**¿Qué son las estructuras de datos?**

Es un modo de representar información en una computadora que, además, cuentan con un comportamiento interno. La cual significa que se rige por determinadas reglas/restricciones que han sido dadas por la forma en que está construida internamente. En otras palabras, es una forma particular de organizar información en un computador para que pueda ser utilizada de manera eficiente. Diferentes tipos de estructuras de datos son adecuados para diferentes tipos de aplicaciones, y algunos son altamente especializados para tareas específicas.

Las estructuras de datos son medios para manejar grandes cantidades de información de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Algunos métodos formales de diseño de lenguajes de programación destacan las estructuras de datos, en lugar de los algoritmos, como el factor clave de organización en el diseño de software.

Es muy importante conocerlas ya que cuando estamos iniciando en el mundo de la programación e intentamos sumergirnos en el mundo de JavaScript, las estructuras de datos se vuelven fundamentales. Para dar a Conocer sobre listas, árboles y grafos que nos permitirá organizar mejor la información y crear código más eficiente.

**Tipos de estructuras de datos**

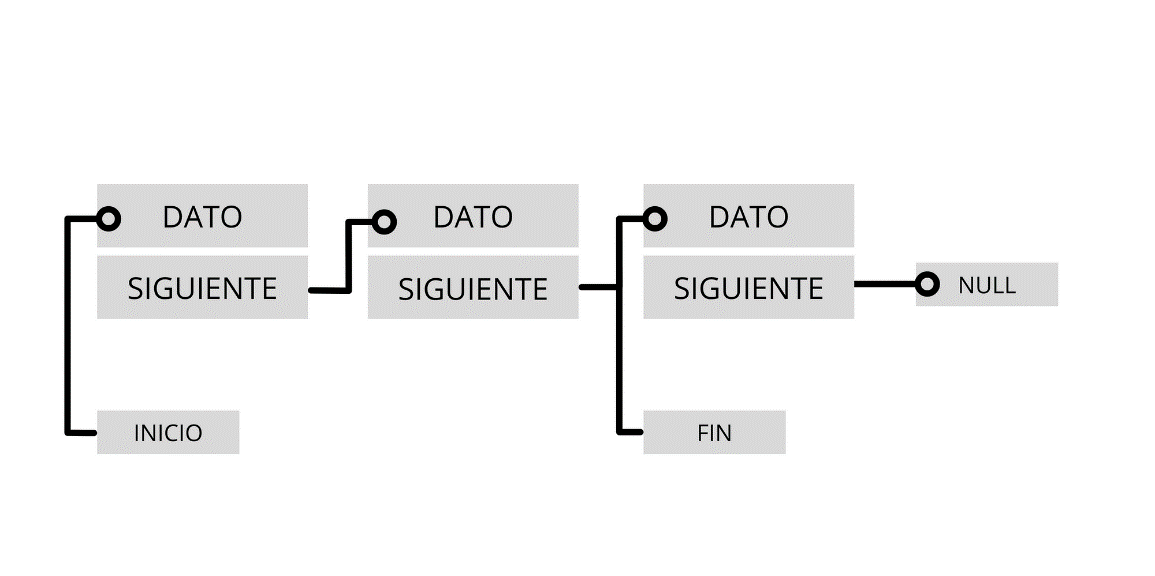
**Estructura de datos lineales**

son aquellas en las que los elementos ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y predecesor, es decir, sus elementos están ubicados uno al lado del otro relacionados en forma lineal.

Existen 3 tipos de estructuras lineales que son:

