

# LISTAS/ARRAYS



# COLLECTIONS JAVA

## Operaciones Basicas

### AGREGAR ELEMENTOS

Cada vez que se agregar un elemento, este se añade al final de la lista

```
lista.add(elemento);
```

Tambien se puede agregar un nuevo elemento en una posición especifica

```
lista.add(indice, elemento);
```

### OBTENER ELEMENTOS

```
Tipo elemento = lista.get(indice);
```

### VERIFICA SI EXISTE UN ELEMENTO

```
boolean existe = lista.contains(elemento);
```

### ENCONTRAR UNA POSICION

```
int posicion = lista.indexOf(elemento);
```

### ELIMINAR UN ELEMENTO

```
lista.remove(indice);
```

### RECORRER LA LISTA

```
for (Tipo elemento : lista) {  
    System.out.println(elemento);  
}
```

### REEMPLAZAR UN ELEMENTO

```
lista.set(indice, nuevoElemento);
```

### OTRA DEFINICION

LIST <>: interfaz (que)

ARRAYLIST<>: implementacion de la interfaz (como)

## QUE ES?

### COLLECTIONS

Una colección es una estructura de datos que permite almacenar y organizar múltiples elementos. Un lugar donde puedes guardar muchos objetos (datos). Estos permiten operaciones como buscar, eliminar y ordenar.



## List

Una list en java es un conjunto de datos, en el que cada elemento va a tener su posicion

Instancia:

```
List<Tipo> nombre = new ArrayList<>();
```

## ARRAYLIST

Es una clase que permite almacenar elementos y ordenarlos segun como fueron insertados. Sin embargo, tiene una manipulacion lenta (recorre todo el arraylist para hacer cambios)

