

Estructuras de Datos Abstracción de Datos

Ing. Armandina Leal Flores

Tipo de Dato

- ▶ Dentro de un lenguaje de programación, un **Tipo de Dato** es definido como:
 - Un rango de valores y
 - El conjunto de operaciones válidas que se pueden aplicar a esos valores.
- ▶ Ejemplo:
 - Tipo de dato int cuyo rango de valores es desde (-2,147,483,648) a (2,147,483,647) y sobre los valores se pueden aplicar las operaciones de suma, resta, etc.

Estructura de Datos

- ▶ **Es un tipo de dato que:**

- Puede descomponerse en un conjunto de elementos, cada uno de los cuales es simple, o bien es otra estructura de datos.
- Incluye un conjunto de asociaciones o relaciones entre los elementos.

- ▶ **Ejemplo:**

- Un arreglo es una estructura de datos predefinida en el lenguaje.

Estructura de Datos

- ▶ Conjunto de datos almacenados en la memoria de la computadora o en un disco.
- ▶ Ejemplos:
 - Arreglos, Listas Encadenadas, Pilas, Filas, Árboles Binarios y Tablas de hash.

Robert Lafore

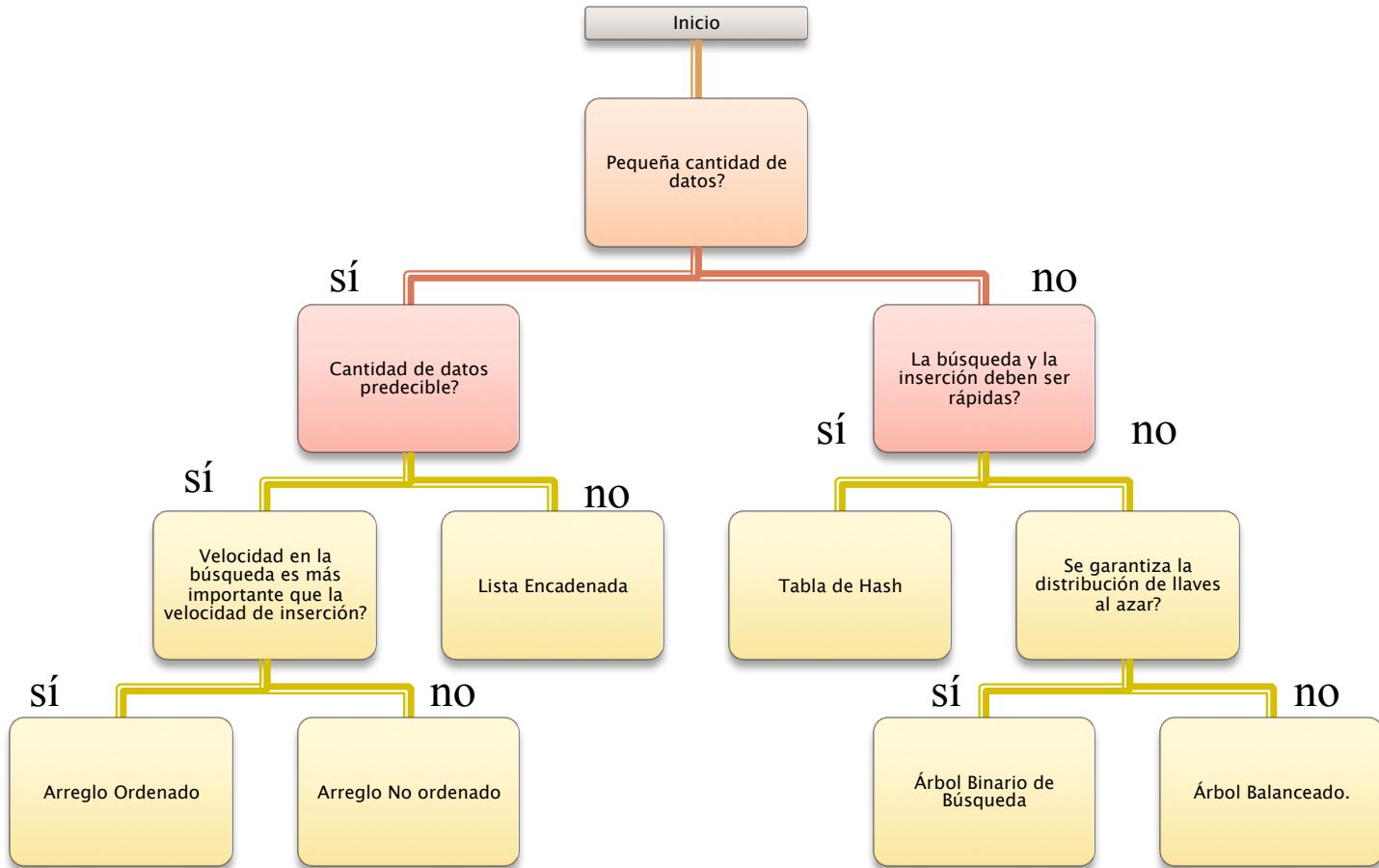
En qué se utilizan?

