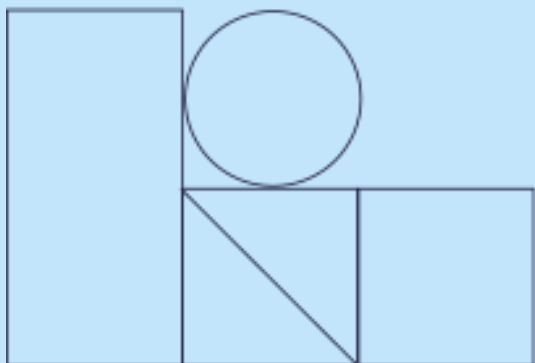


# Conceptos básicos de programación

¿Qué es una estructura de datos?



---

## Índice

Introducción	3
Estructura de una base de datos	4
¿Cuáles son los tipos de datos en informática?	4
¿Cuáles son los tipos de estructura de datos?	5
Arrays	5
Estructura dinámica de datos	5
Estructuras lineales de datos	5
Listas enlazadas	5
Pilas	6
Colas	6
Estructuras no lineales de datos	6

---

# Introducción

Si proponemos una **definición de estructura de datos**, podríamos decir que son las distintas estructuras que se utilizan para representar la información en una computadora. En este sentido, una estructura de datos es una forma de organizar un conjunto de datos elementales, de manera eficiente, con el objetivo de facilitar su uso y manipulación.

Pensemos en un ejemplo de **estructura de datos**, si eres dueño de una librería y estás desarrollando una aplicación para tener un control del stock, con una estructura de datos eficiente podrás tener la información representada, no solo de un libro en particular, sino de todos los libros que tienes en un sistema. De este modo, con las estructuras de datos, puedes manejar grandes cantidades de información de manera eficiente.

Si buscamos un **concepto de estructura de datos** que pertenezca a un manual, veremos que no está muy alejado frente a las que propusimos anteriormente.

“Una estructura de datos es una colección de datos que pueden ser caracterizados por su organización y las operaciones que se definen en ella”.

Para finalizar con esta conceptualización, debemos aclarar que las **estructuras de datos** se basan, generalmente, en la capacidad que tiene la computadora para recuperar y almacenar datos en cualquier lugar de su memoria.

# Estructura de una base de datos

Antes de introducirnos en los distintos tipos de estructura de datos que existen, veamos cuál es la diferencia entre una **estructura de datos** y una estructura de una base de datos.

Como dijimos anteriormente, al plantearnos qué es una **estructura de datos** en programación, especificamos que es una forma de organizar un conjunto de datos con el objetivo de facilitar su manipulación.

En cambio, la **estructura de una base de datos** también es un conjunto de datos, pero con la diferencia de que estos pertenecen a un mismo contexto, siendo almacenados sistemáticamente para su posterior uso. Para entender mejor el concepto veamos un ejemplo.

Piensa en una biblioteca de los años 90. En esa época, para encontrar un libro, uno tenía que buscarlo en unos ficheros que se encontraban en una pared de la biblioteca. Ahí, dentro de unos pequeños estantes, estaban los ficheros ordenados alfabéticamente y por tema. Entonces, anotabas el código y lo llevabas al bibliotecario que, a través de esos números, podía encontrar en qué estante estaba el libro para entregártelo.

Esto que hacían los bibliotecarios, catalogar los libros de forma sistemática para saber dónde buscarlos, es lo que conocemos hoy en día como la **estructura de una base de datos**. ¿Por qué son útiles las estructuras de base de datos? Porque, pensándolo desde la programación, hoy en día se pueden guardar grandes cantidades de información de manera segura, asimismo, agilizar los procesos de búsqueda haciéndolos más eficientes.

Pensándolo desde la programación de **estructura de datos**, podríamos establecer una diferencia más respecto al almacenamiento. La base de datos se almacena en la nube o el disco rígido de la PC en forma de tablas, donde la estructura hace referencia a cómo está compuesta esa tabla y de qué modo está organizada la información.

Ahora que ya conocemos qué es una **estructura de datos** y la diferencia con una estructura de una **base de datos**, veamos los distintos **tipos de datos en informática**.

## ¿Cuáles son los tipos de datos en informática?

Antes de introducirnos en los tipos de estructuras de datos, debemos especificar cuáles son los **tipos de datos en informática** que puede procesar una computadora para después generar una estructura.

