



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



**FIME**

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Materia: Estructura de Datos

**Fase 1**

Actividad 1

Maestro: NOEL ALEJANDRO HORTIALES CORONA

Alumnos: Dante Rodríguez Rodríguez 1899372

Castillo López Alfredo Ventura 2000505 IAS

Sánchez Alvarado Fabritzio Javier 2059273 IAS

Loredo Galindo Esteban Alessandro 2052205 ITS

Las estructuras de datos son formas de organizar y almacenar datos en un programa de computadora, permitiendo el acceso y la manipulación de estos. Estas estructuras son fundamentales en la programación, ya que permiten la optimización de los algoritmos y la gestión eficiente de la información.

## **Algoritmos**

Los algoritmos son una serie de instrucciones que permiten resolver un problema específico en un tiempo determinado. Estos son fundamentales en la programación, ya que permiten la resolución de problemas complejos de forma eficiente y sistemática.

Un algoritmo es una secuencia finita de pasos que, dada una entrada particular, resuelve un problema o realiza una tarea específica.

Un ejemplo de algoritmo simple es el de la receta de cocina. La receta proporciona una serie de pasos precisos que, al seguirlos, permiten cocinar un plato específico de manera consistente.

## **Análisis de algoritmos**

El análisis de algoritmos se refiere al estudio de la eficiencia de los algoritmos, considerando el tiempo y el espacio necesarios para su ejecución. Este análisis permite la optimización de los algoritmos para reducir su complejidad y mejorar su rendimiento.

## **Notación Big O**

La notación Big O es una forma de expresar la complejidad de un algoritmo en términos de la cantidad de operaciones que realiza en relación con el tamaño de los datos de entrada.

La notación Big O se utiliza para expresar la complejidad asintótica de un algoritmo, indicando su comportamiento en el peor caso.

## ***¿En qué consiste?***

La notación Big O consiste en identificar la función que describe la complejidad del algoritmo en términos del tamaño de los datos de entrada, y luego eliminar los términos de menor orden y las constantes.

La notación Big O se utiliza para comparar la complejidad de diferentes algoritmos y para optimizar el rendimiento de estos.

## ***Ejemplos de comportamientos***

Un ejemplo de comportamiento de la notación Big O es  $O(1)$ , que indica una complejidad constante en tiempo y espacio, y  $O(n^2)$ , que indica una complejidad cuadrática en tiempo y espacio.

## **Diagramas de flujo**

Los diagramas de flujo son una representación gráfica de los pasos necesarios para resolver un problema mediante un algoritmo.

Un diagrama de flujo es una representación gráfica de un algoritmo que utiliza símbolos y flechas para indicar los pasos necesarios para resolver un problema.

Los bloques en un diagrama de flujo representan una acción específica que se realiza en el algoritmo.

Un ejemplo de bloque en un diagrama de flujo es el bloque de entrada, que representa la acción de recibir datos del usuario.

## Estructuras lineales

Las estructuras lineales son aquellas en las que los datos se organizan en una secuencia lineal

### Pilas

Una pila es que es una estructura de datos lineal que permite la inserción y eliminación de elementos en una sola dirección, siguiendo el principio "último en entrar, primero en salir" (LIFO, por sus siglas en inglés).

En una pila, los elementos se insertan y eliminan siempre por un extremo llamado "cima" o "top". Las operaciones básicas que se pueden realizar en una pila son:

**Push:** Insertar un elemento en la cima de la pila.

**Pop:** Eliminar el elemento que se encuentra en la cima de la pila.

**Peek:** Obtener el elemento que se encuentra en la cima de la pila sin eliminarlo.

El funcionamiento de una pila se basa en dos operaciones principales: push (insertar) y pop (eliminar). Los elementos se insertan en la parte superior de la pila y se eliminan desde la misma posición. La pila sigue el principio LIFO (Last In, First Out), lo que significa que el último elemento en ser insertado será el primero en ser eliminado.

Además de las operaciones básicas de push y pop, también se pueden realizar otras operaciones con las pilas, como peek (para ver el elemento superior sin eliminarlo), size (para obtener el tamaño actual de la pila) y isEmpty (para comprobar si la pila está vacía).

## **Colas**

Las colas son estructuras de datos lineales que permiten la inserción de elementos en una extremidad y la eliminación de elementos en la otra extremidad

Una cola es una estructura de datos lineal que permite la inserción de elementos en una extremidad y la eliminación de elementos en la otra extremidad, siguiendo el principio "primero en entrar, primero en salir" (FIFO, First In, First Out).

El funcionamiento de una cola se basa en dos operaciones principales: enqueue (insertar) y dequeue (eliminar). Los elementos se insertan en una extremidad de la cola y se eliminan desde la otra extremidad. La cola sigue el principio FIFO, lo que significa que el primer elemento en ser insertado será el primero en ser eliminado.

Además de las operaciones básicas de enqueue y dequeue, también se pueden realizar otras operaciones con las colas, como peek (para ver el elemento frontal sin eliminarlo), size (para obtener el tamaño actual de la cola) y isEmpty (para comprobar si la cola está vacía).

## **Arreglos**

Los arreglos son estructuras de datos que permiten el almacenamiento de una colección de elementos del mismo tipo de datos en una secuencia lineal.

Un arreglo es una estructura de datos que permite el almacenamiento de una colección de elementos del mismo tipo de datos en una secuencia lineal, donde cada elemento es identificado por un índice.

El funcionamiento de un arreglo se basa en la indexación de los elementos, donde cada elemento se identifica por su posición en la secuencia. Los elementos se pueden insertar y eliminar en cualquier posición del arreglo, pero esto puede requerir una reorganización de los elementos existentes.

Entre las operaciones que se pueden realizar con los arreglos se encuentran la inserción, eliminación y búsqueda de elementos en una posición específica, así como también la obtención del tamaño total del arreglo.

### ***Listas enlazadas***

Las listas enlazadas son estructuras de datos que permiten el almacenamiento de una secuencia de elementos mediante una serie de nodos enlazados entre sí.

Una lista enlazada es una estructura de datos que permite el almacenamiento de una secuencia de elementos mediante una serie de nodos enlazados entre sí, donde cada nodo contiene un elemento y una referencia al siguiente nodo en la secuencia.

El funcionamiento de una lista enlazada se basa en la manipulación de las referencias entre los nodos, lo que permite la inserción y eliminación de elementos en cualquier posición de la lista. Cada nodo tiene una referencia al siguiente nodo en la secuencia, excepto por el último nodo que tiene una referencia nula. Las listas enlazadas también pueden ser simples, dobles o circulares, dependiendo de si cada nodo tiene una o dos referencias.

Entre las operaciones que se pueden realizar con las listas enlazadas se encuentran la inserción, eliminación y búsqueda de elementos en una posición específica, así como también la obtención del tamaño total de la lista. Además, se pueden implementar otras operaciones como la concatenación de dos listas enlazadas, la inversión de una lista y la eliminación de elementos duplicados.

## Referencias

- "Data Structures and Algorithms in Python" de Michael T. Goodrich, Roberto Tamassia y Michael H. Goldwasser.
  - "Estructuras de datos en Java" de Mark Allen Weiss.
  - "Algorithms" de Robert Sedgewick y Kevin Wayne.
- "Introduction to Algorithms" de Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest y Clifford Stein.
  - "The Algorithm Design Manual" de Steven S. Skiena.
- "The Art of Computer Programming, Volume 1: Fundamental Algorithms" de Donald E. Knuth.
- "Flow-Based Programming: A New Approach To Application Development" de J. Paul Morrison.
- "Flowcharting and Basic" de Raymond C. Wickenden.