

DEV.F.:

# Clase 5

**Estructuras de datos**

**Comenzamos en 10 min**

6:40pm (hora CDMX)

Presentación creada por contribución de Diego Lechuga



# | Temas de la clase (180 min)

## Estructuras de Datos

Concepto

Tipos

Arreglos y Objetos

## Estructuras Lineales

Pilas

Colas

Listas Enlazadas

## Estructuras No Lineales (Concepto)

Árboles

Grafos

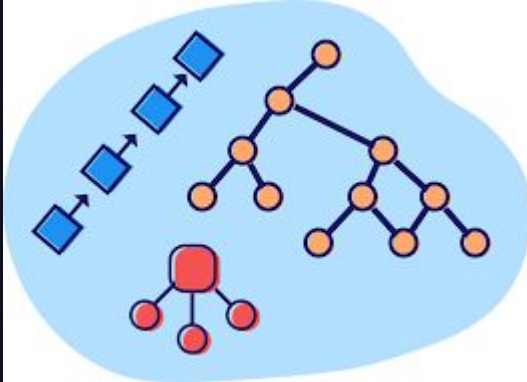
## Reto de la semana



DEV.F.

# | Estructuras de Datos

Una estructura de datos es una manera de organizar, manejar y almacenar datos para que puedan utilizarse de manera eficiente. Cada estructura ofrece diferentes **métodos** para realizar operaciones como inserción, eliminación, búsqueda y acceso a los datos, dependiendo del problema que se quiera resolver.



DEV.F:



# | Tipos de Estructuras de Datos

Existen dos tipos de **estructuras de datos**, por un lado tenemos las **estructuras lineales** que están conformadas por:

- Arreglos
- Pilas
- Colas
- Listas Enlazadas

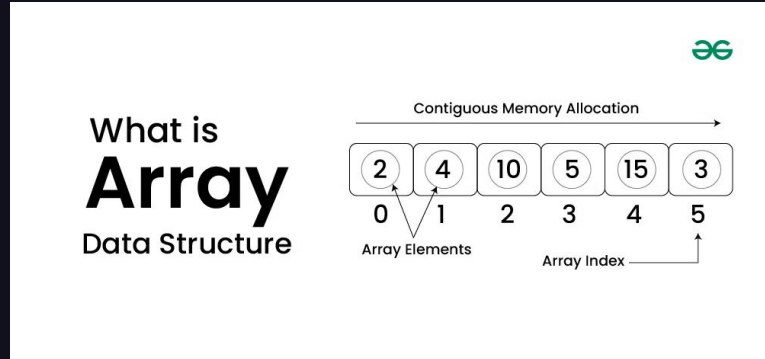
Y así mismo tenemos las **estructuras no lineales** de las cuales tenemos ejemplos como:

- Árboles
- Grafos
- Objetos (Tablas de Hash)



# | Arreglos

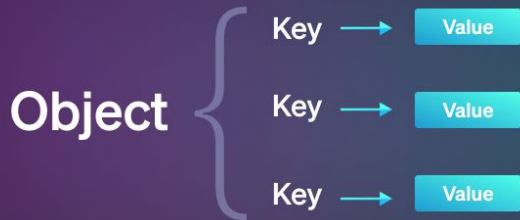
Un arreglo (o array) en programación es una estructura de datos que permite almacenar un conjunto de elementos del mismo tipo bajo un nombre común. Los elementos en un arreglo se organizan de manera contigua en memoria y se accede a ellos utilizando un índice o subíndice.





# | Objetos

En JavaScript, un objeto es una estructura de datos que permite almacenar colecciones de datos mediante pares de clave-valor. Cada clave (también llamada "propiedad") está asociada a un valor, que puede ser de cualquier tipo de dato (números, cadenas, funciones, etc.).

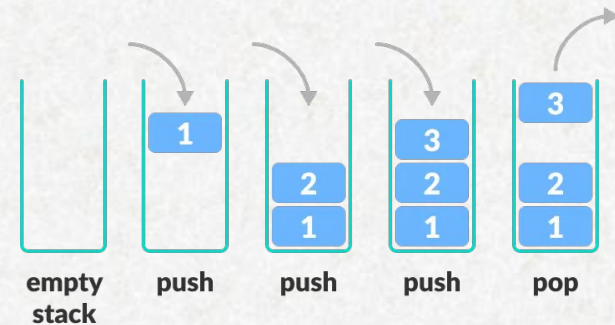




# Pilas

Una pila (en inglés, stack) es una estructura de datos que sigue el principio LIFO (Last In, First Out), lo que significa que el último elemento que se agrega es el primero en ser eliminado. Esta estructura funciona de manera similar a una pila de libros: agregas un libro en la parte superior y para sacar uno, debes quitar el que está en la parte superior.

