

## Clasificación de la estructura de datos

Son estructuras que pueden aumentar o disminuir su tamaño mientras se ejecuta el programa

Las estructuras dinámicas son generadas mediante un tipo de dato conocido con el nombre de puntero. Una variable de tipo puntero almacena la dirección o posición de otra variable, variable dinámica (o anónima), y la principal ventaja que representa manejar punteros es que se pueden adquirir posiciones de memoria a medida que se vayan necesitando y liberarlas cuando ya no se requieran. Una estructura de datos dinámica es una colección de elementos llamados nodos, normalmente de tipo registro, que se enlazan o encadenan unos con otros. El enlace se establece gracias a que cada nodo contiene al menos un elemento de tipo puntero que apunta a otro nodo de la estructura. Estos punteros permiten el cambio y la liberación de las posiciones de memoria que tienen asociadas y, de esta manera, la estructura dinámica puede modificar su tamaño en tiempo de ejecución.

Existen diferentes tipos de estructuras dinámicas de datos, siendo las más notables y significativas las listas enlazadas, los árboles y los grafos.

Cualquier colección o grupo de datos organizados de tal forma que tengan asociados un conjunto de operaciones para poder manipularlos, se dice que conforma una estructura de datos.

Por ejemplo, cualquier lenguaje de alto nivel provee típicamente de tipos de datos estructurados o estructuras de datos predefinidas, como los arreglos o los registros. Un arreglo es un conjunto de datos, todos del mismo tipo, con una organización lineal y con métodos claros de acceso a través de sus subíndices. Las operaciones tradicionales sobre los arreglos incluyen la comparación, la asignación, la escritura, etc. En un nivel más bajo, podría verse a los números enteros como estructuras de datos: se componen de un grupo de dígitos y tienen asociadas operaciones como sumar, restar y multiplicar, entre otras cosas.

## Elementos que conforman la estructura de datos

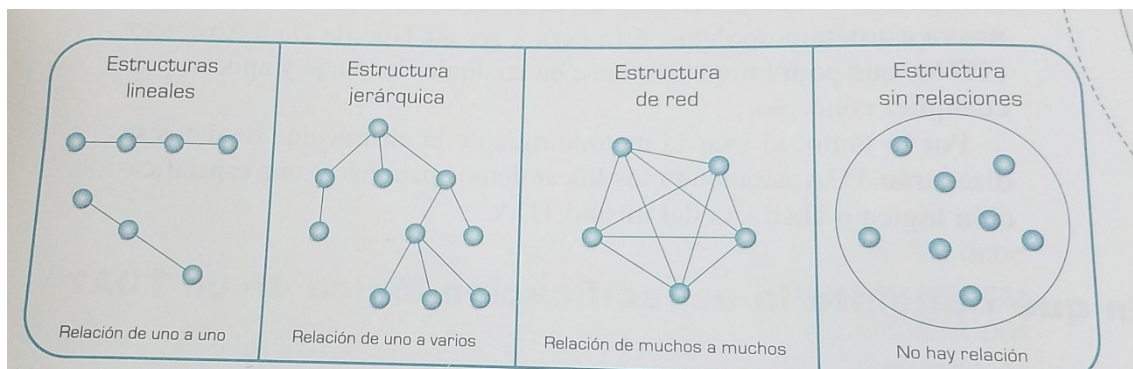
En este punto se describe el tipo de datos individuales que guardará la estructura. Por ejemplo, números enteros, caracteres, fechas, registros con los datos de un empleado, etcétera.

## Tipo de organización en que se guardarán los elementos

Existen solamente cuatro tipos de organización para los datos en la estructura, la cual deberá tener alguna de las siguientes organizaciones:

- Lineal: Si hay una relación de uno a uno entre los elementos.
- Jerárquica: Si hay una relación de uno a muchos entre los elementos.
- Red: Si hay una relación de muchos a muchos entre los elementos.
- Sin relación: Si no hay relaciones entre los elementos.

Las relaciones entre los elementos se dan, por ejemplo, cuando existe algún motivo para que un elemento esté antes o después que otro.



## Dominio de la estructura

Este punto es opcional, en él se describe la capacidad de la estructura en cuanto al rango posible de datos por guardar.

## Descripción de las operaciones de la estructura

Cada operación relacionada con la estructura debe describirse con los siguientes puntos:

- Nombre de la operación.
- Descripción breve de su utilidad.
- Datos de entrada a la operación.
-

## Tipos de representaciones existentes para una estructura de datos

Independientemente de las facilidades de implementación que ofrezcan los lenguajes de programación, una estructura de datos (TDA) puede representarse de dos formas:

- Por posiciones (almacenamiento contiguo).
- Por ligas (almacenamiento disperso).

### Funcionamiento de representación de una estructura de datos por posiciones

En este tipo de representación, el lugar físico donde se almacena un elemento determina automáticamente su posición relativa en la estructura de datos. Básicamente, se puede pensar en ella como un espacio de almacenamiento contiguo donde cada lugar sirve para almacenar un elemento, de tal forma que, si un elemento está almacenado en el lugar  $K$ , tendrá la  $K$ -ésima posición dentro de la estructura y, obviamente, el elemento  $K + 1$  estará ubicado exactamente después del  $K$  y se almacenará en el lugar  $K + 1$  del espacio de almacenamiento.

## Referencias

1. Estructura de datos: Referencia práctica con orientación a objetos, Román Martínez, THOMSON LEARNING, 2004, México, Pag 3.
2. Estructura de datos: Referencia práctica con orientación a objetos, Román Martínez, THOMSON LEARNING, 2004, México Pag 4.
3. Estructura de datos: Referencia práctica con orientación a objetos, Román Martínez, THOMSON LEARNING, 2004, México Pag 30.
4. Estructura de datos> Libro de problemas, Luis Joyanes Aguilar, McGrawHill, 1999, España, Pag 87.