



1. Datos Generales de la asignatura

Nombre de la asignatura:	Algoritmia y estructuras de datos.
Clave de la asignatura:	IAD-2401
SATCA¹:	2-3-5
Carrera:	Ingeniería en Inteligencia Artificial.

2. Presentación

Caracterización de la asignatura
<p>La importancia de la materia radica en que aporta al perfil del egresado el conocimiento, la correcta selección y aplicación de las estructuras de datos, además del desarrollo de algoritmos aplicados en la solución de problemas, así como el análisis de la eficiencia de algoritmos y las estructuras de datos usadas en el desarrollo de soluciones eficientes.</p> <p>La algoritmia y estructura de datos engloba diferentes recursos y procedimientos que permiten el análisis, diseño e implementación de estructuras de datos y control, que facilitan el procesamiento, estructuración y búsqueda de datos.</p> <p>La materia de algoritmos y estructura de datos es de relevancia en la formación de los estudiantes, ya que desarrolla las habilidades para el análisis, diseño y estructuración de algoritmos aplicados. De forma general, la materia consiste en analizar problemas, diseñar algoritmos generales, aplicar estructuras de datos para ordenamiento y búsquedas avanzadas, lo que fortalecerá al estudiante en materias de programación como paradigmas de programación.</p> <p>Para su óptimo desarrollo, es importante haber cursado las materias de fundamentos de programación y programación orientada a objetos, siendo fundamental en el análisis, diseño y desarrollo de soluciones informáticas con inteligencia artificial.</p>
Intención didáctica
<p>Esta asignatura está proyectada a abordarse de forma teórico-práctica, ya que el docente explica y propone problemáticas, y el estudiante deberá resolverlas, aplicando analítica y estructura de datos. El contenido se encuentra organizado en cuatro temas.</p> <p>El primer tema de análisis y diseño de algoritmos, el docente, mediante actividades teórico-prácticas proporciona a los estudiantes las habilidades necesarias para analizar, diseñar y evaluar algoritmos eficientes.</p> <p>En el tema estructuras de datos y algoritmos de búsqueda, con el planteamiento de problemas y desarrollo de prácticas que involucren el uso de estructuras y algoritmos, se busca familiarizar a los estudiantes con las estructuras de datos y los algoritmos de búsqueda.</p>

¹ Sistema de Asignación y Transferencia de Créditos Académicos



En el tema de algoritmos de ordenamiento y estructuras de datos avanzadas, busca profundizar en las estructuras de datos y los algoritmos de ordenamiento.

Finalmente, en el tema técnicas avanzadas de diseño de algoritmos, se pretende la aplicación de algoritmos y estructuras de datos en soluciones con inteligencia artificial y su aplicación en problemas diversos.

El docente en su rol de facilitador del aprendizaje proporciona orientación y apoyo a los estudiantes en el proceso de enseñanza-aprendizaje. Además de un ambiente óptimo de trabajo, el docente retroalimenta los contenidos y mejora el desempeño de los estudiantes, guiándolos en el desarrollo de habilidades y competencias clave.

Estos temas fortalecerán la capacidad de pensamiento analítico en el estudiante, tendrá creatividad e innovación, con un nivel de abstracción y razonamiento lógico para la gestión y organización.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Tecnológico Nacional de México del 4 al 6 de marzo de 2024.	Representantes de los Institutos Tecnológicos de: Celaya, Chihuahua, Iztapalapa III, La Paz, Matehuala, Mérida, Minatitlán, Querétaro, Saltillo, Tijuana. Instituto Tecnológico Superior de Teziutlán. Tecnológico de Estudios Superiores de Ixtapaluca.	Propuesta sintética de la carrera de Ingeniería en Inteligencia Artificial.
Tecnológico Nacional de México del 22 al 26 de abril de 2024	Representantes de los Institutos Tecnológicos de: Celaya, Chihuahua, Iztapalapa III, La Paz, Matehuala, Mérida, Minatitlán, Querétaro, Saltillo, Tijuana. Instituto Tecnológico Superior de Teziutlán, Tecnológico de Estudios Superiores de Ixtapaluca.	Diseño y/o desarrollo curricular de la carrera de Ingeniería en Inteligencia Artificial.
Tecnológico Nacional de México del 27 al 31 de mayo de 2024	Representantes de los Institutos Tecnológicos de: Celaya, La Paz, Matehuala, Mérida, Minatitlán.	Consolidación curricular de la carrera de Ingeniería en Inteligencia Artificial.



