



1. Datos Generales de la asignatura

Nombre de la asignatura:	Estructura de datos
Clave de la asignatura:	CDD-2407
SATCA¹:	2-3-5
Carrera:	Ingeniería en Ciencia de Datos

2. Presentación

Caracterización de la asignatura

La asignatura proporciona al perfil del egresado una comprensión sólida de las diferentes estructuras de datos y algoritmos utilizados en ciencia de datos y en el desarrollo de software en general. A lo largo del curso, los estudiantes adquirirán habilidades en el diseño, implementación y análisis de estructuras de datos y algoritmos eficientes.

La asignatura busca que el alumno comprenda los conceptos fundamentales de las estructuras de datos y su importancia en la resolución eficiente de problemas. Además, el alumno debe aprender a seleccionar y aplicar estructuras de datos apropiadas, e implementarlas en lenguajes de programación comunes.

Para cursar esta asignatura se requiere tener habilidades básicas de programación e interpretación de algoritmos y tener el dominio del paradigma orientado a objetos. Además, debe de conocer y manejar los conceptos generales de la lógica matemática, relaciones y la teoría de grafos, por esta razón se encuentra ubicada para ser cursada después de fundamentos de programación y de programación orientada a objetos y matemáticas discretas, a su vez, esta asignatura es el pilar fundamental en el análisis, diseño y desarrollo de aplicaciones de software de bajo y alto nivel.

Intención didáctica

La intención didáctica de esta asignatura es proporcionar a los estudiantes una comprensión sólida y habilidades prácticas en el análisis, diseño e implementación de estructuras de datos orientados a la aplicación en ciencia de datos.

Esta asignatura, propone el estudio detallado de las estructuras de datos, para permitir durante su desarrollo, aportar habilidades lógicas y creativas en para solucionar problemas de ciencias de datos.

Se propone el uso de un lenguaje como C, C++, que permita la programación detallada y tratar a detalle los algoritmos, sin distraerse de su esencia, cuando estos sean tratados por su profesor, posteriormente, los alumnos pueden materializarse en algún lenguaje de alto nivel orientado a objetos.

¹ Sistema de Asignación y Transferencia de Créditos Académicos



El contenido temático de esta asignatura es base importante en diversas asignaturas de ciencias de datos, para su programación y/o para el desarrollo del pensamiento lógico computacional.

La asignatura está organizada en cinco unidades donde se conceptualiza y materializa un tipo de dato abstracto como una estructura de datos lineales y no lineales aplicadas en diferentes etapas y procesos en la ciencia de datos.

En la **primera unidad**, se establecen los conceptos de las estructuras de datos y su aplicación y la forma de optimizar la organización de datos que se emplean en diferentes etapas y procesos de ingeniería en ciencia de datos. Para estudiar los tipos de datos abstractos (TDA) se describen con definiciones en casos de la vida real y casos aplicables en ciencia de datos.

En la **segunda unidad**, se define, conceptualiza, diseña e implementa la recursividad. También, se analiza a detalle su funcionalidad aplicando la recursión a diferentes problemas de diferentes áreas de conocimiento, destacando modelos matemáticos aplicados a la ciencia de datos. Se concluye con un análisis profundo de sus ventajas y desventajas en su aplicación en el desarrollo de problemas algorítmicos de ciencia de datos.

En la **tercera unidad**, se pretende que los alumnos comprendan las estructuras de datos lineales, su funcionamiento y aplicación como una base sólida para el manejo de datos en ciencia de datos. En esta unidad se inicia con el estudio de listas dinámicas (con el uso de apuntadores o referencia), se propone su estudio completamente detallado, lo que posteriormente permitirá la optimizar del estudio de las demás estructuras lineales propuestas.

En la **cuarta unidad**, corresponde al estudio de la estructura de datos no lineal: árbol. Su análisis detallado, está motivado por ser una estructura de datos ampliamente utilizada en ciencia de datos. Se propone su estudio partiendo del árbol binario de búsqueda, su conceptualización, diseño, implementación iterativa y profundizando en su manejo recursivo. Posteriormente, realiza el estudio completo del árbol AVL como una estructura de datos con una amplia capacidad de manejo eficiente de grandes volúmenes de información. Para el estudio del árbol AVL, se propone que en su implementación almacene datos compuestos, tipo registros, y con ellos permita su análisis en diferentes órdenes.

