoritmos o estructuras de datos para resolver un problema dado, comparando las soluciones basadas en técnicas elementales con las obtenidas mediante técnicas de carácter avanzado.
Competencias genéricas	**Resultados de aprendizaje**
CG1	RA6: Redacta argumentos claros y concisos para justificar técnicamente los resultados de la implementación de algoritmos y estructuras de datos con su correspondiente análisis e interpretación, considerando conceptos y aspectos del diseño y análisis de algoritmos.

D. Unidades temáticas:

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
1	RA1	Conceptos básicos y complejidad	3 semanas
Contenidos		**Indicador de logro**	
1.1. Técnicas para la demostración de cotas inferiores: adversario, teoría de la información, reducción. 1.2. Ejemplos de casos de estudio: cota inferior para mínimo y máximo de un arreglo, cota inferior para búsqueda en un arreglo con distintas probabilidades de acceso, cota inferior para la cápsula convexa, 3SUM reducido a puntos colineales, árboles de búsqueda óptimos.		La/el estudiante: 1. Relaciona el concepto de cota inferior de problemas con la optimalidad de los algoritmos. 2. Aplica técnicas como la estrategia del adversario, teoría de la información, y reducciones, para demostrar cotas inferiores de problemas. 3. Distingue y analiza ejemplos representativos de cotas inferiores.	
Bibliografía de la unidad		[1] Cap 1-4. [3] Cap 2, 10.	

	[4] Cap 1, 4. [5] Cap 2. [6] Cap 1-2, 10. [7] Cap 6. [9] Cap 8-9. [10] Cap 1,2.

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
2	RA2, RA3. RA5, RA6	Algoritmos y Estructuras de Datos para Memoria Secundaria	3 semanas

Contenidos	Indicador de logro
2.1. Modelo de computación en memoria secundaria. Accesos secuenciales y aleatorios 2.2. Ordenamiento en memoria secundaria: Mergesort. Cota inferior. 2.3. Colas de prioridad en memoria secundaria. Cotas inferiores. 2.4. Diccionarios en memoria secundaria: árboles B, *hashing* lineal y extendible.	La/el estudiante: 1. Compara los modelos de costo de memoria principal y secundaria, en base a características físicas de dichas memorias. 2. Diseña y analiza soluciones algorítmicas eficientes en memoria secundaria. 3. Implementa soluciones eficientes para resolver problemas de ordenamiento, colas de prioridad y diccionarios en memoria secundaria. 4. Redacta textos breves donde describe, analiza e interpreta los resultados de la implementación de algoritmos y estructuras de datos, en el contexto de las tareas a desarrollar.
Bibliografía de la unidad	[1] Cap 18. [5] Cap 4.7, 7.11. [7] Cap 13, 18. [9] Cap 11. [10] Cap 5.7, 6.3.

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
3	RA2, RA4, RA5, RA6	Técnicas avanzadas de diseño y análisis de algoritmos	4 semanas

Contenidos	Indicador de logro
3.1. Análisis amortizado de algoritmos y estructuras de datos: análisis completo, contabilidad de costos, función potencial. 3.2. Uso de dominios discretos y finitos en el diseño de algoritmos. 3.3. Algoritmos en línea. Competitividad. 3.4. Ejemplos de casos de estudio: estructuras para union-find, colas binomiales, splay trees, radix sort, árboles de van Emde Boas, árboles Patricia y de sufijos, paginamiento, move-to-front, búsqueda no acotada, k-server problem.	La/el estudiante: 1. Diseña y analiza algoritmos y estructuras de datos eficientes en términos del costo amortizado. 2. Diseña y analiza algoritmos y estructuras de datos especializadas en dominios discretos y finitos. 3. Diseña algoritmos en línea, analizando su competitividad. 4. Identifica y analiza algunos ejemplos representativos de algoritmos y estructuras de datos relacionados con los conceptos de análisis amortizado, dominios discretos y algoritmos en línea. 5. Implementa soluciones eficientes de estructuras amortizadas, técnicas para dominios discretos, y algoritmos en línea. 6. Describe y analiza por escrito los resultados de la implementación considerando precisión y concisión en la elaboración de sus ideas y argumentos.
Bibliografía de la unidad	[1] Cap 8, 17, 19, 21. [2] Cap 4. [3] Cap 6. [5] Cap 4-6, 8, 11. [7] Cap 17. [8] Cap 9. [9] Cap 5. [10] Cap 3.3.

Número	RA al que tributa	Nombre de la unidad	Duración en semanas
4	RA3, RA4, RA5, RA6	Algoritmos no convencionales	5 semanas

