

CLASE ArrayList

Colección que permite elementos repetidos, tiene orden en sus elementos, rápida para accesos aleatorias, poco eficiente para insertar y borrar registros.

Paso 1: Importar paquete útil.

```
Import.java.util.*
```

Paso 2: Declarar el ArrayList.

```
ArrayList <tipo_dato> “nombre del ArrayList” = new ArrayList<tipo_dato>();
```

```
Ejemplo : ArrayList <Empleado> listaEmpleados=new ArrayList<Empleado>()
```

Ver en API <https://docs.oracle.com/javase/8/docs/api/> los tres constructores que existen.

Paso 3: Añadir Elementos

a) Método add

```
boolean add ( e E)
```

```
void add (int index, E elemento)
```

```
Ej1: Empleado emp 1 = new Empleado(“pepe”,2500);
```

```
listaEmpleados.add(emp1);
```

```
Ej2: listaEmpleados.add(new Empleado(“pepe”,2500));
```

Método para agregar un bloque de elementos vacíos.

```
ensureCapacity (int); - añade elementos al ArrayList.
```

Eliminar los elementos vacíos del final:

```
trimToSize() : elimina los elementos del final liberando espacio en memoria.
```

- b) **Método Set** :Reemplaza el elemento existente en una posición determinada por el elemento enviado.

E set(int index, E element)

Ejem: listaEmpleados.set(4, new Empleado(“pepe”,2500));

4º) **Recuperación-lectura de un elemento determinado:**

Método get:

E get (int Index); devuelve la referencial al objeto.

Si queremos obtener el contenido utilizar un método de la clase para obtenerlo.

String s = listaEmpleados.get (4).toString()

5º) **Recorrido de un ArrayList**

a) **Bucle for-convencional**

```
for (int i=0; i<listaEmpleado.size();i++){  
    Empleado e 0 listaEmpleado.get(i);  
}
```

b) **Bucle for-each**

Sirve para recorrer una colección de objetos de forma secuencial de principio a fin.

for (tipo <var_iteración> : colección) {bloque de instrucciones}

Ejem 1: con un array.

```
int num[] = { 1, 2, 3, 4, 5, 6, 7 };  
int sum = 0;  
for (int x : num ) suma+=x;
```

Ejem 2:

```
for (Empleado var : listaEmpleados){  
    System.out.println (var.toString());  
}
```

c) **Copiar un ArrayList en un array estático.: método toArray**

T [] toArray(T[])

Ejem:

```
Empleado arrayEstatico[] = new Empleado[listaEmpleados.size()];  
ListaEmpleado.toArray(arrayEstatico)
```

d) **Iteradores.**

Método iterator() : nos devuelve un iterador

Crear el iterador:

```
Iterator<tipo_Dato> "nombre iterador" = "nombre_coleccion".iterator();
```

Ejemplo : `Iterator<Empleado> miIterador = listaEmpleados.iterator();`

Utilizarlo:

```
While (miIterador.hasNext()){  
    System.out.println(iter.next().toString);  
}
```