**TECNOLOGICO NACIONAL DE MEXICO**

**INSTITUTO TECNOLOGICO DE ORIZABA**

**Asignatura:** Estructura de Datos

**Carrera:** Ingeniería en Informática

**Estudiantes:**

* Hernandez Angón Diego - No. Control 21010193
* Dorantes Rodríguez Diego Yael- No. Control 21010184

**Profesora:** María Jacinta Martínez Castillo

**Grupo:** 3a3A

**Fecha de entrega:** 22/04/2023

**Reporte de unidad 1: Introducción a las estructuras de datos**

INTRODUCCION

En el siguiente reporte, abordaremos el tema de los tipos de datos abstractos, utilizando recursos como un IDE de programación, además de adquirir conocimiento acerca de los TDAs, obteniendo como resultado un proyecto enfocado a los datos ordenados y datos desordenados. Aprenderemos a diferencias las estructuras e datos tanto lineales como las no lineales

COMPETENCIA ESPECIFICA

Desarrollar la habilidad de poder reconocer e identificar los tipos de datos abstractos, mediante la utilización de estructuras de datos lineales y no lineales, además de poder trabajar correctamente con datos desordenados y datos ordenados mediante código de programación

MARCO TEORICO

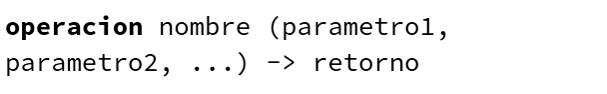
Las estructuras de datos son una manera específica de ordenar y organizar la información en una computadora, con el fin de hacer un uso más efectivo de ella. Existen distintos tipos de estructuras de datos, cada uno apropiado para diferentes propósitos, incluso algunas están diseñadas especialmente para realizar tareas muy particulares. Estas estructuras de datos son una herramienta importante para manejar grandes volúmenes de información de manera efectiva, en aplicaciones como bases de datos extensas o servicios en la web.

Un Tipo de Dato Abstracto (TDA) es un modelo que define valores y las operaciones que se pueden realizan sobre ellos. Y se denomina abstracto ya que la intención es que quien lo utiliza, no necesita conocer los detalles de la representación interna o bien el cómo están implementadas las operaciones.

Es por esto una práctica que nos provee un grado de abstracción que permite desacoplar al código que usa un TDA de aquel código que lo implementa.

Ejemplo:

Se define un tipo Pantalla y dos operaciones sobre él.

La operación tiene un nombre y, como una función, un dominio e imagen. Se lee así

En nuestro caso simple, tenemos dos funciones: para escribir en pantalla, y para leer.

Fíjense que, mediante este tipo, yo puedo escribir código que opere con la pantalla, sin conocer cómo realmente se escribe o lee de ésta.

* Memoria dinámica

Es memoria que se reserva en tiempo de ejecución. Su principal ventaja frente a la estática, es que su tamaño puede variar durante la ejecución del programa. (En C, el programador es encargado de liberar esta memoria cuando no la utilice más). El uso de memoria dinámica es necesario cuando a priori no conocemos el número de datos/elementos a tratar.

* Memoria estática

Es el espacio en memoria que se crea al declarar variables de cualquier tipo de dato (primitivas [int, char] o derivados [struct, matrices, punteros]). La memoria que estas variables ocupan no puede cambiarse durante la ejecución y tampoco puede ser liberada manualmente

MATERIAL Y EQUIPO

El proyecto se lleva a cabo con los debidos requisitos necesarios. Contar con una computadora con los componentes adecuados, capaz de soportar IDES de programación con lenguaje de java. Se necesita de software específico para la creación de código en Java, el cual puede ser Eclipse o NetBeans. Se cuenta además con los conocimientos otorgados por la profesora en clase, además de material de apoyo para reforzar el conocimiento

DESARROLLO Y RESULTADOS

Para el desarrollo de los códigos para datos ordenados y datos desordenados

CONCLUSIONES

Hemos llegado a la conclusión de que, utilizando los TDA podemos ser capaces de crear códigos para datos ordenados y desordenados, y algunas de las estructuras de datos más comunes como lo son matrices, listas enlazadas, pilas, colas, árboles y grafos. La comprensión de las estructuras de datos es un conocimiento esencial para cualquier programador, y su aplicación adecuada puede mejorar significativamente la eficiencia y la capacidad de manejo de los datos de un programa.