En la **quinta unidad**, se dedica al estudio de la estructura de datos no lineal: grafo. El estudio del grafo es motivado por su destacada importancia en diferentes aplicaciones y etapas de la ciencia de datos. Se propone iniciar su estudio desde la conceptualización como una estructura matemática, su análisis funcional, su diseño algorítmico y su implementación con algún lenguaje de programación que permita su implementación a un nivel medio (lenguajes como C y C++) son una opción adecuada. Se aplicará con el análisis algorítmico en recorrido de profundidad y amplitud, y concluye con el estudio e implementación de algún algoritmo clasificado como voraz (greedy)



3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Alvarado del 21 al 23 agosto de 2023.	Representante del Instituto Tecnológico Superior de Alvarado.	Propuesta inicial.
Tecnológico Nacional de México 30 octubre 2023	Representante del Instituto Tecnológico de: Querétaro y del Instituto Tecnológico Superior de Alvarado.	Presentación de la propuesta de la carrera de Ingeniería en Ciencia de Datos.
Instituto Tecnológico de Querétaro Campus Norte del 19 al 22 de marzo 2024.	Representantes de los Institutos Tecnológicos de: Morelia, Puebla, Querétaro, Tehuacán. Instituto Tecnológico Superior de Alvarado. CENIDET. Representante de Ciencias Básica de los Institutos de: Celaya, Morelia y CIIDET.	Diseño y/o desarrollo curricular de la carrera de Ingeniería en Ciencia de Datos.
Tecnológico Nacional de México del 22 al 24 de abril del 2024	Representante del Instituto Tecnológico de Querétaro e Instituto Tecnológico Superior de Alvarado.	Contraste y ajuste de las asignaturas de Ingeniería en Ciencia de Datos con respecto a las de Ing. en Inteligencia Artificial, Ing. en Desarrollo WEB e Ing. en Ciberseguridad
Tecnológico Nacional de México del 27 al 31 de mayo del 2024.	Representantes de los Institutos Tecnológicos de: Morelia, Querétaro. Instituto Tecnológico Superior de Alvarado. CENIDET.	Consolidación curricular de la carrera de Ingeniería Ciencia de Datos

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
<ul style="list-style-type: none"> Comprender los conceptos fundamentales de las estructuras de datos y su importancia en la resolución eficiente de problemas. Aprender a seleccionar y aplicar estructuras de datos apropiadas para diferentes situaciones y problemas. Desarrollar habilidades en la implementación de estructuras de datos en lenguajes de programación comunes. Analizar la complejidad temporal y espacial de algoritmos y estructuras de datos. Aplicar estructuras de datos en contextos específicos de ciencia de datos y análisis de datos.



5. Competencias previas

- **Programación básica:** debes tener un conocimiento sólido de programación en al menos un lenguaje de programación, como Python, Java, C++, etc. Debes entender conceptos como variables, estructuras de control (condicionales y bucles), funciones o métodos, y estructuras de datos simples como listas o arrays.
- **Álgebra y matemáticas básicas:** un entendimiento básico de conceptos matemáticos como álgebra y aritmética es importante para comprender la complejidad de algoritmos y análisis de eficiencia.
- **Pensamiento lógico y resolución de problemas:** la capacidad de pensar de manera lógica y resolver problemas es esencial para diseñar algoritmos y estructuras de datos eficientes.
- **Conocimiento de variables y tipos de datos:** debes comprender los conceptos de variables y tipos de datos en un lenguaje de programación, incluyendo enteros, flotantes, cadenas, booleanos, etc.
- **Uso de IDE (entorno de desarrollo integrado):** familiaridad con el uso de un IDE para escribir, depurar y ejecutar código es útil. Esto incluye conocer las características básicas del entorno de desarrollo, como la edición de código, la depuración y la ejecución de programas.
- **Conocimiento de estructuras de control:** debes estar cómodo con las estructuras de control, como condicionales (if, else) y bucles (for, while), ya que se utilizan en la programación de algoritmos.
- **Gestión de memoria básica:** entender cómo funciona la gestión de memoria en tu lenguaje de programación elegido, especialmente cuando se trabaja con estructuras de datos más avanzadas como punteros en C/C++.
- **Herramientas de control de versiones:** tener conocimiento básico de herramientas de control de versiones como Git puede ser beneficioso, ya que a menudo se trabaja en proyectos de programación colaborativa.
- **Lógica de conjuntos y algebra booleana:** estos conceptos son útiles cuando trabajas con estructuras de datos como conjuntos o mapas.
- **Comprensión de variables y referencias:** si estás usando un lenguaje de programación que permite la manipulación directa de la memoria, debes entender cómo funcionan las variables y las referencias.