Deberás saber que los tipos de datos más frecuentes utilizados en los lenguajes de programación son dos:

- **Simples:** no están compuestos por otra estructura; ocupan solo una casilla de memoria, por lo que una variable simple hace referencia a un único valor a la vez. Entre ellos se encuentran los siguientes valores: enteros, reales y carácter.
- **Compuestos:** que son los datos que con un nombre se hace referencia a un grupo de casillas de memoria; están contruidos en base a los datos simples, un ejemplo puede ser una cadena de caracteres.

Ahora que ya sabes cuáles son los tipos de datos, veamos los distintos **tipos de estructuras de datos**.

## ¿Cuáles son los tipos de estructura de datos?

Los tipos de datos que nombramos anteriormente se pueden organizar mediante dos tipos distintos de estructura:

- **Estructura de datos estática:** “son aquellas en las que el tamaño ocupado en memoria se define antes de que el programa se ejecute y no puede modificarse dicho tamaño durante la ejecución del programa”.
- **Estructura de datos dinámica:** estas no tienen las limitaciones del tamaño de memoria ocupada. “Mediante el uso de un tipo de datos específico, denominado puntero, es posible construir estructuras de datos dinámicas que son soportadas por la mayoría de los lenguajes que ofrecen soluciones eficaces y efectivas en la solución de problemas complejos”.

La elección de una de estas **estructuras de datos** dependerá del tipo de aplicación que se quiera realizar. Dentro de la **estructura de datos estática**, podemos encontrar a los **arrays**.

### Arrays

¿Qué es un **array**? También conocido como arreglo, matriz o vector, un array es un conjunto finito y ordenado de elementos homogéneos. Su condición de ordenado hace que se pueda identificar del primer al último elemento que lo compone y, su condición de homogéneo, indica que sus elementos están compuestos por el mismo tipo de dato.

Pueden llegar a variar según el lenguaje de programación, pero lo importante es que, en cualquier lenguaje, un **arrays** es una estructura de datos que almacena a un conjunto de valores que pertenecen al mismo tipo de datos.

## Estructura dinámica de datos

Las **estructuras dinámicas de datos** son una colección de elementos, llamados **nodos**, que normalmente se utilizan para dejar asentados registros. Al contrario de un **array**, que contiene espacio para almacenar un número fijo de elementos, una **estructura dinámica de datos** se amplía y contrae durante la ejecución del programa.

Se podría decir que se utilizan para el almacenamiento de datos del mundo real que están en constante cambio. Se las puede clasificar en dos grupos distintos: **estructura de datos lineales y no lineales**.

## Estructuras lineales de datos

Las **estructuras lineales de datos** se caracterizan porque sus elementos están ubicados uno al lado del otro relacionados en forma lineal. Cada elemento de la estructura puede estar conformado por uno o varios subelementos o campos que pueden pertenecer a cualquier tipo de dato.

Dentro de las estructuras lineales podemos encontrar tres tipos:

- Listas enlazadas
- Pilas
- Colas

### Listas enlazadas

Estas **listas**, en la **estructura de datos**, se confeccionan con elementos que están ubicados uno tras otro en secuencia, donde cada elemento se conecta con el siguiente a través de un enlace (link) que contiene la posición del siguiente elemento.

Sus operaciones dependen de la necesidad de aplicación y no siguen un mecanismo específico.

Una lista enlazada se define por tres puntos:

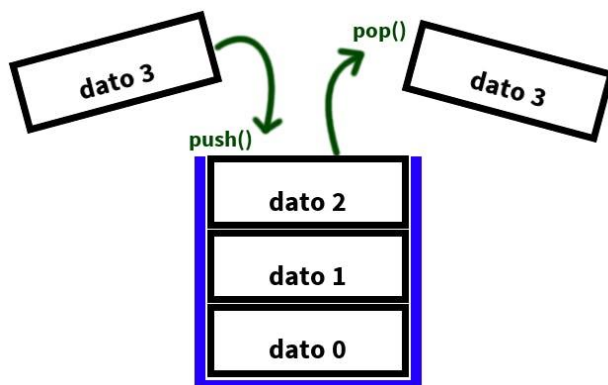
- La información que contienen sus elementos y sus respectivos enlaces.
- El enlace que permite acceder al primer elemento de la lista.
- Un medio para detectar el último elemento de la lista.

## Pilas

La **pila**, en una **estructura de datos**, hace referencia a un tipo especial de lista lineal en la que la inserción y el borrado de nuevos elementos se realiza sólo por un extremo que se denomina cima o tope.

