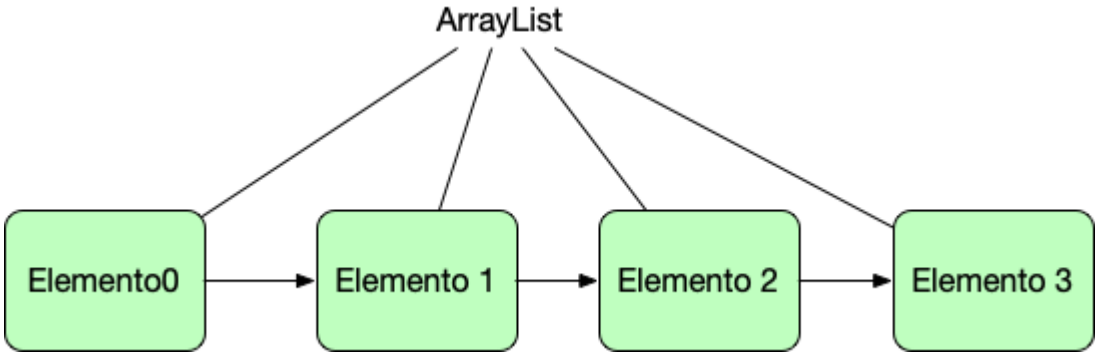
Tabla de Contenidos



Java ArrayList es una de las clases más utilizadas dentro del lenguaje Java. Si estás aprendiendo colecciones o quieres reforzar conceptos básicos, entender cómo funciona Java ArrayList es fundamental. Se trata de una estructura dinámica, flexible y muy práctica para almacenar y manipular datos.

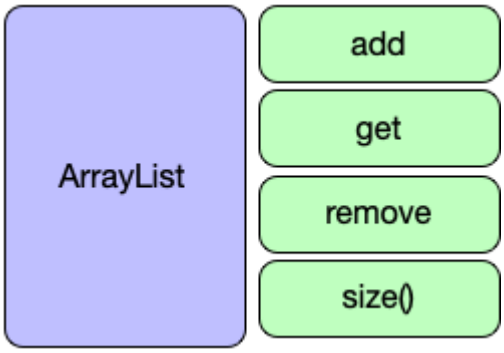



La clase ArrayList pertenece al paquete java.util y es una implementación de la interfaz List. A diferencia de los arrays tradicionales, ArrayList no necesita que definas el tamaño inicial

de forma estricta, ya que crece automáticamente cuando lo necesita.

# ¿Qué es exactamente Java ArrayList?

En términos sencillos, Java ArrayList es un array redimensionable. Internamente utiliza un array, pero añade lógica para ampliar su capacidad cuando se llena. Esto evita que tengas que crear manualmente un nuevo array y copiar elementos. Sus métodos fundamentales son muy sencillos vamos a verlos




Crear una instancia es muy simple:

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

Gracias a los genéricos, podemos indicar el tipo de datos que almacenará nuestra Java ArrayList, lo que aporta seguridad en tiempo de compilación.

# Métodos fundamentales de ArrayList

Estos son los métodos básicos que debes dominar cuando trabajas con Java ArrayList:

## add(E elemento)

Añade un elemento al final de la lista.

```
nombres.add("Ana");
nombres.add("Luis");
```

## get(int indice)

Devuelve el elemento en la posición indicada (empieza en 0).

```
String primero = nombres.get(0);
```

## set(int indice, E elemento)

Reemplaza el valor de una posición concreta.

```
nombres.set(1, "Carlos");
```

## remove()

Permite eliminar por índice o por valor.

```
nombres.remove(0);
nombres.remove("Carlos");
```

## size()

Devuelve el número de elementos almacenados.

```
int total = nombres.size();
```

## contains()

Comprueba si un elemento existe dentro de la lista.

```
if (nombres.contains("Ana")) {
  System.out.println("Está en la lista");
}
```

# Los métodos funcionales desde Java 8

Desde Java 8 incorpora métodos funcionales que permiten escribir código más limpio utilizando expresiones lambda.

## forEach(Consumer)

Recorre la lista aplicando una acción a cada elemento.

```
nombres.forEach(nombre -> System.out.println(nombre));
```

## removeIf(Predicate)

Elimina elementos que cumplan una condición.

```
nombres.removeIf(nombre -> nombre.startsWith("A"));
```

## replaceAll(UnaryOperator)

Transforma todos los elementos de la lista.

```
nombres.replaceAll(nombre -> nombre.toUpperCase());
```

## sort(Comparator)

Permite ordenar la lista fácilmente.

```
nombres.sort(String::compareTo);
```

# Conclusión

ArrayList es una estructura imprescindible en cualquier proyecto Java. Es sencilla, dinámica y muy versátil. Dominar sus métodos básicos como add, get o remove, junto con sus métodos funcionales introducidos en Java 8, te permitirá escribir código más limpio,

moderno y mantenible.

Si estás empezando con colecciones en Java o trabajas con frameworks como Spring Boot, entender bien cómo funciona ArrayList te dará una base sólida para construir aplicaciones más robustas.

Oracle Docs

## Otros articulos realacionados

- ¿List vs ArrayList que es mejor?
- Java Collections Remove con Java

- Java String y metodos fundamentales
- JPA Remove y Objetos gestionados