Contenidos	Indicador de logro
4.1. Algoritmos aleatorizados y probabilísticos. Casos en que se justifican. Relación con la NP-completitud. 4.2. Algoritmos tipo Monte Carlo y Las Vegas. 4.3. Aleatorización de la entrada. Independencia de la distribución de la entrada. Estructuras de datos aleatorizadas. 4.4. Solución de problemas NP-completos: búsqueda exhaustiva. Concepto de algoritmos aproximados. 4.5. Nociones de aproximabilidad. Problemas que son o no aproximables. 4.6. Algoritmos paralelos y distribuidos. Modelo PRAM. Medidas de complejidad. Técnicas de diseño. 4.7. Ejemplos de casos de estudio: primalidad, árboles binarios de búsqueda aleatorizados, *quicksort*, *hashing* universal y perfecto, aproximaciones para recubrimiento de vértices, vendedor viajero, mochila. Ordenamiento paralelo, *parallel-prefix*, *list* ranking.	La/el estudiante: 1. Analiza las ventajas y desventajas de utilizar algoritmos y estructuras de datos aleatorizados y probabilísticos, y los tipos de aleatorización. 2. Diseña y analiza algoritmos y estructuras de datos aleatorizados y probabilísticos. 3. Determina cuándo es necesario utilizar algoritmos aproximados y los tipos de aproximación. 4. Diseña y analiza algoritmos aproximados para resolver problemas NP-completos. 5. Distingue modelos de computación paralela y sus modelos de costo. 6. Diseña y analiza algoritmos paralelos en el modelo PRAM. 7. Identifica y analiza algunos casos emblemáticos de algoritmos y estructuras de datos aleatorizados y probabilísticos, aproximados y paralelos. 8. Implementa soluciones aleatorizadas, probabilísticas y aproximadas para la solución de problemas computacionalmente complejos. 9. Elabora por escrito textos breves donde describe, analiza e interpreta los resultados de la implementación de algoritmos y estructuras de datos.
Bibliografía de la unidad	[1] Cap 5, 31.8, 32.2, 35. [3] Cap 6, 11, 12. [4] Cap 6. [5] Cap 10. [6] Cap 8. [7] Cap 35, 40, 44. [8] Cap 1, 7, 8, 12, 14. [10] Cap 4, 12.

E. Estrategias de enseñanza -aprendizaje:

El curso considera las siguientes posibles estrategias de enseñanza:

- **Clases expositivas** donde se presentan los principales conceptos a trabajar en la sesión, considerando la participación activa de los y las estudiantes y se les estimula, en determinados momentos, a proponer la solución a un problema o desafío planteado.
- **Resolución de problemas**, a partir de problemas que se le presentan los y las estudiantes proponen una solución de diseño cuyos resultados entrega en los plazos asignados.
- ***Flipped Classroom:*** los y las estudiantes revisan y estudian los apuntes o materiales antes de la sesión para luego resolver ejercicios en grupos pequeños durante la clase y/o de manera individual después del horario de las clases.

El trabajo del y la estudiante es acompañado por el equipo docente, resolviendo dudas cuando se requiera.

F. Estrategias de evaluación:

Al inicio de cada semestre el académico o académica informará los y las estudiantes sobre los tipos de evaluación, la cantidad definida, así como las ponderaciones correspondientes.

Para esta propuesta se consideran las siguientes instancias de evaluación:

- **Controles** (entre dos y tres): resolución de ejercicios a partir de problemas que se le plantean.
- **Tareas** (entre dos y tres): en el contexto de la ejecución de las tareas los y las estudiantes deben describir, analizar e interpretar por escrito los resultados de la implementación de algoritmos y estructuras de datos, utilizando argumentos basados en aspectos disciplinares.
- **Examen** (1); evalúa de manera integradora los contenidos y aprendizajes de todo el curso.

*No obstante, opcionalmente se podrán aplicar otras estrategias de aprendizaje directamente relacionadas con los procedimientos de evaluación, tal como **Mastery Grading** donde los y las estudiantes reciben retroalimentación sobre la solución a un problema que se le presenta (evaluación formativa), para luego realizar los ajustes correspondientes y entregar la nueva propuesta para su evaluación final.*

G. Recursos bibliográficos:

Bibliografía obligatoria:

Navarro. G. **Apuntes del curso Diseño y análisis de algoritmos.** Disponible en Material Ucursos.

Bibliografía complementaria:

[1] Cormen, T., Leiserson, C., Rivest, R., Stein, C. (2009). *Introduction to Algorithms*. MIT Press: 3rd edition.
[2] Aho, A., Hopcroft, J., Ullman, J., (1974). *The Design and Analysis of Computer Algorithms*. Addison-Wesley.
[3] Manber, U. (1989). *Introduction to Algorithms.* Addison-Wesley.
[4] Rawlins, G. (1992). *Compared to what?* Computer Science Press.
[5] Weiss, M. (1995). *Data Structures and Algorithm Analysis,* 2nd edition. Benjamin Cummings.
[6] Brassard, G, Bratley, P. Algorithmics. (1988). Theory and Practice. Prentice-Hall.
[7] Sedgewick, R. (19929. *Algorithms in C++.* Addison-Wesley.
[8] Motwani, R., Raghavan, P. (2000). *Randomized Algorithms.* Cambridge.
[9] Aho, A., Hopcroft, J., Ullman, J. (1983). *Data Structures and Algorithms.* Addison-Wesley.
[10] Mehlhorn, K. and Sanders. P. (2008). *Algorithms and Data Structures*. Springer.
[11] Skiena, S. (2008). *The Algorithm Design Manual. Springer.*
[12] Mitzenmacher, M. y Upfal, E. (2005). *Probability and Computing.* Cambridge.
[13] Ziviani, N. (2007). **Diseño de Algoritmos.** Thomson.
[14] Levitin, A. (2007). The Design & Analysis of Algorithms, Pearson: 2nd ed.
[15] Lee, R., Tseng, S., Chang, R., Tsai, T. (2005). Introduction to the Design and Analysis of Algorithms. McGraw-Hill.
[16] Kleinberg, J. y Tardos, E. (2006). *Algorithm Design.* Addison-Wesley.
[17] Jájá, J. (1992). *An Introduction to Parallel Algorithms.* Addison-Wesley.
[18] Borodin, A. y El-Yaniv, R. (2005). *Online Computation and Competitive Analysis*. Cambridge.

H. Datos generales sobre elaboración y vigencia del programa de curso:

Vigencia desde:	Otoño, 2021 Ajuste 2025
Elaborado por:	Gonzalo Navarro
Validado por:	Revisión académico par: Jeremy Barbay. Validación CTD de Computación
Revisado por:	Área de Gestión Curricular