- ▶ **Almacenamiento de datos reales**
 - Ejemplo. Base de datos del Departamento de Tránsito.
 - ¿Cómo almacenar los datos si son cientos, miles o millones de datos? ¿Cómo agrego, borro o modiflico un dato rápidamente? ¿Cómo busco rápidamente un dato? ¿Cómo ordeno los datos?,
- ▶ **Herramientas de programación**
 - Estructuras que almacenan datos que no son accesados por el usuario
 - Ejemplo: Pilas, Filas, Filas Priorizadas
 - Ejemplo: Fila de impresión
- ▶ **Modelación del mundo real**
 - Ejemplo: Representación de rutas de aerolíneas (Grafo)
 - Ejemplo: Simular una fila de espera (Fila).

Comparación

Estructura	Ventaja	Desventaja
Arreglo	Inserción rápida. Acceso rápido si se conoce el índice.	Tamaño fijo. Buscar y borrar elementos es lento.
Arreglo Ordenado	La búsqueda es más rápida comparada con un arreglo no ordenado.	Tamaño fijo. Aregar y borrar elementos es lento.
Pila (Stack)	Último dato en entrar es el primero en salir.	Acceso lento para los demás elementos.
Fila (Queue)	Primer dato en entrar primero en salir.	Acceso lento para los demás elementos.
Listas Encadenadas (Linked List)	Agregar y borrar elementos es rápido.	Buscar elementos es lento.
Árbol Binario (Binary Tree)	Agregar, buscar y borrar datos es rápido si está balanceado.	El algoritmo para borrar datos es complejo.
Tabla de Hash (Hash Table)	Acceso rápido si se conoce la llave. Inserción rápida.	Borrar un dato es lento. Buscar un datos es lento si no se conoce la llave. Uso ineficiente de memoria.
Grafo (Graph)	Modela situaciones del mundo real	Algunos algoritmos son lentos y complejos.

Comparación



Abstracción de Datos

- ▶ Metodología que se utiliza para diseñar Estructuras de Datos.
- ▶ Durante el diseño de la estructura, se ignoran los detalles de la implementación como por ejemplo en qué lenguaje se va a implementar.

Abstracción

Proceso mental que extrae los rasgos esenciales de “algo” para representarlos por medio de un lenguaje gráfico o escrito.

Tipo de Datos Abstracto

► ADT (Abstract Data Type)

- Es un tipo de dato, que combina datos y operaciones.
- Los datos almacenados en el ADT se encuentran escondidos y solo se puede tener acceso a ellos a través de las operaciones.
- Las estructuras de datos se pueden diseñar como un ADT.

Diseño de una Estructura de Datos

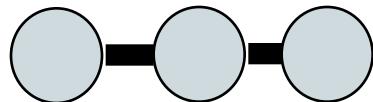
► Elementos a especificar:

- Elementos
- Tipo de Organización
- Operaciones

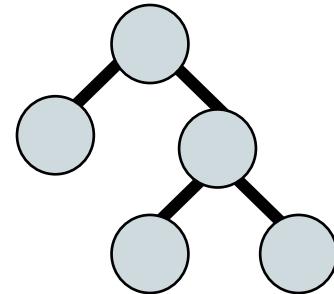
Elementos

- ▶ Tipo de dato de los elementos que almacenará la estructura.
 - Ejemplos:
 - Fechas
 - Cadenas
 - Datos de un curso