4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura

Adquiere conocimientos sobre algoritmos y estructuras de datos y puede aplicarlos eficazmente para resolver problemas complejos. Además, tiene la habilidad de analizar críticamente uso y aplicaciones de algoritmos y estructuras, adaptándose a diferentes contextos y requerimientos de desarrollo de software, con las siguientes competencias:

- Diseña algoritmos y estructuras de datos para disponer y procesar información en diversas plataformas tecnológicas.
- Elige los paradigmas y lenguajes de programación más adecuados para el uso, implementación y optimización de estructuras y algoritmos.
- Aplica conocimientos sobre procedimientos algorítmicos básicos y utiliza los tipos y estructuras de datos apropiados.
- Optimiza procesos mediante el uso de algoritmos.
- Facilita tareas de búsqueda de información con la implementación de estructuras y algoritmos.

5. Competencias previas

El estudiante de la asignatura de algoritmia y estructura de datos debe tener una base sólida de conocimientos previos que incluyen:

- Comprende los fundamentos de programación.
- Aplica los conceptos de la programación orientada a objetos en el planteamiento de problemas.
- Maneja conceptos básicos de algoritmos y estructuras de datos para asimilar los distintos paradigmas de programación.
- Conoce los conceptos básicos de la programación orientada a objetos para relacionarlos con otros paradigmas.
- Utiliza habilidades de resolución de problemas de forma óptima y estructurada.

6. Temario

No.	Temas	Subtemas
1	Análisis y diseño de algoritmos	1.1. Definición de algoritmo y estructuras de datos. 1.2. Conceptos de complejidad algorítmica. 1.3. Análisis de la eficiencia de los algoritmos: tiempo y espacio. 1.4. Notación Big O y su importancia en la eficiencia de algoritmos.



2	Estructuras de datos y algoritmos de búsqueda.	2.1. Estructuras de datos. 2.1.1. Arreglo. 2.1.2. Listas. 2.1.3. Matrices multidimensionales. 2.1.4. Pilas. 2.1.5. Colas. 2.1.6. Punteros. 2.2. Algoritmos de búsqueda. 2.2.1. Búsqueda lineal. 2.2.2. Búsqueda binaria.
3	Algoritmos de ordenamiento y estructuras de datos avanzadas.	3.1. Algoritmos de Ordenamiento. 3.1.1. Inserción. 3.1.2. Selección. 3.1.3. QuickSort. 3.2. Árboles binarios de búsqueda (BST) y árboles 3.3. Almacenamiento libre y colas de prioridad.
4	Técnicas avanzadas de diseño de algoritmos.	4.1. Conceptos de grafos y sus representaciones. 4.2. Algoritmos de búsqueda en grafos. 4.2.1. DFS. 4.2.2. BFS. 4.3. Algoritmos de búsqueda informada. 4.3.1. A(A-Star)*. 4.3.2. Greedy Best-First. 4.4. Técnicas avanzadas de diseño de algoritmos. 4.4.1. Programación dinámica. 4.4.2. Algoritmos voraces.

