



1INF27

Algoritmia y Estructura de Datos

2024

Profesores:

Cueva, R. | Allasi, D. | Roncal, A. | Huamán, F.

0581

0582

0583

0584

OBJETIVOS



Resultados de Aprendizaje:

- Analizar el procesamiento de listas y algunas técnicas de ordenación.
- Definir el TAD para las listas.
- Indicar cómo puede utilizarse una lista para resolver determinados problemas.
- Analizar las diversas implementaciones con listas.



Capítulo 2

LISTAS





Imagen: Flaticon.com

LISTAS

Definición

Una lista es un tipo de estructura **lineal** y **dinámica** de datos. Lineal porque a cada elemento le puede seguir solo otro elemento; dinámica porque se puede manejar la memoria de manera flexible, sin necesidad de reservar espacio con antelación.

Ventajas: se pueden adquirir posiciones de memoria a medida que se necesitan; éstas se liberan cuando ya no se requieren.





Imagen: Flaticon.com

LISTAS

Definición

Una lista en el contexto de Algoritmia y Estructura de Datos se refiere a una estructura de datos que organiza elementos de manera secuencial, permitiendo el acceso y la manipulación eficiente de dichos elementos.

En el mundo real, las listas se utilizan para representar colecciones de elementos relacionados entre sí. Algunos ejemplos comunes de listas en situaciones cotidianas incluyen:





LISTAS

Ejemplos de listas

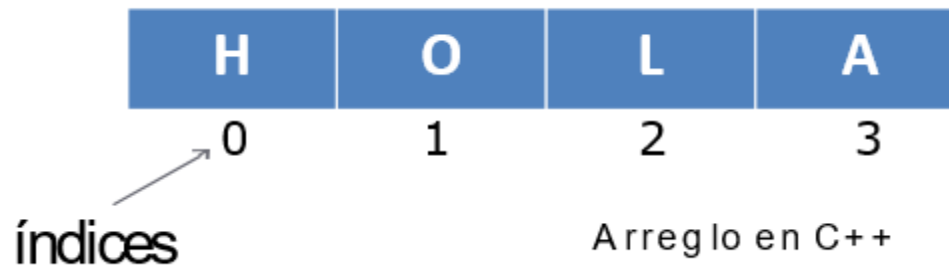
- **Lista de canales de televisión:** Los canales se enumeran secuencialmente en una lista, lo que permite a los espectadores cambiar entre ellos fácilmente.
- **Lista de alumnos de la clase:** En una clase, los alumnos se pueden organizar en una lista secuencial, facilitando tareas como la toma de asistencia, la asignación de tareas o la calificación.
- **Lista de espera para una operación:** En un hospital, por ejemplo, los pacientes que esperan una cirugía pueden organizarse en una lista según la prioridad de sus necesidades médicas.





Listas estáticas

- Una forma simple de implementar una lista estática es mediante arreglos. Dónde:



- El tamaño de los arreglos tiene que ser conocido en el tiempo de compilación.
- Los datos en el arreglo se encuentran organizados de forma lineal.
- Esto significa que insertar un elemento dentro del arreglo requiere mover los demás datos del arreglo.





LISTAS

Preguntas

¿Puede haber huecos en la lista?

¿Puede haber elementos repetidos en una lista?



LISTAS - Resumen

- Una lista es una secuencia ordenada de elementos del mismo tipo, sobre la que podemos insertar, borrar, consultar o buscar elementos.
- Las listas son una herramienta fundamental en la programación y la ciencia de la computación, y su comprensión es esencial para el desarrollo de software eficiente y escalable.



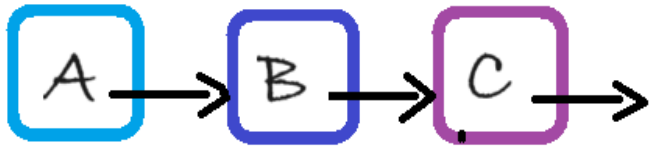
Listas simplemente enlazadas



Listas dinámicas

Implementar una lista enlazada es más complejo que usando arreglos pero, al ser dinámica, nos proporciona mucha mas flexibilidad.

