



INSTITUTO TECNOLÓGICO DE FRONTERA COMALAPA

MATERIA

Estructura de Datos

TEMA 1

Act 1.3 síntesis del tema ED

ALUMNO

Amilcar Alejandro Corona Escobar

Tercer semestre, ing. Sistemas Computacionales.

NC: 241260074

MODALIDAD

Escolarizada

DOCENTE

Ing. Mingo Velázquez Francisco Javier.

Frontera Comalapa, Chiapas, a 05 de septiembre de 2025

Introducción

El estudio de las estructuras de datos es una parte esencial de la programación y de la informática en general. Estas nos permiten organizar y manejar la información de forma más eficiente, lo que se traduce en programas más rápidos y mejor optimizados. Al aprenderlas, entendemos no solo cómo guardar datos, sino también cómo acceder a ellos y procesarlos de la mejor manera posible.

Justificación

En la programación no basta con que un programa funcione; también debe ser eficiente y práctico. Un mal manejo de datos o de memoria puede hacer que un sistema sea lento o que incluso falle. Por eso es importante conocer las diferentes estructuras de datos, el uso de la memoria y el análisis de algoritmos. Estos conocimientos son herramientas básicas que nos ayudan a resolver problemas reales con soluciones más efectivas.

Las estructuras de datos se clasifican principalmente en dos tipos: lineales (listas, pilas y colas) y no lineales (árboles y grafos). Las lineales organizan los datos de manera secuencial, mientras que las no lineales permiten representaciones más complejas, como jerarquías o redes de información.

Dentro de este tema también se estudian los Tipos de Datos Abstractos (TDA), que son modelos que describen cómo se pueden manipular los datos sin importar cómo están programados. Ejemplos de TDA son las pilas, que funcionan con el principio “último en entrar, primero en salir”, y las colas, que aplican el principio “primero en entrar, primero en salir”. Estos conceptos se aplican en la vida diaria, como en el historial de navegación (pila) o en los turnos de espera (cola).

Otro punto fundamental es el manejo de memoria, que puede ser estática o dinámica. La memoria estática asigna un espacio fijo desde el inicio del programa, como ocurre con los arreglos, mientras que la memoria dinámica se ajusta en tiempo de ejecución, como en las listas enlazadas. Cada una tiene sus ventajas: la estática es sencilla y rápida, mientras que la dinámica es más flexible y eficiente en el uso de recursos.

Finalmente, el análisis de algoritmos nos ayuda a medir el rendimiento de un programa. Se estudia la complejidad en el tiempo (qué tanto tarda en ejecutarse) y la complejidad en el espacio (qué tanta memoria utiliza). Con estas métricas se puede determinar la eficiencia de un algoritmo, lo cual es clave para elegir la mejor solución a un problema.

Conclusión

El tema 1 deja en claro que las estructuras de datos son la base para la creación de programas eficientes y organizados. Comprender cómo se clasifican, cómo se usan los TDA, cómo se maneja la memoria y cómo se analiza la eficiencia de un algoritmo es fundamental para cualquier programador. En definitiva, no solo se trata de que un programa funcione, sino de que lo haga de la manera más rápida y con el mejor uso de recursos posibles.

Fuentes de información

* GeeksforGeeks. (s.f.). *Data Structures - Basics*. Recuperado de: https://www.geeksforgeeks.org/data-structures
* Tutorialspoint. (s.f.). *Data Structures Tutorial*. Recuperado de: https://www.tutorialspoint.com/data\_structures\_algorithms
* Programiz. (s.f.). *Data Structures*. Recuperado de: https://www.programiz.com/dsa