6. Temario

No.	Temas	Subtemas
1	Introducción a estructuras de datos.	1.1. Definición y conceptos básicos de estructuras de datos. 1.2. Tipos de datos abstractos (TDA).
2	Recursividad.	2.1. Definición. 2.2. Procedimientos recursivos. 2.3. Ejemplos de casos recursivos.



3	Estructuras lineales: listas, pilas y colas.	<p>3.1. Listas.</p> <p>3.1.1. Listas enlazadas simples.</p> <p>3.1.2. Listas circulares.</p> <p>3.1.3. Arrays y matrices.</p> <p>3.1.4. Operaciones básicas en listas y arrays.</p> <p>3.1.5. Comparación de rendimiento entre listas y arrays.</p> <p>3.2. Pilas y Colas</p> <p>3.2.1. Concepto de pila (stack) y cola (queue).</p> <p>3.2.2. Implementación de pilas y colas.</p> <p>3.2.3. Aplicaciones de pilas y colas en ciencia de datos.</p>
4	Árboles	<p>4.1. Árboles binarios.</p> <p>4.2. Árboles balanceados</p>
5	Grafos	<p>5.1. Grafos dirigidos y no dirigidos.</p> <p>5.2. Recorridos en árboles y grafos</p> <p>5.3. Aplicaciones de árboles y grafos en ciencia de datos.</p>

7. Actividades de aprendizaje de los temas

1. Introducción a estructuras de datos	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Conoce y comprende las diferentes estructuras de datos, su clasificación y forma de manipularlas para buscar la manera más eficiente de resolver problemas.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Comprensión conceptual. ● Identificación de problemas. ● Capacidad para trabajar en equipo. ● Habilidad para buscar y analizar información proveniente de fuentes diversas. ● Resolución de problemas básicos. 	<ul style="list-style-type: none"> ● Investigar las diferentes estructuras de datos y su clasificación para elaborar un cuadro sinóptico. ● Hacer un resumen sobre el concepto de Tipos de datos abstractos y su implementación en la programación orientada a objetos.



2. Recursividad	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Aplica la recursividad en la solución de problemas valorando su pertinencia en el uso eficaz de los recursos.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Habilidad para buscar y analizar información proveniente de fuentes diversas. ● Capacidad de análisis y síntesis ● Habilidad en el manejo de equipo de cómputo. ● Capacidad para trabajar en equipo. ● Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> ● Redactar una definición propia del concepto de recursividad después de consultar en diferentes fuentes bibliográficas y discutirla en trinas. ● Enumerar las ventajas y desventajas del uso de la recursividad en una plenaria. ● Trasladar un catálogo de problemas iterativos a recursivos, donde distinga el segmento recursivo y la condición de salida, elaborar un reporte de práctica de ejercicios. ● Desarrollar programas en los cuales se aplique la recursividad y entregar informes.
3. Estructuras lineales: listas, pilas y colas	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Comprende y aplica estructuras de datos lineales para solución de problemas.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> ● Habilidad para buscar y analizar información proveniente de fuentes diversas. ● Capacidad de análisis y síntesis ● Habilidad en el manejo de equipo de cómputo. ● Capacidad para trabajar en equipo. ● Capacidad de investigación. ● Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> ● Consultar las características y aplicación de las estructuras lineales en diversas fuentes bibliográficas y comentarlas en plenaria. ● Programar las operaciones básicas de pilas y colas de manera estática (usando arreglos) entregando los códigos correspondientes. ● Utilizar las clases predefinidas para el manejo de pilas, colas y listas enlazadas (dinámicas) y describir en un texto la diferencia de hacerlo con arreglos. ● Utilizar las estructuras lineales en la elaboración de códigos para la resolución de problemas elaborando un reporte.
4. Árboles	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Comprende y aplica estructuras no lineales para la solución de problemas.</p> <p><i>Genéricas:</i></p> <ul style="list-style-type: none"> ● Habilidad para buscar y analizar información proveniente de fuentes diversas. ● La comprensión y manipulación de ideas y pensamientos. 	<ul style="list-style-type: none"> ● Elaborar un cuadro sinóptico o esquema con la clasificación de los árboles y sus aplicaciones. ● Implementar las operaciones básicas de inserción, eliminación y búsqueda en un árbol binario. ● Implementar las operaciones básicas de inserción, eliminación y búsqueda en un árbol balanceado.



