

# Nota Extra ArrayList

## Estructuras de datos

Hemos visto los array pero hay muchos otros tipos (vectores, listas, listas simples, etc...)

Los arrays y listas tienen complejidad  $n$

Los metodos de ordenar de burbuja o inserccion son de complejidad  $n^2$

---

## ArrayList

El array puede ser de 1 dimensión o de muchas dimensiones mientras que el `ArrayList` solo puede ser de `1`.

Todas las operaciones de una `ArrayList` se ejecutan por métodos debido a que es una *Clase Java*.

### Size

```
a.size()
```

Es dinámico y puede ser incrementado o reducido cuando se necesite.

El `ArrayList` puede almacenar objetos.

## Crear ArrayList

Hay que importarlos

```
import java.util.ArrayList;
```

Luego crearlo como un objeto

```
ArrayList<String> cars = new ArrayList<String>();
```

## Añadir items

```
cars.add("Volvo");  
cars.add("BMW");
```

```
cars.add("Ford");  
cars.add("Mazda");
```

## Acceder a items

Hay que utilizar el método `get()`.

```
cars.get(0);
```

## Cambiar un item

Hay que utilizar el método `set()`.

```
cars.set(0, "Opel");
```

## Eliminar un item

Hay que utilizar el método `remove()`.

```
cars.remove(0);
```

**CUIDADO!** Cuando un elemento se elimina la ArrayList se reestructura (se reduce el tamaño del array).

Para eliminar todos lo items puedes usar el método `clear()`.

## Size

Sirve para ver cuantos elementos hay en la ArrayList

Hay que utilizar `size()`

## Recorrer una ArrayList

Hazlo con un bucle `for` y usa `size()` para especificar cuantas veces tiene que iterar el bucle.

```
public class Main {  
    public static void main(String[] args) {  
        ArrayList<String> cars = new ArrayList<String>();  
        cars.add("Volvo");  
        cars.add("BMW");  
        cars.add("Ford");  
        cars.add("Mazda");  
        for (int i = 0; i < cars.size(); i++) {  
            System.out.println(cars.get(i));  
        }  
    }  
}
```

```
}  
}
```

## Ordenar el ArrayList

El ejemplo de Internet usa la librería Collections que incluye un `short()` para palabras

```
import java.util.ArrayList;  
import java.util.Collections; // Import the Collections class  
  
public class Main {  
    public static void main(String[] args) {  
        ArrayList<String> cars = new ArrayList<String>();  
        cars.add("Volvo");  
        cars.add("BMW");  
        cars.add("Ford");  
        cars.add("Mazda");  
        Collections.sort(cars); // Sort cars  
        for (String i : cars) {  
            System.out.println(i);  
        }  
    }  
}
```

Para números:

```
import java.util.ArrayList;  
import java.util.Collections; // Import the Collections class  
  
public class Main {  
    public static void main(String[] args) {  
        ArrayList<Integer> myNumbers = new ArrayList<Integer>();  
        myNumbers.add(33);  
        myNumbers.add(15);  
        myNumbers.add(20);  
        myNumbers.add(34);  
        myNumbers.add(8);  
        myNumbers.add(12);  
  
        Collections.sort(myNumbers); // Sort myNumbers  
  
        for (int i : myNumbers) {  
            System.out.println(i);  
        }  
    }  
}
```

## Clonar el ArrayList

Hay que usar el método `clone()`.

```
arraylist.clone()
```

## Comprobar si un elemento esta en el ArrayList

Hay que usar el método `contains()`.

```
import java.util.ArrayList;

class Main {
    public static void main(String[] args) {
        // create an ArrayList
        ArrayList<String> languages = new ArrayList<>();
        languages.add("Java");
        languages.add("Python");
        languages.add("JavaScript");
        System.out.println("ArrayList: " + languages);

        // checks if 3 is present in the arraylist
        System.out.print("Is Java present in the arraylist: ");
        System.out.println(languages.contains("Java"));
    }
}

// Output: ArrayList: [Java, Python, JavaScript]
//          Is Java present in the arraylist: true
```

## Ver el índice de un elemento

Se usa el método `indexOf()`

Retorna un `int` o `-1` si no existe.

## Ver la última posición de un elemento

Se usa `lastIndexOf()`.

Retorna un `int` o `-1` si no existe.