|  |  |  |
| --- | --- | --- |
| Logotipo, nombre de la empresa  El contenido generado por IA puede ser incorrecto. | Imagen que contiene Logotipo  El contenido generado por IA puede ser incorrecto. | Diagrama  El contenido generado por IA puede ser incorrecto. |

**INSTITUTO TECNOLÓGICO DE FRONTERA COMALAPA**

**MATERIA**

**ESTRUCTURA DE DATOS**

**TEMA**

**RESUMEN**

**ESTUDIANTE**

**ISAIAS SALATHIEL LOPEZ TORRES**

**TERCER SEMESTRE, ING. SISTEMAS COMPUTACIONALES.**

**NC:241260085**

**MODALIDAD**

**ESCOLARIZADA**

**DOCENTE**

**ING. FRANCISCO JAVIER MINGO VELAZQUEZ**

**FRONTERA COMALAPA CHIAPAS, A 26 DE AGOSTO DEL 2025**

**INTRODUCCIÓN A LAS ESTRUCTURAS DE DATOS**

**INTRODUCCIÓN**

El tema de estructuras de datos es muy importante porque nos enseña cómo organizar la información de una forma más eficiente dentro de un programa. Al entender bien estos conceptos podemos hacer que nuestros programas trabajen mejor, ocupen menos memoria y se ejecuten más rápido.

**JUSTIFICACIÓN**

Aprender sobre estructuras de datos es fundamental en la carrera de sistemas, ya que siempre vamos a trabajar con información que necesita ser almacenada y procesada. Saber cuándo usar una lista, una pila, una cola o un árbol, nos ayuda a elegir la mejor herramienta según el problema que queremos resolver. De esta manera se logra que los programas sean más prácticos y fáciles de mantener.

En este tema vimos primero la **clasificación de las estructuras de datos**. Están las lineales, como los arreglos, listas, pilas y colas, donde los datos se guardan uno detrás de otro. También están las no lineales, como los árboles y los grafos, que sirven cuando hay relaciones más complejas entre los datos.

Después revisamos los **tipos de datos abstractos (TDA)**, que son modelos que nos dicen qué operaciones se pueden hacer, sin importar cómo se implementan en memoria. Ejemplos claros son las pilas (LIFO), las colas (FIFO) y las listas enlazadas.

Otro punto fue el **manejo de memoria**. Aquí se diferencia entre memoria estática, que se define al inicio y no cambia, y la memoria dinámica, que se puede modificar durante la ejecución del programa.

Por último, vimos el **análisis de algoritmos**, donde se estudia el tiempo y la memoria que ocupan. Es importante buscar siempre un equilibrio para que el programa sea rápido y no consuma tantos recursos.

**CONCLUSIÓN**

Con este tema entendí que las estructuras de datos no solo son teoría, sino que realmente hacen la diferencia a la hora de programar. Saber cuál usar y cómo aplicarla nos ayuda a resolver los problemas de manera más eficiente. Al final, elegir bien una estructura puede ser la clave para que un programa funcione correctamente y sin desperdiciar recursos.

**FUENTES DE INFORMACIÓN**

* GeeksforGeeks — “Data Structure Types, Classifications and Applications”
* GeeksforGeeks — “Static Data Structure vs Dynamic Data Structure”
* Simplilearn — “What is Data Structure: Types, & Applications”
* Wikipedia — “Data structure”