

Nombre de la asignatura: **ANÁLISIS Y DISEÑO DE ALGORITMOS**

Línea de Trabajo: **Ingeniería de Software/Inteligencia Artificial**

Tiempo de dedicación del estudiante a las actividades de:

DOC	TIS	TPS	Horas Totales	Créditos
48	40	80	168	6

DOC: Docencia; **TIS:** Trabajo independiente significativo; **TPS:** Trabajo profesional supervisado

1. Historial de la asignatura

Fecha revisión/actualización	Participantes	Observaciones, cambios o justificación
19/Junio/2023	Dra. Adriana Mexicano Santoyo M.S.C. Francisco Argüelles Granados	Versión inicial
23/Enero/2025	Dr. Mario Alberto Gómez Rodríguez Dr. Jesús Carlos Carmona Frausto	Actualización incluyendo uso de IA

2. Pre-requisitos y Correquisitos.

Se requieren conocimientos de:

- Fundamentos de Matemáticas Discretas.
- Fundamentos de Programación.
- Fundamentos de Probabilidad.

3. Objetivo de la asignatura

Analizar, diseñar e implementar algoritmos eficientes y correctos para la solución computacional de problemas de diferentes clases de complejidad.

4. Aportación al perfil del graduado

El alumno será capaz de:

- Analizar y diseñar algoritmos aplicando las distintas técnicas vistas en clase, dependiendo del tipo de problema a abordar.
- Aplicar el Análisis y Diseño de Algoritmos en el campo Inteligencia Artificial.
- Determinar la complejidad de un problema mediante su análisis.
- Proponer soluciones (aproximadas o exactas) a problemas computacionales.

5. Contenido temático

Unidad	Temas	Subtemas
1	Introducción: problemas representativos	1.1 Conjuntos 1.2 Conteo y probabilidad 1.3 Matrices 1.4 Aplicaciones de Inteligencia Artificial en Problemas Combinatorios
2	Bases del análisis de algoritmos	2.1 Tratabilidad Computacional 2.2 Orden Asintótico de Crecimiento 2.3 Implementación del Algoritmo de Emparejamiento Estable Usando Listas y Arreglos 2.4 Resumen de los Tiempos de Ejecución Comunes 2.5 Una Estructura de Datos Más Compleja: Colas de Prioridad

		2.6 Inteligencia Artificial y Complejidad Computacional
3	Grafos	3.1. Definición y aplicaciones de los grafos. 3.2. Conectividad y navegación de los grafos. 3.3. Implementando navegación de grafos usando colas y pilas. 3.4. Algoritmo para determinar si un grafo es bipartita. 3.5. Conectividad en los grafos dirigidos. 3.6. Grafos dirigidos sin ciclos y ordenamiento topológico. 3.7 Grafos en Aprendizaje Automático: Redes Neuronales y Grafos de Conocimiento
4	Algoritmos voraces	4.1. Calendarización de intervalos. 4.2. Caches óptimos. 4.3. Caminos más cortos en un grafo. 4.4. El problema del árbol más pequeño. 4.5. El algoritmo de Kruskal. 4.6. Clustering. 4.7. Codificación de Huffman y Compresión de datos. 4.8 Algoritmos Voraces en Inteligencia Artificial: Optimización y Aprendizaje
5	Divide y vencerás	5.1. El algoritmo de MergeSort. 5.2. Relaciones de recurrencia. 5.3. Conteo de acciones de inversión. 5.4. Encontrando el par de puntos más cercano. 5.5. Multiplicación de enteros. 5.6 El enfoque Divide y Vencerás en algoritmos de inteligencia artificial
6	Programación dinámica	6.1 Representación de los principios de programación dinámica. 6.2 Principio de optimalidad. 6.3 Programación dinámica en horizonte limitado 6.4 Programación dinámica en horizonte ilimitado. 6.5 Programación Dinámica en Inteligencia Artificial
7	NP e intratabilidad computacional	7.1 Introducción 7.2 Algoritmos y Complejidad 7.3 Problemas NP Completos 7.4 Problemas Intratables 7.5 Problemas de Decisión 7.6 Algoritmos No Determinísticos 7.7 Inteligencia Artificial y Problemas NP: Reducción de Problemas y Técnicas de Aproximación
8	Algoritmos de aproximación y búsqueda local	8.1 Algoritmos Greedy y Límites sobre el Óptimo: Un Problema de Balanceo de Carga 8.2 El Problema de Selección de Centro 8.3 Cobertura de Conjuntos: Una Heurística Greedy General 8.4 El Método de Precios: Cobertura de Vértices 8.5 Programación Lineal y Redondeo: Una Aplicación a la Cobertura de Vértices 8.6 Balanceo de Carga Revisitado: Una Aplicación Avanzada de Programación Lineal 8.7 Aproximaciones Arbitrariamente Buenas: El Problema de la Mochila 8.8 El Panorama de un Problema de Optimización

