

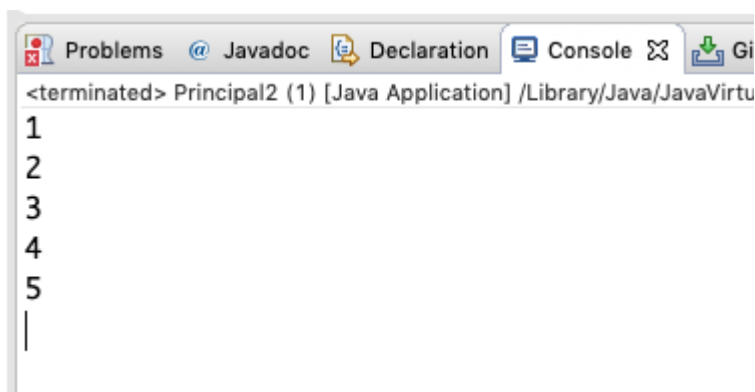
Java ArrayList count es quizás una de las preguntas más habituales en el mundo de Java cuando uno empieza . Todo el mundo necesita saber cuantos elementos dispone un ArrayList . Me acuerdo que cuando yo empecé a programar en Java rápidamente busqué la propiedad length para saber los objetos que contiene un ArrayList .

```
package com.arquitecturajava;

public class Principal {

    public static void main(String[] args) {
        int[] numeros=new int[]{1,2,3,4,5};;
        for( int i=0;i<numeros.length;i++) {
            System.out.println(numeros[i]);
        }
    }
}
```

El resultado le vemos en pantalla:



```
<terminated> Principal2 (1) [Java Application] /Library/Java/JavaVirtu
1
2
3
4
5
|
```

Sin embargo un ArrayList no funciona así y no tiene una propiedad length. Así pues si queremos saber la longitud de un ArrayList deberemos usar la propiedad size(). Esto nos

devolverá el tamaño de nuestro array y nos permitirá recorrerlo.

```
package com.arquitecturajava;

import java.util.ArrayList;

public class Principal2 {

    public static void main(String[] args) {
        ArrayList<Integer> numeros=new ArrayList();
        numeros.add(1);
        numeros.add(2);
        numeros.add(3);
        numeros.add(4);
        numeros.add(5);
        for( int i=0;i<numeros.size();i++) {
            System.out.println(numeros.get(i));
        }

    }

}
```



The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, Console, and a download icon. The console output is as follows:

```
<terminated> Principal2 (1) [Java Application] /Library/Java/JavaVirtu
1
2
3
4
5
|
```

Objetos e igualdad

Hasta aquí todo es muy sencillo sin embargo muchas veces la siguiente pregunta que aparece es cuantas veces aparece el número 3 en el ArrayList o el texto tal en la lista Para eso hay que recordar que lo más importante es tener correctamente sobrescritos los métodos equals y hashCode de la clase en cuestión. Yo voy a hacer un ejemplo apoyándome en la clase String que pertenece al API de Java y usar una lista de cadenas.

```
package com.arquitecturajava;

import java.util.ArrayList;

public class Principal2 {

    public static void main(String[] args) {
        ArrayList<String> textos=new ArrayList<String>();
        textos.add("hola");
        textos.add("que");
        textos.add("tal");
        textos.add("estas");
        textos.add("hola");
        int contador=0;
        for(String texto:textos) {
            if (texto.equals("hola")) {
                contador++;
            }
        }
        System.out.println(contador);
    }
}
```

En este caso estoy buscando cuantas veces aparece la cadena "hola" en un ArrayList. Uso un simple contador y lo voy incrementando según la igualdad se cumple . Esta es una solución digamos.... habitual . Pero se pueden localizar soluciones más elegantes

Java ArrayList count y funciones

```
package com.arquitecturajava;

import java.util.ArrayList;

public class Principal4 {

    public static void main(String[] args) {
        ArrayList<String> textos=new ArrayList<String>();
        textos.add("hola");
        textos.add("que");
        textos.add("tal");
        textos.add("estas");
        textos.add("hola");
        System.out.println(textos.stream().filter(t->t.equals("hola")).count()
        );

    }

}
```

Esta solución es mucho más compacta y nos evitamos el usar variables adicionales . Convertimos la lista en un Stream **y la filtramos** para finalmente contar el número de elementos existentes

¿Es suficiente?

En principio parece que sí y que hemos usado las capacidades de Java 8 de forma correcta y elegante pero a veces es importante conocer las APIs con las clases y métodos que aportan . Mucho mas elegante hubiera sido usar el método frequency de la clase Collections ;).

```
package com.arquitecturajava;

import java.util.ArrayList;
import java.util.Collections;

public class Principal4 {

    public static void main(String[] args) {
        ArrayList<String> textos=new ArrayList<String>();
        textos.add("hola");
        textos.add("que");
        textos.add("tal");
        textos.add("estas");
        textos.add("hola");
        System.out.println(Collections.frequency(textos,"hola"));

    }

}
```

Cuanto más sepamos del API del lenguaje mejores programadores seremos y más sencillo será nuestro código

Otros artículos relacionados

- [Java Stream for loop](#)
- [Java Stream](#)

Java ArrayList count y las APIs de Java.

- [Java Stream y Colecciones](#)
- [Java Collections](#)



Cecilio Álvarez Caules

Cecilio Álvarez Caules Oracle Java Certified Architech

Java ArrayList count y las APIs de Java.

Java ArrayList count y las APIs de Java.