



# Listas

## Laboratorio Estructuras de Datos

Las listas son estructuras que cuentan con operaciones como agregar, eliminar, obtener un elemento, vaciar, etc.

Hay varios tipos de listas y dependiendo de sus características pueden tener algunas ventajas o desventajas sobre otros tipos de listas.

A continuación encontraras algunos tipos de listas:

- **Lista Simple** : Es aquella que sólo cuenta con la referencia al nodo *siguiente*.
- **Lista Doblemente Ligada**: En este tipo de lista tenemos las referencias a los nodos *anterior* y *siguiente*.
- **Lista Circular** : En ella el último nodo apunta al primero, esta puede ser doblemente ligada o simplemente ligada dependiendo de la implementación.

**Ejemplo:** Imagina que tuvieras que ir a comprar el mandado si tienes una buena memoria podrías grabarte lo que tienes que comprar mentalmente pero si no es el caso lo más común es hacer una lista, la lista es un tipo de estructura que es muy fácil de usar y de entender, la ventaja de una lista es que puedes acceder a cualquiera de los elementos a lo largo de ella y a diferencia de una pila o cola (estructuras que tienen operaciones de agregar o eliminar que respetan un orden asignado) la lista puede ser modificada de forma diferente a como se fue creando es decir puedes eliminar cualquier elemento en su interior sin importar si es el primero o el último.



Figura 1: Lista de Súper: De la misma forma que en la imagen en una lista puedes eliminar los elementos en un orden diferente al que se agregaron.

En java podemos crear una Lista de la siguiente forma.

- 
- Listas usando la interfaz `List` y la clase `ArrayList`

```
List<String>pila = new ArrayList<>() // Lista de cadenas  
List<Integer>pila = new ArrayList<>() // Lista de enteros
```

- Listas usando la interfaz `List` y la clase `LinkedList`

```
List<String>pila = new LinkedList<>() // Lista de cadenas  
List<Integer>pila = new LinkedList<>() // Lista de enteros
```

## Métodos para manejar Listas

- *add(E item)*: Este método sirve para guardar un elemento al final de la lista.
- *remove(int i)* Este método nos permite eliminar el elemento en la posición *i*.
- *get(int i)* Este método nos permite ver el elemento en la posición *i*.