

La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

DECLARACIÓN Y CREACIÓN DE UN ARRAYLIST

Declaración de un ArrayList de String cualquier otro Elemento u Objeto (float, Boolean, Object, ...)

De forma general un ArrayList en Java se crea de la siguiente forma:

ArrayList nombreArray = new ArrayList();

Esta instrucción crea el ArrayList *nombreArray* vacío.

Un arrayList declarado así puede contener objetos de cualquier tipo.

Por ejemplo:

```
ArrayList a = new ArrayList();
```

```
a.add("Lenguaje");
```

```
a.add(3);
```

```
a.add('a');
```

```
a.add(23.5);
```

Los elementos del arrayList a son:

"Lenguaje" 2 'a' 23.5

Es decir, **un ArrayList puede contener objetos de tipos distintos**.

Otra alternativa a esta declaración es indicar el tipo de objetos que contiene.

En este caso, el array solo podrá contener objetos de ese tipo.

De forma general:

ArrayList<tipo> nombreArray = new ArrayList();

tipo debe ser una clase. Indica el tipo de objetos que contendrá el array.

No se pueden usar tipos primitivos. Para un tipo primitivo se debe utilizar su clase envolvente.

Por ejemplo:

```
ArrayList<Integer> numeros = new ArrayList();
```

Crea el array *numeros* de enteros.

```
ArrayList<coche> coches = new ArrayList();
```

Crea el array *numeros* de coches.

MÉTODOS DE ARRAYLIST

Algunos métodos que proporciona ArrayList son:

MÉTODO	DESCRIPCIÓN
size()	Devuelve el número de elementos (int)
add(X)	Añade el objeto X al final. Devuelve true.
add(posición, X)	Inserta el objeto X en la posición indicada.
get(posicion)	Devuelve el elemento que está en la posición indicada.
remove(posicion)	Elimina el elemento que se encuentra en la posición indicada. Devuelve el elemento eliminado.

clear()	Elimina todos los elementos.
set(posición, X)	Sustituye el elemento que se encuentra en la posición indicada por el objeto X. Devuelve el elemento sustituido.
isEmpty();	Devuelve True si el ArrayList esta vacio. Sino Devuelve False

RECORRER UN ARRAYLIST

Podemos recorrerlo de forma clásica con un **bucle for**:

```
for(int i = 0;i<array.size();i++){
    System.out.println(array.get(i));
}
```

EJEMPLOS DE USO DE ARRAYLIST

Ejemplo 1:

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

nombres.add("Ana");

nombres.add("Luisa");

nombres.add("Felipe");

System.out.println(nombres);      // [Ana, Luisa, Felipe]

nombres.add(1, "Pablo");

System.out.println(nombres);      // [Ana, Pablo, Luisa, Felipe]

nombres.remove(0);

System.out.println(nombres);      // [Pablo, Luisa, Felipe]

nombres.set(0,"Alfonso");

System.out.println(nombres);      // [Alfonso, Luisa, Felipe]

String s = nombres.get(1);

String ultimo = nombres.get(nombres.size() - 1);

System.out.println(s + " " + ultimo);      // Luisa Felipe
```