7. Actividades de aprendizaje de los temas

1. Análisis y diseño de algoritmos	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Diseña, analiza y evalúa la eficiencia de algoritmos para resolver problemas computacionales diversos. Incluyendo la comprensión de diferentes técnicas de diseño de algoritmos, como la habilidad para implementar soluciones en código funcional.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Capacidad de pensamiento analítico. ● Creatividad e Innovación. ● Habilidades de abstracción. 	<ul style="list-style-type: none"> ● Investigar sobre algoritmos y estructuras de datos, discutir en grupos pequeños las definiciones y ejemplos proporcionados. ● Resolver problemas usando algoritmos básicos y estructuras de datos presentando sus aplicaciones en el mundo real. ● Implementar algoritmos simples evaluando el tiempo de ejecución en diferentes conjuntos de datos para entender cómo afecta el tamaño de la entrada. ● Comparar y analizar el uso de memoria de diferentes algoritmos y estructuras de datos para resolver un mismo problema.



<ul style="list-style-type: none"> ● Razonamiento lógico. ● Gestión del tiempo y organización. 	<ul style="list-style-type: none"> ● Investigar y discutir casos de uso de algoritmos donde la notación Big O es crucial para evaluar su eficiencia.
2. Estructuras de datos y algoritmos de búsqueda	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Implementa, comprende y utiliza de forma adecuada las estructuras de datos y algoritmos de búsqueda para resolver problemas computacionales. Considerando la selección apropiada de estructuras de datos y algoritmos según las necesidades del problema y la evaluación de su eficiencia y efectividad.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Pensamiento analítico. ● Resolución de problemas. ● Creatividad e innovación. ● Habilidades de abstracción. ● Razonamiento lógico. ● Habilidades matemáticas. ● Habilidades de programación. ● Habilidades de comunicación. ● Trabajo en equipo. ● Gestión del tiempo y organización. 	<ul style="list-style-type: none"> ● Implementar arreglos unidimensionales, listas y matrices multidimensionales en un lenguaje de programación de su elección. Luego, realizar operaciones básicas como inserción, eliminación y búsqueda en estas estructuras. ● Presentar problemas del mundo real que puedan ser resueltos utilizando arreglos, listas o matrices multidimensionales. ● Presentar una serie de operaciones de inserción, eliminación y acceso, solicitar que realicen las mismas operaciones en una pila y una cola. Comparar los resultados y discutir las diferencias en el comportamiento de ambas estructuras. ● Presentar conjuntos de datos ordenados y desordenados y solicitar que implementen ambos algoritmos para buscar un elemento específico. Luego, discutir las diferencias en el rendimiento de cada algoritmo en diferentes escenarios.
3. Algoritmos de ordenamiento y estructuras de datos avanzadas	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Diseña, implementa y utiliza eficazmente los algoritmos de ordenamiento y estructuras de datos en la resolución de problemas computacionales complejos. Incluyendo la capacidad para seleccionar y aplicar las estructuras y algoritmos adecuados para un problema dado.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Pensamiento analítico. ● Resolución de problemas. ● Creatividad e innovación. ● Habilidades de abstracción. ● Razonamiento lógico. ● Habilidades matemáticas. ● Habilidades de programación. 	<ul style="list-style-type: none"> ● Presentar problemas del mundo real que puedan ser resueltos utilizando algoritmos de ordenamiento, como la clasificación de datos de ventas o la organización de archivos. Aplicar los algoritmos de inserción, selección y QuickSort para resolver estos problemas. ● Implementar árboles binarios de búsqueda (BST) y árboles AVL en un lenguaje de programación de su elección. ● Analizar la complejidad temporal de las operaciones en árboles binarios de búsqueda y árboles AVL, como la inserción, eliminación y búsqueda. ● Implementar montículos (heaps) y colas de prioridad en un lenguaje de programación de su elección. Proporcionar conjuntos de datos de prueba y realizar operaciones de inserción, eliminación y acceso al elemento de mayor prioridad.