Cada nodo de la lista contendrá un valor, que será el tipo de dato que se requiera almacenar en la lista y un puntero hacia el siguiente elemento, al siguiente nodo de la lista.

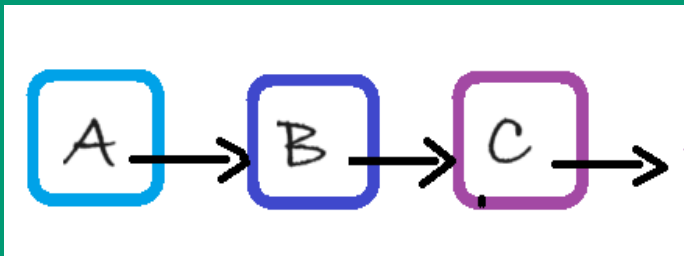


LISTA ENLAZADA

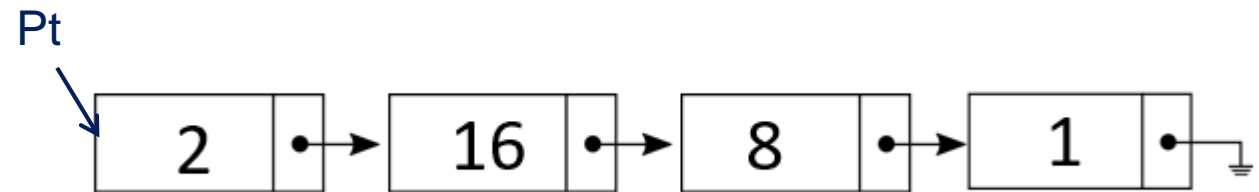


Lista enlazada

Una **lista enlazada** constituye una colección de elementos llamados nodos. El orden entre éstos se establece por medio de punteros; es decir, direcciones o referencias a otros nodos. Por ejemplo:



LISTA ENLAZADA



Un tipo especial de lista enlazada es una lista vacía.



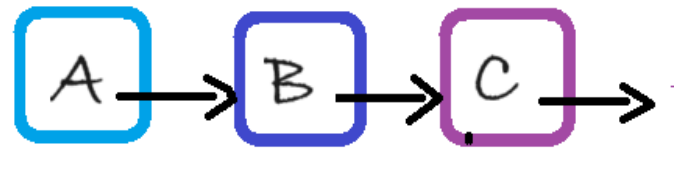
Estructura básica del Nodo

```
struct Nodo {  
    Tipo elemento;  
    struct Nodo * siguiente;  
};
```

Donde:

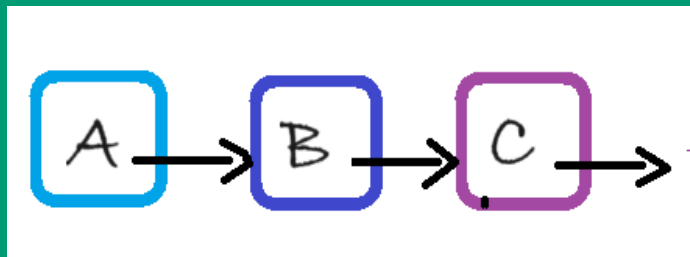
elemento, es cualquier tipo de datos sin distinción, dependerá de lo que se quiera guardar en el conjunto **double**, **int**, **char**, **struct**, ect.

Siguiente, puntero a una variable de tipo **struct Nodo**.



LISTA ENLAZADA





LISTA ENLAZADA

Estructura básica de la Lista

```
#include "Nodo.h"
```

```
struct Lista {
```

```
    struct Nodo * cabeza;
```

```
    int longitud;
```

```
};
```

Donde:

cabeza, es un puntero a una variable de tipo **struct** Nodo.

longitud, es el tamaño de la lista.



Operaciones sobre listas

- **Crear** la lista, prepara el espacio de memoria para albergar la lista.
- **Esta vacía** la lista, consulta si la lista está con o sin datos.
- **Insertar** un elemento a la lista, añade un elemento a la lista.
- **Eliminar** un elemento de la lista, borra un elemento de la lista.
- **Tamaño** de la lista, devuelve el número de elementos que hay en la lista.
- **Destruir** lista, deja la lista vacía.
- **Imprimir** lista, muestra los elementos de la lista.



