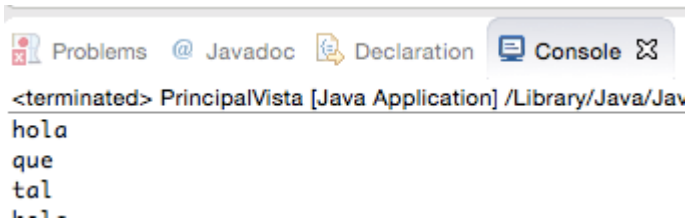
Java Collections Views es uno de los conceptos que más suele sorprender a los programadores cuando comienzan a trabajar con el framework de colecciones de Java. Imaginemonos que disponemos del siguiente Array de Strings en Java.

```
String[] listatexto = new String[] { "hola", "que", "tal" };
```

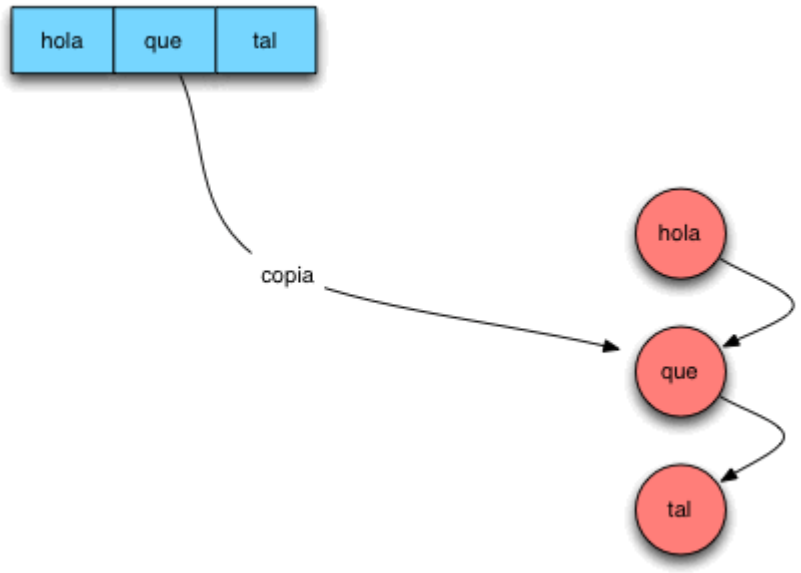
En muchas ocasiones querremos convertir al Array de Strings en un List de Strings. Esta operación es muy sencilla y la podemos realizar a través de la clase Arrays y su método toList(). Hecho esto el siguiente paso es recorrer la lista con una estructura forEach.

```
List<String> lista = Arrays.asList(listatexto);

for (String s : lista) {

System.out.println(s);

}
```

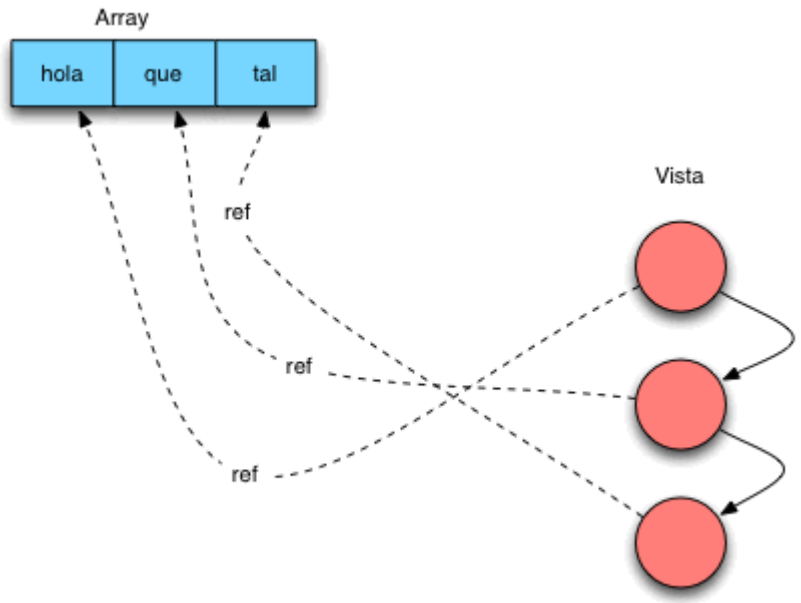
Todo funciona correctamente:



Parece que el JDK nos ha realizado una copia de nuestro Array a una estructura de lista.



Sin embargo esta apreciación no es correcta ya que lo que se ha hecho no es una copia , sino que se ha generado una “Vista” diferente sobre los mismos datos.

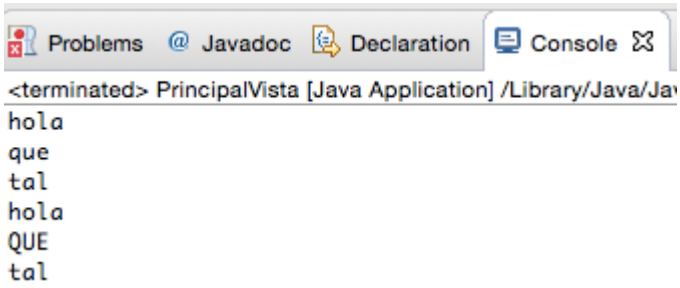


Esto lo podemos comprobar simplemente cambiando uno de los elementos del Array

original.

```
String[] listatexto = new String[] { "hola", "que", "tal" };

List<String> lista = Arrays.asList(listatexto);

for (String s : lista) {

System.out.println(s);

}

listatexto[1] = "QUE";

for (String s : lista) {

System.out.println(s);

}
```

Volvemos a recorrer la lista que habíamos construido y nos podremos dar cuenta que sin haber hecho ninguna modificación a la lista nueva esta se ve afectada por los cambios. Esto se debe a que no hemos construido una nueva Lista sino que hemos generado una vista sobre los datos originales.



Este tipo de situaciones se produce en muchas situaciones a la hora de utilizar el framework de colecciones . Un caso similar al anterior es el manejo de sublistas que tienen el mismo comportamiento.

```
List<String> otraLista= new ArrayList<String>();

otraLista.add("Programando");
otraLista.add("con");
otraLista.add("Java");

List<String> nueva= otraLista.subList(0, 2);

for (String s : nueva) {

System.out.println(s);

}

otraLista.set(1, "EN");


for (String s : nueva) {
```

```
System.out.println(s);

}
```

En este caso generamos una sublista “nueva” a partir de la lista original “otraLista” . Sin embargo cuando cambiemos la original la sublista se ve afectada.

```
Programando
con
Programando
EN
```

Tengamos en cuenta estos conceptos cuando trabajamos con el framework de colecciones de Java .

Otros artículos relacionados: Java Generics , Colecciones y Google Guava , Java Collections List vs Set