<ul style="list-style-type: none"> ● Habilidades de comunicación. ● Trabajo en equipo. ● Gestión del tiempo y organización. 	
4. Técnicas avanzadas de diseño de algoritmos	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Aplica y analiza diversas técnicas de diseño de algoritmos para resolver problemas computacionales complejos. Comprendiendo las técnicas de programación dinámica, algoritmos voraces, algoritmos de vuelta atrás (backtracking), y divide y vencerás, así como la capacidad para seleccionar y aplicar la técnica adecuada para un problema específico.</p> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> ● Pensamiento analítico. ● Resolución de problemas. ● Creatividad e innovación. ● Habilidades de abstracción. ● Razonamiento lógico. ● Habilidades matemáticas. ● Habilidades de programación. ● Habilidades de comunicación. ● Trabajo en equipo. ● Gestión del tiempo y organización. 	<ul style="list-style-type: none"> ● Crear diferentes tipos de grafos (dirigidos, no dirigidos, ponderados, etc.) para realizar operaciones de agregar y eliminar nodos y aristas, así como buscar vecinos de un nodo específico. ● Comparar el rendimiento y la eficiencia de DFS y BFS en diferentes tipos de grafos. Analizar factores como el tiempo de ejecución, el espacio en memoria y la complejidad del algoritmo en diferentes escenarios. ● Presentar problemas utilizando algoritmos de búsqueda informada, como la planificación de rutas o la resolución de laberintos. Aplicar los algoritmos A* y Greedy Best-First para resolver estos problemas y comparar sus resultados. ● Proporcionar problemas que puedan resolverse utilizando técnicas de programación dinámica y algoritmos voraces. Implementar soluciones utilizando estas técnicas y comparar los resultados.

8. Práctica(s)

<p>Análisis y diseño de algoritmos.</p> <ul style="list-style-type: none"> ● Elaborar una exploración de datos en distintas fuentes de información y plantear un algoritmo de exploración. ● Hacer el análisis de los algoritmos recientes de procesamiento de información Crea un plan de mejora continua para un servicio existente en una empresa. Incluye acciones específicas para mejorar la calidad y eficiencia del servicio. <p>Estructuras de datos y algoritmos de búsqueda.</p> <ul style="list-style-type: none"> ● Aplicar las estructuras de datos para almacenar datos de análisis. ● Implementar los algoritmos de búsqueda recientes en datos no ordenados. ● Desarrollar el software que permite buscar datos dentro de las estructuras.
--



Algoritmos de ordenamiento y estructuras de datos avanzadas.

- Trabajar con una base de datos desordenada y ordenarla con un algoritmo de búsqueda seleccionado.
- Elaborar un programa con cualquier lenguaje de programación para ejecutar los algoritmos más comunes cómo: Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort, Counting Sort, Radix Sort y Bucket Sort.
- Crear un programa que represente y simula un árbol binario de búsqueda.

Técnicas avanzadas de diseño de algoritmos.

- Programar un algoritmo de búsqueda para un sistema de archivo distribuido.
- Programar un algoritmo para búsqueda en grafos.
- Crear un programa para optimización de rutas.
- Crear un programa para búsqueda de aproximación de objetivos.

9. Proyecto de asignatura

El objetivo del proyecto es demostrar el desarrollo y alcance de las competencias de la asignatura de algoritmia y estructura de datos.

Proyecto:

Sistema de gestión de recursos materiales de una institución pública.

Fundamentación:

El objetivo de este proyecto es desarrollar un sistema de gestión de recursos, que use una base de datos plana que permita a los usuarios buscar, prestar y devolver recursos de manera eficiente. Los departamentos tienen varios recursos, y se necesita un sistema automatizado para llevar un registro de los préstamos y devoluciones.

Planeación:

Análisis de Requisitos:

- Identificar los actores del sistema (usuarios, prestadores).
- Definir los casos de uso (buscar recursos, prestar recursos, devolver recursos).
- Diseñar la base de datos para almacenar información sobre los recursos, usuarios y préstamos.