		8.9 El Algoritmo de Metrópolis y Recocido Simulado 8.10 Una Aplicación de Búsqueda Local a Redes Neuronales de Hopfield 8.11 Aproximación de Máximo Corte a través de Búsqueda Local 8.12 Clasificación a través de Búsqueda Local 8.13 Algoritmos de Aproximación e Inteligencia Artificial
9	Algoritmos aleatorios	9.1 Una Primera Aplicación: Resolución de Contención 9.2 Encontrar el Corte Mínimo Global 9.3 Variables Aleatorias y Sus Expectativas 9.4 Un Algoritmo de Aproximación Aleatorizado para MAX 3-SAT 9.5 Divide y Venceras Aleatorizado: Búsqueda de la Mediana y Quicksort 9.6 Hashing: Una Implementación Aleatorizada de Diccionarios 9.7 Encontrar el Par de Puntos Más Cercanos: Un Enfoque Aleatorizado 9.8 Caché Aleatorizada 9.9 Algoritmos Aleatorios e Inteligencia Artificial

6. Metodología de desarrollo del curso

El docente sirve de guía para que el estudiante adquiera el conocimiento necesario para realizar el análisis y diseño de diferentes tipos de algoritmos mediante el análisis de la complejidad computacional de los mismos.

7. Sugerencias de evaluación

Se recomienda la siguiente ponderación:

Examen de medio término	20%
Tareas	40%
Proyecto Final	40%

8. Bibliografía y Software de apoyo

- Introduction to Algorithms (Fourth Edition). The MIT Press, 2022. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
- Introduction to Algorithms (Third Edition). The MIT Press, 2009. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein.
- Introducción al diseño y análisis de algoritmos - Un enfoque estratégico. R. C. T. Lee, S. S. Tseng, R. C. Chang, and Y. T. Tsai. McGraw-Hill Interamericana, 2007.
- Introduction to the Design and Analysis of Algorithms - A Strategic Approach. R. C. T. Lee, S. S. Tseng, R. C. Chang, and Y. T. Tsai. McGraw-Hill Education (Asia), 2005.

9. Prácticas propuestas

Unidad	Práctica
1	Complejidad temporal de un algoritmo: Análisis del mejor caso, caso promedio y el peor caso del Algoritmo de Ordenamiento por Inserción Directa.
2	Implementar (en Java) el Algoritmo de Ordenamiento por inserción directa.
3	Diseño de Algoritmos: Análisis de la complejidad temporal del Método Divide y Vencerás Merge-Sort.
4	Implementar (en Java) los Algoritmos Merge y Merge-Sort.
5	Revisión de avance del proyecto final
6	Proyecto final

10. Catedrático (s) responsable (s)

A handwritten signature in black ink that reads "Mario Gómez". The signature is written in a cursive style with a horizontal line underlining the name.

Dr. Mario Alberto Gómez Rodríguez

A handwritten signature in black ink that reads "Jesús F.". The signature is written in a cursive style with a horizontal line underlining the name.

Dr. Jesús Carlos Carmona Frausto