- Último en entrar, primero en salir (LIFO): El último elemento agregado es el primero en ser eliminado.
- Operaciones básicas:
  - push: Agregar un elemento al final de la pila.
  - pop: Eliminar el último elemento agregado (el de la parte superior de la pila).
  - peek o top: Ver el valor del elemento en la parte superior sin eliminarlo.
  - isEmpty: Comprobar si la pila está vacía.



# Colas

Una cola (en inglés, queue) es una estructura de datos que sigue el principio FIFO (First In, First Out), lo que significa que el primer elemento que se agrega es el primero en ser eliminado. Funciona de manera similar a una fila de personas esperando: el primero en llegar es el primero en ser atendido.

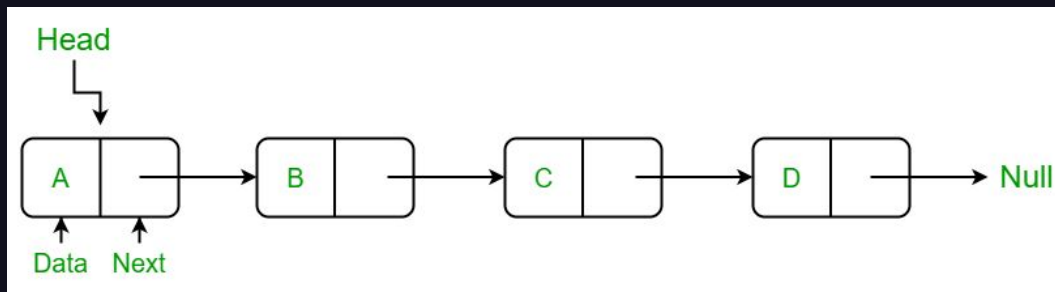
- **Primero en entrar, primero en salir (FIFO):** El primer elemento agregado es el primero en ser eliminado.
- **Operaciones básicas:**
  - **enqueue():** Agregar un elemento al final de la cola.
  - **dequeue():** Eliminar el primer elemento agregado.
  - **front()** o **peek():** Ver el valor del primer elemento sin eliminarlo.
  - **isEmpty():** Comprobar si la cola está vacía.





# | Lista Enlazada

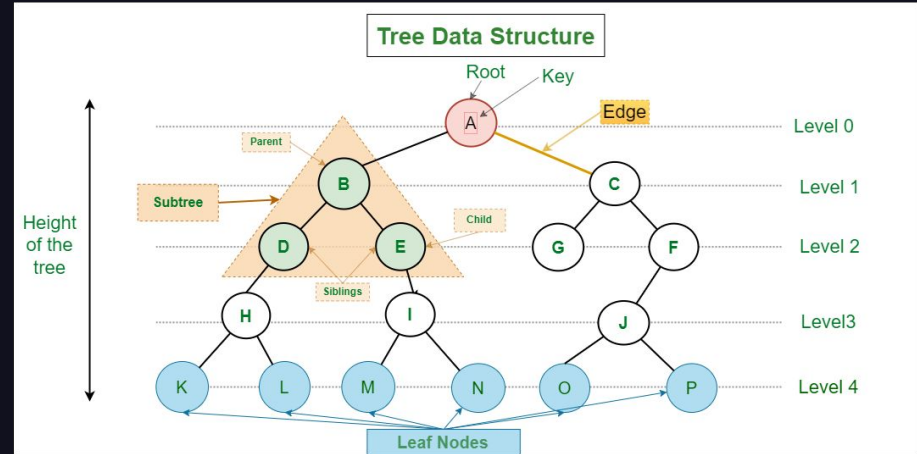
Una lista enlazada es una estructura de datos lineal en la que los elementos (llamados nodos) están conectados entre sí mediante enlaces o referencias. Cada **nodo** contiene dos partes principales: un **valor** o **dato** y un enlace que apunta al siguiente nodo en la **secuencia**. A diferencia de los arreglos, las listas enlazadas no almacenan los elementos de manera contigua en memoria, lo que permite un **manejo dinámico del tamaño**.