Diseño del Sistema:

- Crear un diagrama de clases para representar las entidades del sistema.
- Diseñar la interfaz de usuario (pantallas de búsqueda, préstamo y devolución).
- Planificar la arquitectura del sistema (backend en el lenguaje de programación, frontend).

Implementación:

- Desarrollar las funcionalidades del sistema:
- Búsqueda de recursos por tipo, uso.
- Registro de préstamos y devoluciones.
- Validaciones (por ejemplo, no prestar un recurso ya prestado).
- Crear una base de datos plana para almacenar la información.



Ejecución:

Desarrollo del Sistema:

- Implementar las funcionalidades según el diseño.
- Realizar pruebas unitarias para asegurar la calidad del código.

Integración:

- Conectar el frontend con el backend.
- Probar el sistema completo.

Evaluación:

Pruebas de Usuario:

- Invitar a los usuarios a probar el sistema.
- Evaluar la usabilidad, eficiencia y satisfacción.

Optimización:

- Identificar posibles mejoras (por ejemplo, agregar una función de reserva de recursos).
- Optimizar el rendimiento del sistema (tiempo de búsqueda, escalabilidad).

Resultados Esperados:

- Un sistema de gestión de recursos funcional y eficiente.
- Usuarios satisfechos con la facilidad de uso y la rapidez del sistema.
- Reducción del tiempo dedicado a la gestión manual de préstamos y devoluciones de recursos.

10. Evaluación por competencias

Las técnicas, herramientas y/o instrumentos sugeridos que permiten obtener el producto del desarrollo las actividades de aprendizaje

- **Evaluación Continua.**
 - Tareas periódicas durante el semestre que evalúen diferentes aspectos de las competencias específicas y genéricas.
 - Ejercicios prácticos en clase y tareas para completar en casa que aborden diferentes paradigmas de programación.
- **Cuestionarios.**
 - Cuestionarios que incluyan preguntas teóricas y prácticas sobre las temáticas abordadas en el curso.
 - Evaluación continua para medir el rendimiento y comprensión de los temas de algoritmia y estructura de datos.
- **Proyectos y Trabajos Prácticos.**
 - Desarrollo de proyectos prácticos que integren el uso y aplicación de estructuras de datos, algoritmos de ordenamiento y técnicas de búsqueda múltiples paradigmas de programación, evaluando la capacidad de los estudiantes para aplicar los conceptos aprendidos en situaciones reales.
 - Entrega de trabajos escritos que analicen y comparen las estructuras y algoritmos en función de criterios específicos, como eficiencia, legibilidad y mantenibilidad.
 - Evaluación de la colaboración y el trabajo en equipo en proyectos prácticos y discusiones grupales.



11. Fuentes de Información

1. Bhargava, A. (2016). Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People. Manning Publications.
2. Cestero, V., & Caballero, E. (2023). Inteligencia Artificial: Fundamentos matemáticos, algorítmicos y metodológicos (Spanish Edition).
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.
4. Ebel, F. (2019). Algoritmia - Técnicas fundamentales de programación: Ejemplos en Python (numerosos ejercicios corregidos). BTS, DUT informática recursos informáticos. ediciones ENI.
5. Goodrich, M. T., & Tamassia, R. (2014). Data Structures and Algorithms in Java. John Wiley & Sons.
6. Hemant Jain. (2021). Estructuras de datos y algoritmos simplificados.
7. Lafore R. (2021). Estructura de datos y algoritmos en java.
8. Nadal, M. (2019). Estructuras de datos y algoritmos: Guía ilustrada para programadores [Pasta blanda]. Anaya multimedia. ISBN-13: 978-8441545199
9. Skiena, S. S. (2008). The Algorithm Design Manual. Springer Science+Business Media, LLC.
10. Sedgewick, R., & Wayne, K. (2011). Algorithms. Addison-Wesley.