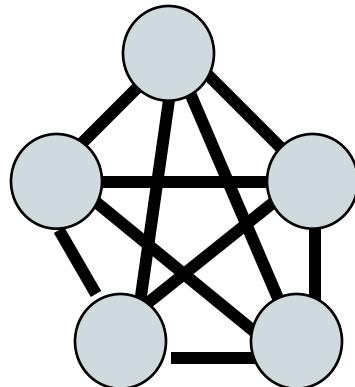
Tipo de organización



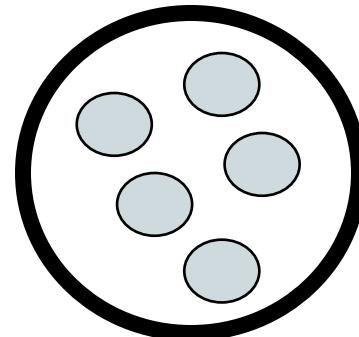
Lineal
Relación uno a uno



Jerárquica
Relación uno a muchos



Red
Relación muchos a muchos



Sin relación

Operaciones

- ▶ Para cada una de las operaciones que se deben realizar sobre la estructura se debe especificar lo siguiente:
 - Nombre de la operación
 - Descripción breve de su utilidad
 - Datos de entrada a la operación
 - Datos que se generan como salida
 - Precondiciones
 - Postcondiciones

Precondición

▶ Precondición:

- Es la condición que debe cumplirse antes de ejecutar la operación, para que ésta funcione correctamente.

▶ Ejemplo:

- Si la operación tiene como función obtener la raíz cuadrada de un número, la precondición sería que el número debe ser positivo (ya que no se puede sacar la raíz cuadrada de un número negativo).

Postcondición

▶ Postcondición:

- Descripción del estado en el que queda el ADT después de ejecutar la operación.

▶ Ejemplo:

- Si la operación ordena un arreglo de valores enteros de menor a mayor, la Postcondición podría ser: “El arreglo conserva los mismos valores pero ahora ordenados de menor a mayor”.

Niveles de Abstracción

► Lógico o abstracto.

- En este nivel se diseña el ADT por medio de la especificación lógica.

► Físico o de implementación.

- En este nivel se implementan cada una de las operaciones del ADT.

► Aplicación o de uso.

- En este nivel se implementan programas en la que se utiliza la Estructura de Datos (ADT).

Recomendación....

- ▶ Como buenos profesionales en el área, debemos respetar los niveles de abstracción es decir:
 - Cuando estamos en el nivel Lógico no debemos estar pensando en cómo la vamos a implementar o cómo se va a utilizar la estructura.
 - Cuando estamos en el nivel de Aplicación no debemos ni siquiera considerar la posibilidad de modificar el ADT.
 - Cuando estamos en el nivel Físico no debemos pensar en cambiar el diseño o en hacer cosas que son específicas a una cierta aplicación.
- ▶ Esto permitirá crear ADT más útiles.

¿En qué nivel de abstracción....

- ▶ Si la precondición de la operación que obtiene la raíz cuadrada de un número es: “el número debe ser positivo” ¿en qué nivel de abstracción se debe verificar que el número al que se le va a sacar la raíz cuadrada es positivo?
- ▶ Respuesta:
 - Nivel de Aplicación (ya que se debe verificar que el número es positivo ANTES de llamar a la operación).

Características de la Programación Orientada a Objetos (POO)

▶ Encapsulación

- Empaca los datos y su correspondiente código, ocultando detalles específicos al exterior.

▶ Polimorfismo

- Ejecución de una misma operación de diferentes formas dependiendo de los objetos involucrados.

▶ Herencia

- Capacidad de definir nuevos objetos heredando características y comportamientos, sin necesidad de volver a definir, y con opción a redefinir lo necesario.

¿Por qué utilizar POO para implementar una Estructura de Datos?

- ▶ El usuario de la Estructura de datos NO puede invadir el nivel físico de la misma, por lo que se cumple la abstracción de datos.
- ▶ La Estructura de datos se convierte en un paquete de software REUTILIZABLE.
- ▶ La POO convierte al proceso de desarrollar software en un proceso de fabricación de piezas de software, que a su vez sirven para construir otras piezas.

Ejercicios

- ▶ Escribe la definición de una función que sea útil para ordenar los N valores almacenados en un arreglo.
- ▶ Define el ADT Complex que permita almacenar un número complejo y que tenga las operaciones aritméticas +, -, *, /, ==, !=, >>, <<.