



# Definición y tipos de estructuras de datos

Bienvenidos a este módulo sobre estructuras de datos y algoritmos. Aprenderemos qué son las estructuras de datos, cómo se clasifican, y los conceptos clave relacionados con su diseño y análisis.

# ¿Qué es una estructura de datos?

## Definición

Una estructura de datos es una forma de organizar y almacenar datos en la memoria de una computadora, con el fin de facilitar su acceso y manipulación.

## Características

Las estructuras de datos tienen propiedades como tamaño, forma y relaciones entre elementos que influyen en cómo se pueden utilizar.

## Objetivos

Su objetivo es permitir un procesamiento eficiente de la información, optimizando el uso de recursos como tiempo y espacio.

# Tipos básicos de estructuras de datos

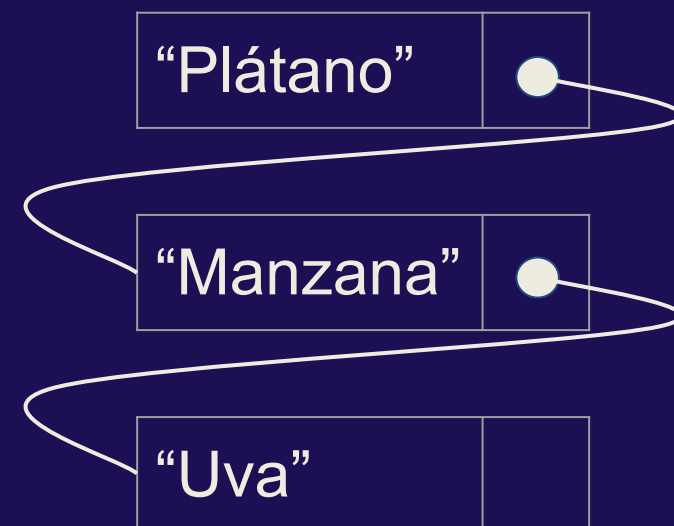
## Arrays

Colección de elementos del mismo tipo, almacenados en posiciones contiguas de la memoria.

0	"Plátano"
1	"Manzana"
2	"Pera"
3	"Uva"
4	"Fresa"

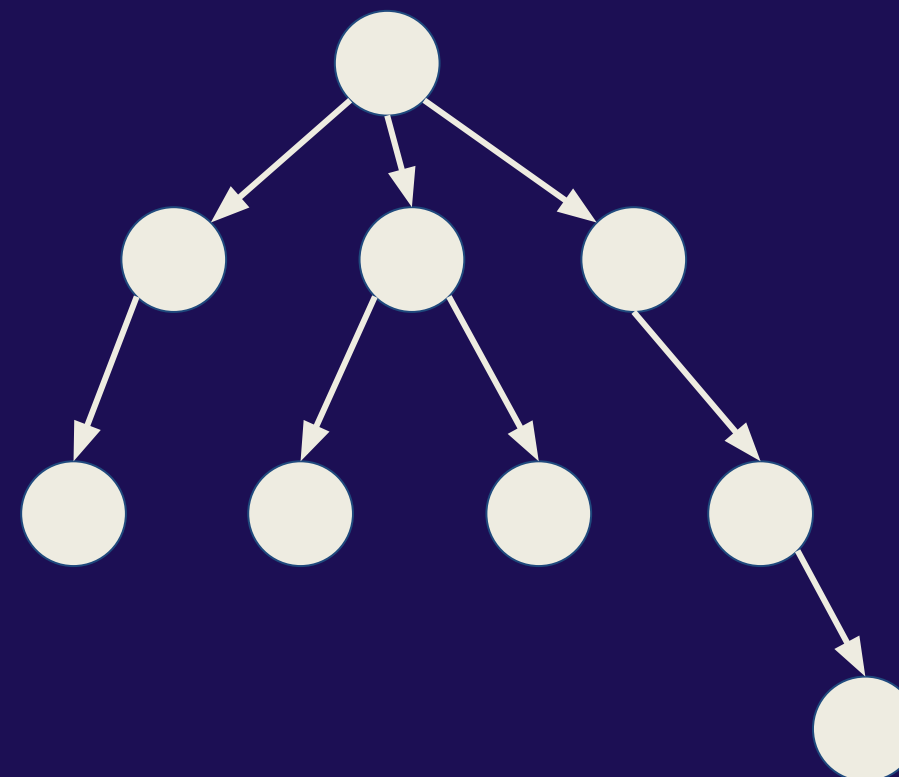
## Listas Enlazadas

Secuencia de nodos, cada uno con un valor y un puntero al siguiente nodo.