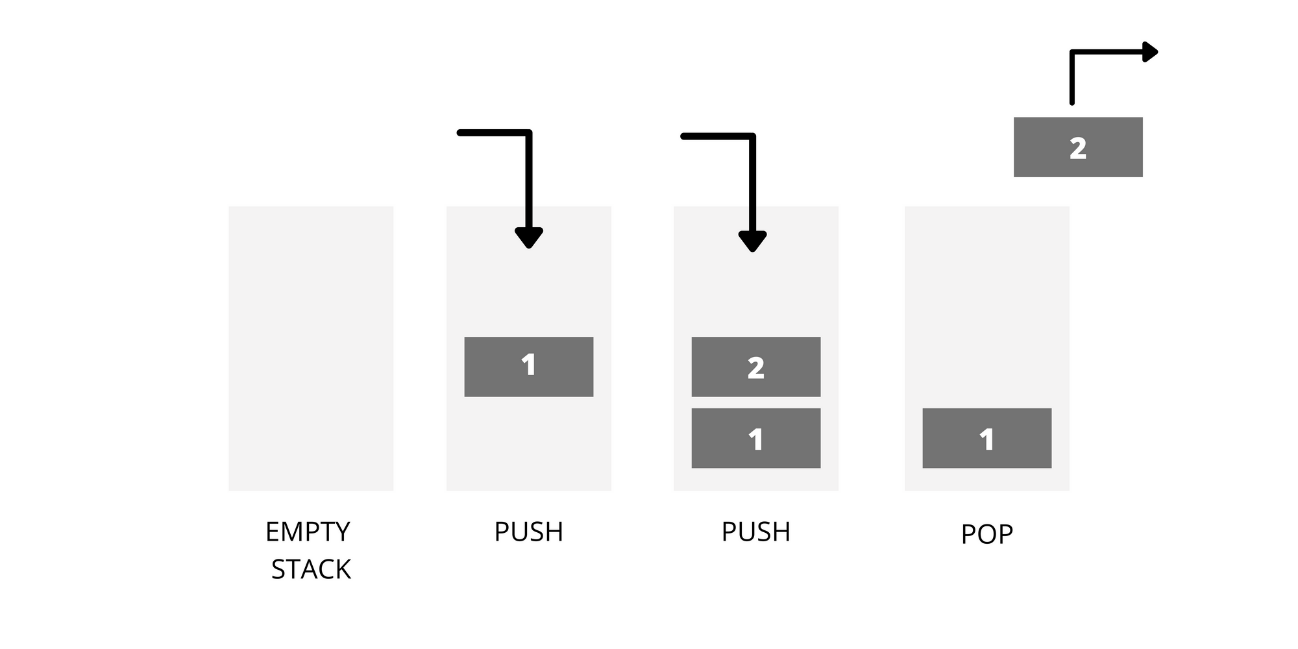
* **Listas enlazadas**

se construyen con elementos que están ubicados en una secuencia. cada elemento se conecta con el siguiente a través de un enlace que contiene la posición del siguiente elemento. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma.



* **Pilas**

es un tipo especial de lista lineal dentro de las estructuras de datos dinámicas que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés Last In, First Out, es decir, último en entrar, primero en salir). ¿Cómo funciona? A través de dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, desapilar (pop), que retira el último elemento apilado.

* **Colas**

Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista. Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, cabeza) de la lista. En otras palabras, Los elementos se eliminan (se quitan) de la cola en el mismo orden en que se almacenan y, por consiguiente, una cola es una estructura de tipo FIFO (first-in-first-out , primero en entrar, Primero en salir o bien primero en llegar/primero en ser servido).

El servicio de atención a clientes es un ejemplo típico de cola o el cajero de un banco.

Gráfico

Descripción generada automáticamente



**Estructura de datos no lineales**

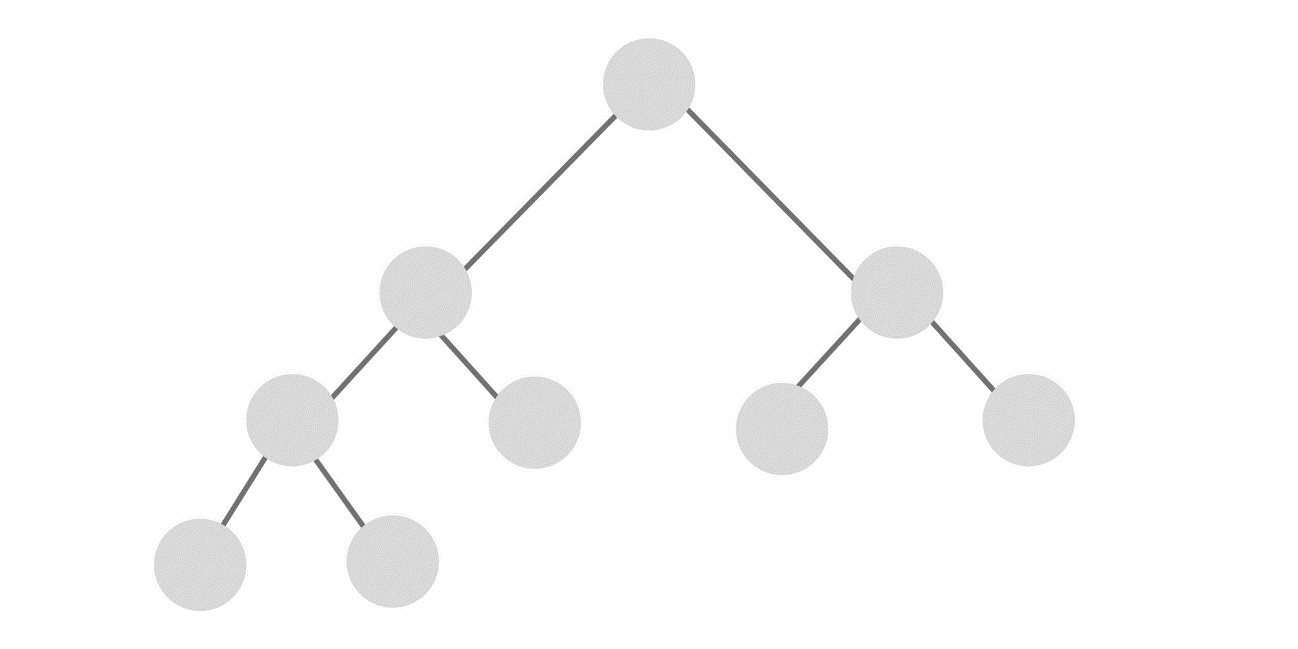
Las estructuras de datos no lineales, también llamadas multienlazadas, son aquellas en las que cada elemento puede estar enlazado a cualquier otro componente. Es decir, cada elemento puede tener varios sucesores o varios predecesores.