Ejercicios



- Implementar una función que invierta los elementos de una lista enlazada.
- Escriba una función que rote una lista enlazada una posición a la izquierda (no usar nuevos nodos).
- Escribir una función para determinar si una lista enlazada de caracteres contiene un palíndromo.
- Escribir una función que intercambie pares de elementos de una lista enlazada simple.
- Escriba un programa que elimine un elemento repetido de una lista, los repetidos se encuentran adyacentes en la lista.
- Escriba un programa que “complete” la lista, de tal manera que la misma, una vez modificada, alcance todos los valores a partir del número del primer nodo hasta el número del último.



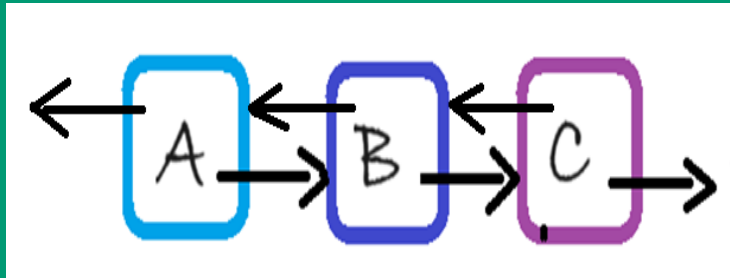
LISTAS ENLAZADAS - Resumen

- Una **lista enlazada** es una estructura de datos en la que los objetos se organizan en un **orden lineal**.
- A diferencia de un arreglo, en donde el orden lineal está determinado por sus índices, el orden en una lista enlazada está determinado por un puntero en cada objeto.
- Las listas enlazadas brindan una representación simple y flexible para conjuntos dinámicos, que admiten una serie de operaciones típicas.
- Existen varias formas de implementar listas enlazadas.



Listas doblemente enlazadas

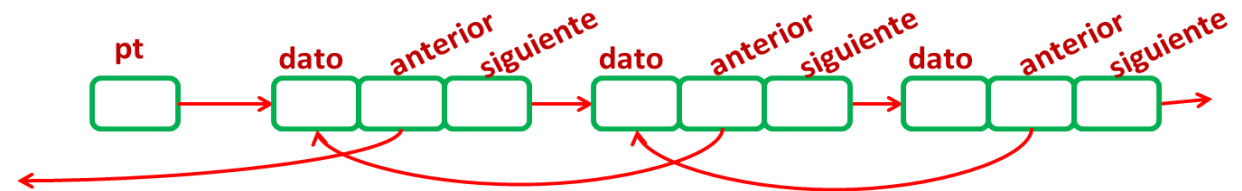


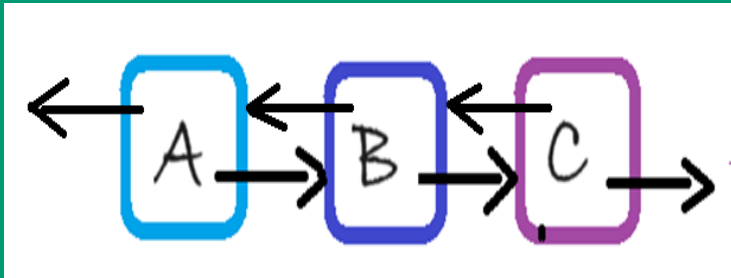


LISTA DOBLEMENTE ENLAZADA

Definición

Una **lista doblemente enlazada** es una colección de nodos. Donde cada uno de ellos tiene dos apuntadores, uno apunta a su **predecesor** y otro a su **sucesor**.