# | Árboles



Un árbol es una estructura de datos jerárquica compuesta por nodos conectados por **enlaces** o aristas, donde cada nodo puede tener **hijos**. El nodo superior es llamado raíz, y cada nodo puede tener múltiples nodos descendientes, formando una estructura ramificada. Los nodos que no tienen hijos se llaman hojas.

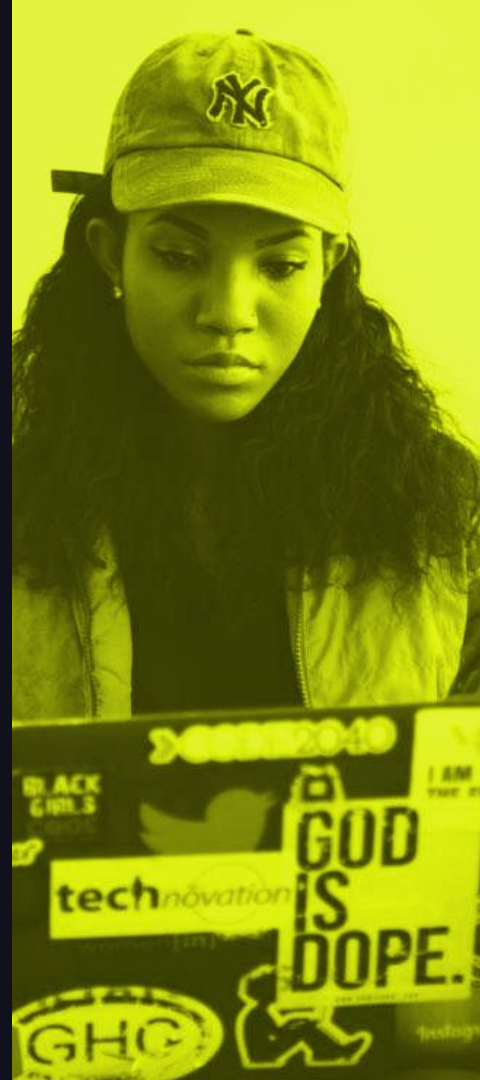
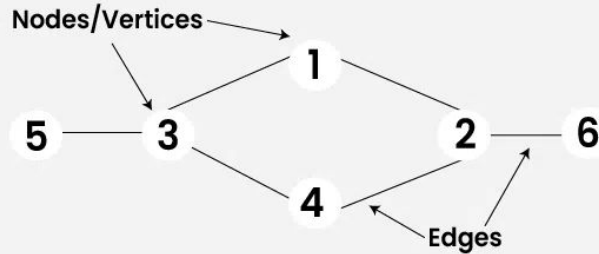


# | Grafos

Un grafo es una estructura de datos que consiste en un conjunto de **nodos** (también llamados vértices) conectados por **aristas** (o bordes). Los grafos se utilizan para representar **relaciones** o conexiones entre diferentes elementos.



## Graph Data Structure



# Visualising data structures and algorithms through animation

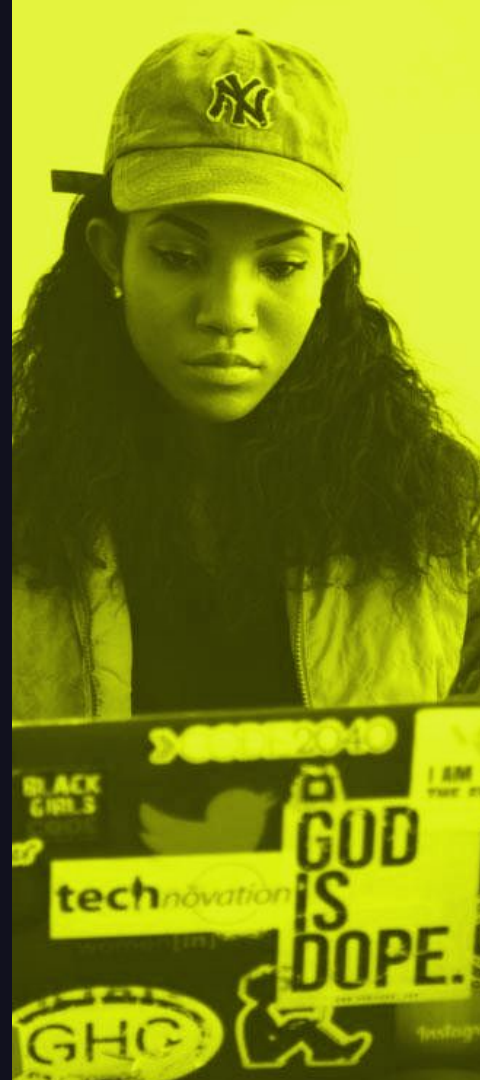
<https://visualgo.net/en>

The grid displays 10 visual algorithm cards:

- Segment Tree**: A binary tree with nodes containing pairs like [0,3], [1,2], [0,0], [1,1], [2,2], [0,1], [1,3]. Tags: dynamic, range, sum, min, max, cs2233.
- Recursion Tree/DAG**: A tree diagram for a function f(4). Tags: dynamic programming, dp, generic, cs1010.
- Graph Traversal**: A graph with 6 nodes and 7 edges. Tags: bfs, dfs, k5003, cs2040, bipartite, scc.
- Min Spanning Tree**: A graph with 6 nodes and 7 edges, showing a subset of edges. Tags: mst, prim, kruskal, graph, min, spanning.
- SS Shortest Paths**: A graph with 6 nodes and 7 edges. Tags: ssdp, single-source, bfs, dijkstra, bellman for.
- Cycle Finding**: A graph with 6 nodes and 7 edges, showing a cycle. Tags: floyd, tortoise-hare, math, cs2233, algorithm.
- Suffix Tree**: A tree diagram for the string "BANANAS". Tags: string, matching, lrs, lcs, cs2233, suffix.
- Suffix Array**: A table showing the suffix array for "BANANAS".

i	SA[i]	Suffix
0	6	S
1	5	AS
2	3	ANAS
3	1	ANANAS
4	0	BANANAS
5	4	NAS
6	2	NANAS

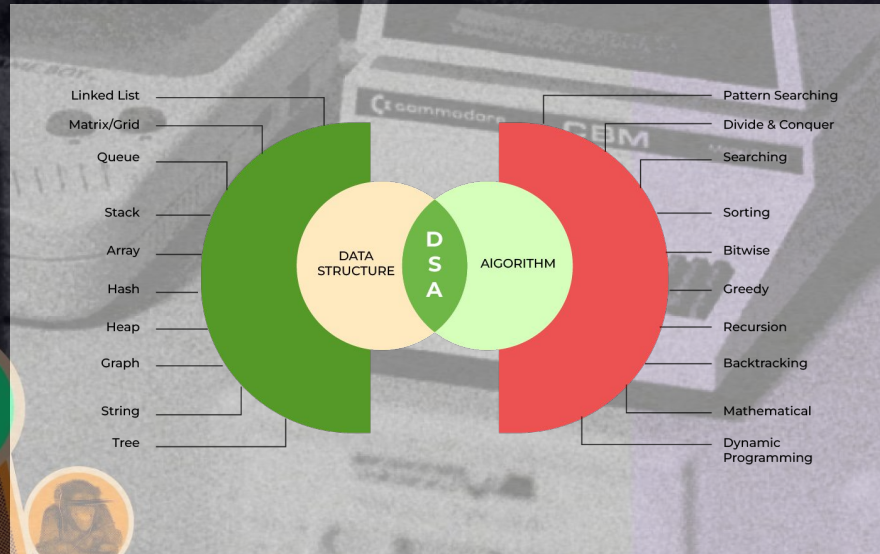
Tags: lcp, cs2233, matching, lrs, lcs, suffix.
- Geometry (Polygon)**: A graph with 6 nodes and 7 edges, showing a polygon. Tags: convex, cut, winding, concave, cs2233, g.
- Convex Hull**: A graph with 6 nodes and 7 edges, showing a convex hull. Tags: andrew, monotone chain, graham, scan, jarv.



# Reto de la semana: Practicar Pilas y Colas



- Aprender cómo se programan
  - Boilerplate code
  - Aprender a replicar los métodos de estos



CONST : DEV.F

GENIO O HACKER?

DEV.F FUNCTION ()

404

SOMOS DEV.F

DEV.F = EDUCATION

&&



No olviden



# Las lecturas de **edu.devf.la**