Es un elemento que encuentra varias comparaciones con situaciones de la vida real, como lo es ordenar una pila de libros. Esto se debe a su mecanismo llamado **LIFO** (last-in, first-out, último en entrar, primero en salir), en el cual el último elemento que se pone en la pila es el primero que se puede sacar. Los elementos se incorporan solo por un extremo que, en este caso, sería la cima de la **pila**.

Un **ejemplo de método LIFO** que se da en la vida real, puede ser cuando intentamos apilar libros, ropa o lo que sea, siempre agregamos objetos por encima y si necesitamos sacar alguno, quitamos el que se encuentra más arriba.



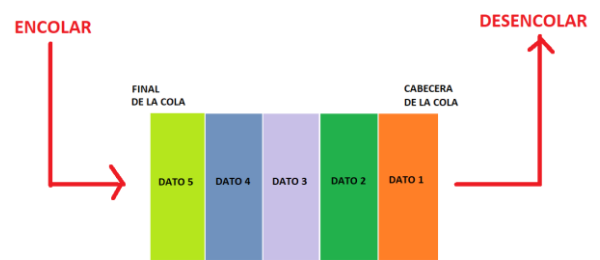
## Colas

¿Qué es una **cola en una estructura de datos**?

Podemos decir que son otro tipo de **estructura lineal** similar a las **pilas**, diferenciándose de ellas en el modo de insertar/eliminar elementos. En este caso, las eliminaciones se realizan por el **inicio de la fila**, en tanto que las inserciones se realizan por el otro extremo, es decir, el final.

Las **colas** en la **estructura de datos** tienen un mecanismo que se conoce como **FIFO** (first-in, first-out - primero en entrar, primero en salir). De este modo, la diferencia con las **pilas** está en el modo en que ingresan y salen los datos. La utilidad de las **colas** yace en almacenar datos que necesitan ser procesados según el orden de llegada.

Aquí también hay varias analogías con la vida real. Pongamos como ejemplo la cola de un supermercado en la que el primer elemento que llega (cliente) es el primero en salir al ser atendido.



## Estructuras no lineales de datos

Las estructuras de datos no lineales también son conocidas por el nombre de *multienlazadas*. En ellas, cada elemento puede estar enlazado a cualquier otro componente. Esto quiere decir que cada elemento puede tener varios sucesores o varios predecesores.

Las estructuras no lineales de datos son dos:

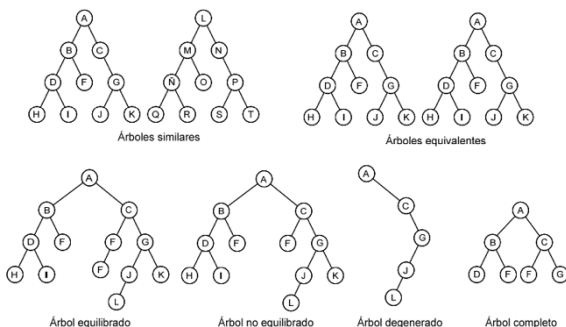
- Árboles
- Grafos

## Árboles

Los **árboles**, en la **estructura de datos**, son 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.

Se puede realizar una clasificación de árboles en estructura de datos:

- **Árbol general:** que es un árbol donde cada elemento puede tener un número ilimitado de sub árboles.
- **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. Dentro de ellos podemos encontrar:
  - **Distintos:** posee una estructura distinta a la de otros árboles.
  - **Similares:** cuando su estructura es idéntica a la de otros árboles binarios, pero la información que guardan los nodos es diferente entre sí.
  - **Equivalentes:** cuando su estructura e información de sus nodos es idéntica a la de otros árboles binarios.
  - **Equilibrados:** las alturas de los dos subárboles de cada uno de los nodos tienen como máximo una diferencia de una unidad.
  - **Degenerados:** sus nodos tienen solo un sub árbol.



## Grafos

Los **grafos**, en las **estructura de datos**, son otra **estructura no lineal**, así como los árboles. **Aguilar** los define de la siguiente forma:

“Formalmente un grafo es 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”.

Si nos preguntamos ¿qué es un **grafo** y para qué sirve? Diremos que es una estructura matemática que permite modelar problemas de la vida cotidiana, mediante una representación gráfica, formada por **nodos**, que muestra las relaciones que hay entre los distintos componentes.

Podemos poner el ejemplo de una red social donde se establecen relaciones entre las personas que a su vez generan relaciones entre ellas interactuando, conformando así un grafo.