Existen dos tipos:

* **Árboles**

los árboles consisten en una estructura no lineal que se utiliza para representar datos con una relación jerárquica en la que cada elemento tiene un único antecesor y puede tener varios sucesores.

Los mismos se encuentran clasificados en: árbol general, un árbol donde cada elemento puede tener un número ilimitado de sub árboles y árboles binarios, que son una estructura de datos homogénea, dinámica y no lineal en donde a cada elemento le pueden seguir como máximo dos nodos.



* **Grafos**

Otro tipo de no lineal de estructura de datos en programación, son los grafos. Se trata de una estructura matemática formada por un conjunto de puntos —una estructura de datos— y un conjunto de líneas, cada una de las cuales une un punto a otro. Los puntos se llaman nodos o vértices del grafo y las líneas se llaman aristas o arcos.

Dibujo de un animal

Descripción generada automáticamente con confianza baja

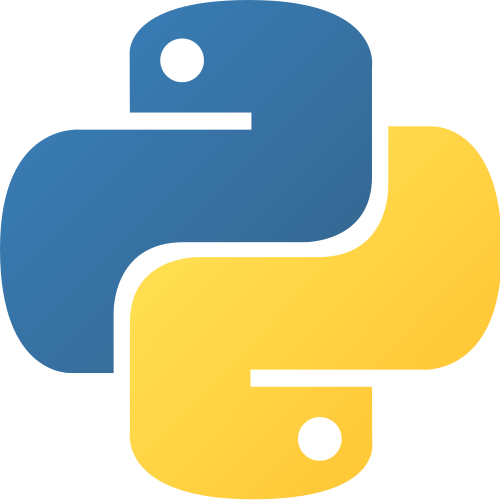


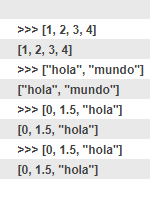
**Estructura de datos en Python**

Las estructuras de datos en Python se pueden entender como un tipo de dato compuesto, debido a que en una misma variable podemos almacenar una estructura completa con información. Dichas estructuras, pueden tener diferentes características y funcionalidades. De hecho, existen múltiples tipos de estructuras de datos en Python

Las estructuras de datos más comunes en Python son:

* **Listas**

Las listas o arrays en Python son estructuras de datos muy flexibles en las que podemos mezclar valores de varios tipos, o bien que sean de un solo tipo. Su declaración es sencilla y obedece un formato JSON estándar.



* **Tuplas**

Las tuplas son como las listas, excepto que en este caso son inmutables. Una tupla consiste en varios valores separados por comas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una cosa importante que hay que notar con las tuplas, es que como los paréntesis también se usan en python para agrupar expresiones, si quisiéramos crear una tupla con un solo valor simplemente agregamos una coma al final de esta manera:

Icono

Descripción generada automáticamente Interfaz de usuario gráfica

Descripción generada automáticamente

* **Sets**

Los sets en Python son listas sin un orden específico pero cuyos elementos son únicos, es decir, no existe la repetición. Y para definir uno hay que explícitamente indicar que queremos que la lista sea un set con la función con el mismo nombre,

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* A partir de Python 2.7, podemos usar llaves para definir sets
* Podemos agregar nuevos elementos con la función add
* Igual que en las listas, podemos revisar si hay un elemento en el set usando la función in
* **Strings**

Una de las cosas curiosas sobre python, es que en muchas maneras podemos manipular strings de la misma manera como se manipulan las listas. Esto convierte a los strings en una especie de estructura de datos conveniente de usar.

Podemos usar len en un string

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Y encontrar partes o segmentos del string según nos parezca conveniente.

Icono

Descripción generada automáticamente Tabla

Descripción generada automáticamente

* **Diccionarios**

Un diccionario en python actua de manera similar a una lista, excepto que el indice de este no necesariamente tiene que ser un numero entero, y se asemejan bastante a un objeto JSON

Tabla

Descripción generada automáticamente