LISTA DOBLEMENTE ENLAZADA

Estructura básica del Nodo

```
struct Nodo {  
    elemento elemento;  
    struct Nodo * anterior;  
    struct Nodo * siguiente;  
};
```

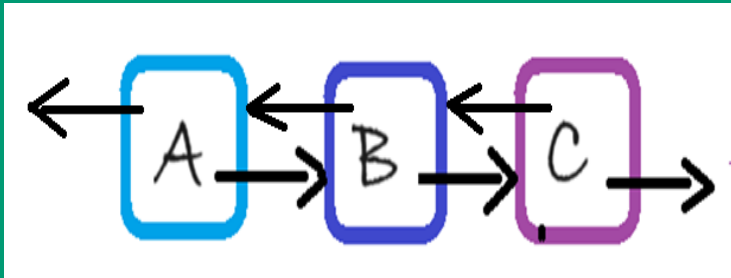
Donde:

Elemento, es cualquier tipo de datos sin distinción.

Anterior, puntero a su predecesor.

Siguiente, puntero a su sucesor.





LISTA DOBLEMENTE ENLAZADA

Estructura básica de la lista

```
#include "Nodo.h"
```

```
struct Lista {
```

```
    struct Nodo * cabeza;
```

```
    struct Nodo * cola;
```

```
    int longitud;
```

```
};
```

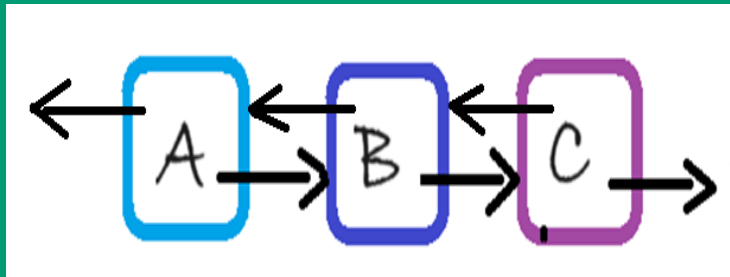
Donde:

Cabeza, es un puntero al inicio de la lista.

Cola, es un puntero al final de la lista, se recomienda usar para tener un fácil acceso a la información de la lista.

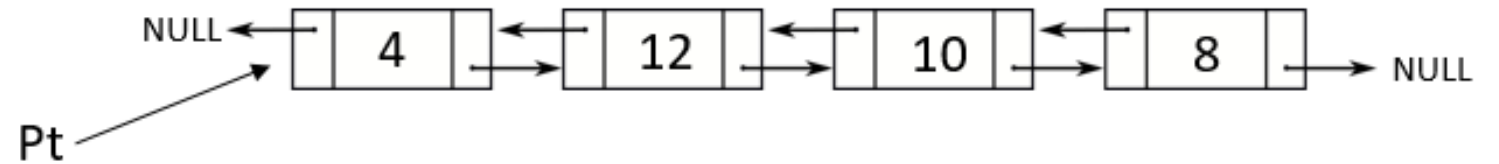
Longitud, es el tamaño de la lista.





LISTA DOBLEMENTE ENLAZADA

Ejemplo:



Ejercicio

Torneo de equipos

Considere los ocho equipos de bolos con sus respectivas victorias para crear los emparejamientos de **la primera ronda del torneo**, los equipos se almacenarán en una lista ordenada según el número de victorias. El primer equipo se extrae de la lista y se empareja con el último de la misma. El proceso se repite emparejando los siguientes equipos. Este proceso continua hasta que la lista está vacía.

Nombre del equipo	Victorias
GutterBalls	9
KingPins	8
PinDoctors	7
Scorecards	10
Spares	5
Splits	4
Tenpins	3
Woodsplitters	2



LISTAS DOBLEMENTE ENLAZADAS - Resumen

- La ventaja de las **listas doblemente enlazadas** es que se puede recorrer la lista en ambos sentidos usando los apuntadores.
- Las operaciones que se pueden realizar con listas doblemente enlazadas son las mismas que con listas simplemente enlazadas.