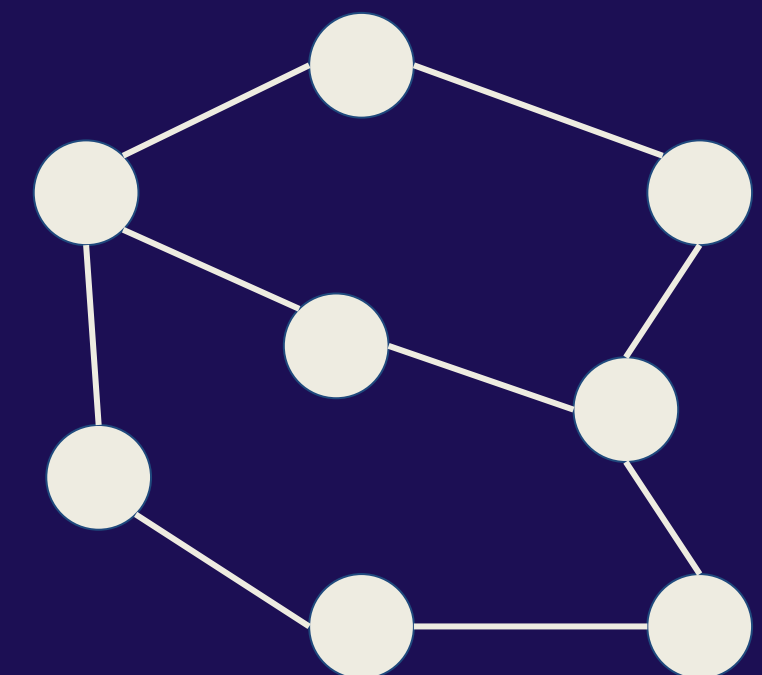
## Árboles

Estructura jerárquica con un nodo raíz y subárboles, utilizados para búsqueda y organización.



## Grafos

Conjunto de nodos (vértices) conectados por enlaces (aristas), útiles para representar relaciones.



# ¿Qué es un algoritmo?

## 1 Definición

Un algoritmo es un conjunto de instrucciones o pasos bien definidos y ordenados que resuelven un problema o realizan una tarea específica.

## 2 Importancia

Los algoritmos son fundamentales en la programación, ya que permiten automatizar procesos y solucionar problemas de manera eficiente.

## 3 Características

Un algoritmo debe ser preciso, finito, efectivo y genérico para poder ser aplicado a diferentes situaciones.

# Órdenes de complejidad

## Definición

La complejidad de un algoritmo se refiere al tiempo y espacio necesarios para ejecutarlo, en función del tamaño de la entrada.

## Big O Notation

La notación Big O permite clasificar los algoritmos según su complejidad, como constante ( $O(1)$ ), lineal ( $O(n)$ ), logarítmica ( $O(\log n)$ ), etc.

## Importancia

Conocer la complejidad de un algoritmo es crucial para diseñar soluciones eficientes y predecir el rendimiento de un programa.



# Tipos básicos de algoritmos

1

## Búsqueda

Algoritmos para encontrar un elemento dentro de una estructura de datos, como búsqueda secuencial o binaria.

2

## Ordenación

Algoritmos para organizar los elementos de una estructura de datos en un orden específico, como quicksort o mergesort.

3

## Camino mínimo

Búsqueda del recorrido de menos coste en un grafo desde el punto A al punto B.





# Hashing

## Definición

El hashing es una técnica para asignar datos a una estructura de almacenamiento mediante una función hash, que transforma la información en un índice.

## Aplicaciones

Se utiliza en tablas hash, caches, índices de bases de datos y otros sistemas que requieren acceso rápido a los datos.

## Ventajas

El hashing permite realizar operaciones como búsqueda, inserción y eliminación de datos de manera eficiente.



# Ejemplos de código

1

## Implementación de una Pila

Una pila es una estructura de datos que sigue el principio LIFO (Last In, First Out), donde los elementos se van apilando y el último en entrar es el primero en salir.

2

## Búsqueda Binaria

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada. Divide repetidamente el espacio de búsqueda por la mitad hasta encontrar el elemento.

3

## Ordenamiento Quicksort

El algoritmo Quicksort es un método de ordenamiento basado en dividir y conquistar. Selecciona un elemento como pivote y particiona la lista en dos sublistas, antes y después del pivote.

