

Colecciones



Listas



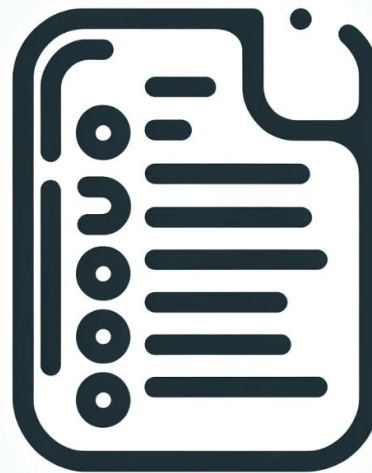
Listas

Definición: Colección de datos **ordenada**.

Propiedades interesantes:

- Mantiene el **orden** de inserción.
- **Acceso directo** a elementos por el índice.
- Permite la inserción de valores **duplicados y nulos**.
- **Iterable**.

ArrayList, Vector, Stack, LinkedList



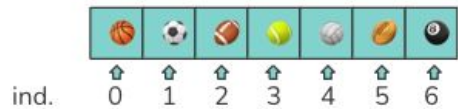


Clase: ArrayList

Definición: Array de tamaño variable

Propiedades interesantes:

- Array de tamaño “dinámico” 🎉
- Rápidos en la obtención y modificación de datos
- Lento en la inserción o eliminación de datos





ArrayList: Métodos

- **Añadir elementos**
 - void add(int posición, E elemento)
 - boolean add(E elemento)
 - boolean addAll(int posición, Collection<? extends E> c2)
- **Recuperar elemento en una posición**
 - E get(int posición)
- **Devuelve el (primer o último) índice de un elemento**
 - int indexOf(Object o)
 - int lastIndexOf(Object o)
- **Reemplaza elementos en una posición**
 - E set(int posición, E elemento)
- **Ordena los elementos de un array**
 - sort(Comparator<? super E> c)
- **Elimina los elementos situados en una posición**
 - E remove(int posición)

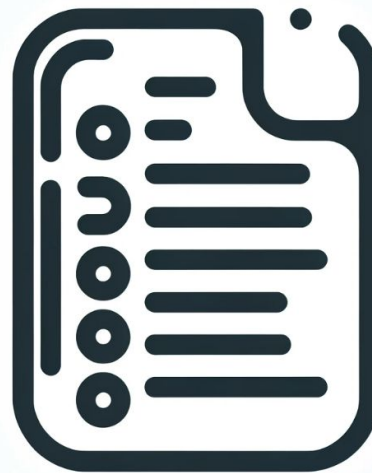
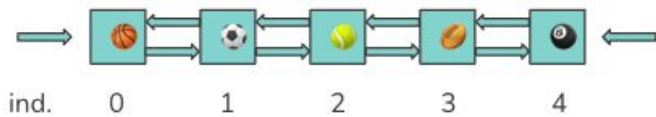


LinkedList

Definición: Implementación de lista con nodos de doble enlace.

Propiedades interesantes:

- ~~Array~~ de tamaño dinámico 🎉🎉
- Rápidos en la inserción o eliminación de datos cabeza/cola
- Lentos en el acceso directo / modificación intermedios
- Puede actuar como una Cola/Pila





LinkedList: Métodos

- **Inserta elementos**
 - `addFirst(E e)`
 - `addLast(E e)`
- **Devuelve elementos**
 - `E getFirst()`
 - `E getLast()`
- **Elimina elementos**
 - `E removeFirst()`
 - `E removeLast()`
- **Recorre descendientemente**
 - `Iterator<E> descendingIterator()`
- **Elimina la primera/última ocurrencia**
 - `boolean removeFirstOccurrence(Object o)`
 - `boolean removeLastOccurrence(Object o)`