Listas circulares enlazadas



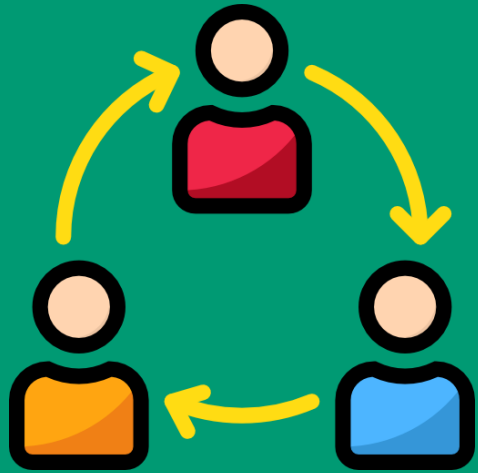
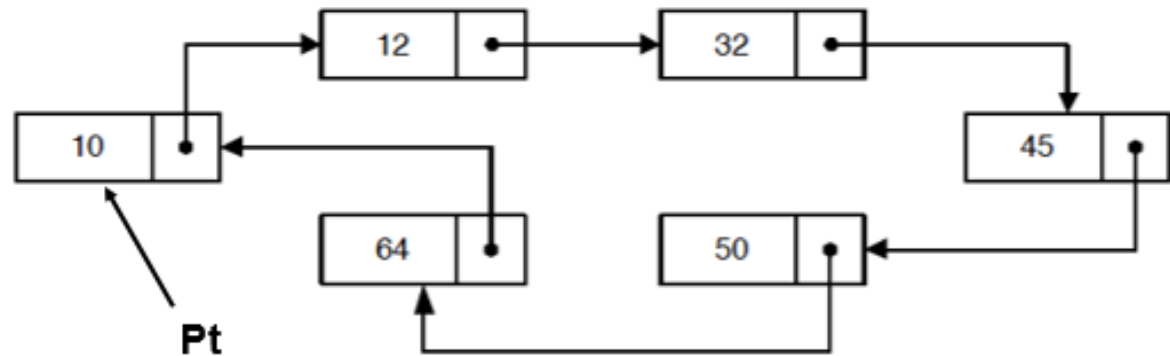


Imagen: Flaticon.com

LISTAS CIRCULARES ENLAZADAS

Definición

Una **lista circular enlazada** es una lista enlazada en donde los nodos forman un **anillo** y cada nodo tiene un único **sucesor**.



Ejercicio

¡Sorteo en ronda al estilo de Josefo!

Los participantes se reúnen en una ronda y compiten por ser el último en pie para ganar el premio. Explicación del juego:

Formación de la Ronda: Todos los participantes se colocan en una formación circular.

Elección del Ganador: Se elige un número clave, llamémoslo "k", que determinará el patrón de eliminación. Comenzando desde el primer participante en la ronda, se eliminarán secuencialmente aquellos cuya posición coincide con un múltiplo de "k". Cuando llega el turno de un participante cuya posición es un múltiplo de "k", este es eliminado de la ronda. El proceso continúa, pasando de un participante al siguiente en sentido horario, eliminando a los participantes según el patrón establecido por el número "k".

El Último en Pie: El juego continúa hasta que solo queda un participante en la ronda. Ese individuo es declarado el ganador del sorteo en ronda al estilo de Josefo y se lleva el premio.



LISTAS CIRCULARES ENLAZADAS - Resumen

- Las **listas circulares** permiten la navegación en cualquier sentido a través de la misma, se puede recorrer toda la lista partiendo de cualquier nodo siempre que se tenga un apuntador a este. Se suele utilizar un nodo de cabecera.



Bibliografía

Cueva, R. (2022, Agosto 21). Capítulo 2. *Estructuras de Datos (TAD) y sus implementaciones*. [PowerPoint slides]. Facultad de Ciencias e Ingeniería. PUCP.

Cairó, O., & Guardati, S. (2006). *Estructura de datos*. Graw Hill. México, D. F.

Hernández, R., Lázaro, J. C., Dormido, R. y Ros, S. (2001). *Estructuras de Datos y Algoritmos*. Pearson Education, S.A.

Nadal, M. (2022). *Estructuras de Datos y Algoritmos*. Grupo ANAYA, S.A. Madrid.

Lewis, J., & Chase, J. (2006). *Estructuras de datos con Java*. PEARSON EDUCACIÓN S.A. Madrid.