<ul style="list-style-type: none"> • Metodologías para solución de problemas, organización del tiempo y para el aprendizaje. • Habilidad en el manejo de equipo de cómputo • Capacidad para trabajar en equipo. • Capacidad de aplicar los conocimientos en la práctica. 	
5. Grafos	
Competencias	Actividades de aprendizaje
<p><i>Específica(s):</i> Comprende y aplica estructuras no lineales para la solución de problemas.</p> <p><i>Genérica(s):</i></p> <ul style="list-style-type: none"> • Habilidad para buscar y analizar información proveniente de fuentes diversas. • La comprensión y manipulación de ideas y pensamientos. • Metodologías para solución de problemas, organización del tiempo y para el aprendizaje. • Habilidad en el manejo de equipo de cómputo. • Capacidad para trabajar en equipo. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Hacer un mapa conceptual o mental del uso y aplicaciones de los grafos. • Desarrollar aplicaciones con grafos y elaborar un reporte de prácticas.

8. Práctica(s)

<ul style="list-style-type: none"> • Desarrollar una aplicación que resuelva mediante la técnica de recursividad un problema específico. • Desarrollar una aplicación que simula una lista de espera para la asignación de mesas en un restaurante. • Desarrollar una aplicación para simular las llamadas a funciones utilizando una pila. • Desarrollar una aplicación para simular el almacenamiento de las páginas visitadas en un navegador de internet haciendo uso del concepto de pilas. • Desarrollar una aplicación que simula la salida de aviones en un aeropuerto utilizando colas de prioridad. • Desarrollar una aplicación utilizando los conceptos de árboles binarios. • Implementar el algoritmo “El viajero” mediante (camino mínimo) grafos. • Construir un grafo ponderado que modele una sección de su estado de origen. Utilice el algoritmo de Dijkstra para determinar el camino más corto, desde el vértice inicial hasta el último vértice. • Realizar ejercicios con mapas para almacenar las estructuras de envío de información con el lenguaje JSON.



9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

Fundamentación: marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.

Planeación: con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.

Ejecución: consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.

Evaluación: es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Las técnicas, herramientas y/o instrumentos sugeridos que permiten obtener el producto del desarrollo de las actividades de aprendizaje: códigos, exposición, mapas mentales o conceptuales, resúmenes, reportes de prácticas, cuadros comparativos, informes. Las técnicas, herramientas y/o instrumentos sugeridos que me permiten constatar el logro o desempeño.

- **Proyectos integradores:** evaluación basada en el desarrollo de proyectos que aplican conceptos de la asignatura a problemas reales de inteligencia artificial. Estos proyectos permitirán a los estudiantes demostrar su capacidad para integrar conocimientos y habilidades en la creación de soluciones innovadoras, evaluando tanto el proceso de desarrollo como el resultado final.
- **Presentaciones orales:** cada estudiante o grupo presentará los resultados de sus proyectos o investigaciones ante la clase. Esta actividad evalúa la capacidad de comunicar ideas complejas de manera efectiva, el uso de soportes visuales adecuados y la habilidad para responder preguntas y recibir retroalimentación constructiva.
- **Exámenes escritos:** se realizarán exámenes que incluyan problemas aplicados, preguntas teóricas y casos prácticos para evaluar la comprensión de conceptos fundamentales de la teoría de grafos, álgebra booleana, lógica matemática y su aplicación en Ciencias de la Computación. Los exámenes escritos permitirán medir el conocimiento individual de cada estudiante de manera objetiva.



- **Evaluación continua de tareas y ejercicios:** a lo largo del semestre, se asignan tareas y ejercicios que cubren diferentes temas de la asignatura. Esta evaluación continua facilita el seguimiento del progreso del estudiante, permitiendo identificar áreas de fortaleza y oportunidades de mejora, además de fomentar la práctica constante de los conceptos aprendidos.

Estos métodos de evaluación están diseñados para proporcionar una visión integral del desempeño de los estudiantes, fomentando el aprendizaje activo, la aplicación práctica de conocimientos y el desarrollo de habilidades clave para su futuro profesional en las ciencias de la computación

11. Fuentes de información

1. "Estructura de Datos en Java" Weiss, M. A. (2013) México. Editorial Pearson Educación.
2. "Estructuras de Datos y Algoritmos en Java":
3. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2010) México. Ed McGrawHill.
4. "Data Structures And Algorithms In PythonbyMichael"Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2016). Ed Wiley india Pvt. Ltd.
5. "Estructura de datos en Java™: compatible con Java™2".
6. Weiss, M. (2010)México: Ed. Addison Wesley.
7. "Estructuras de Datos en C++." Joyanes, L. (2007) España: McGraw – Hill.