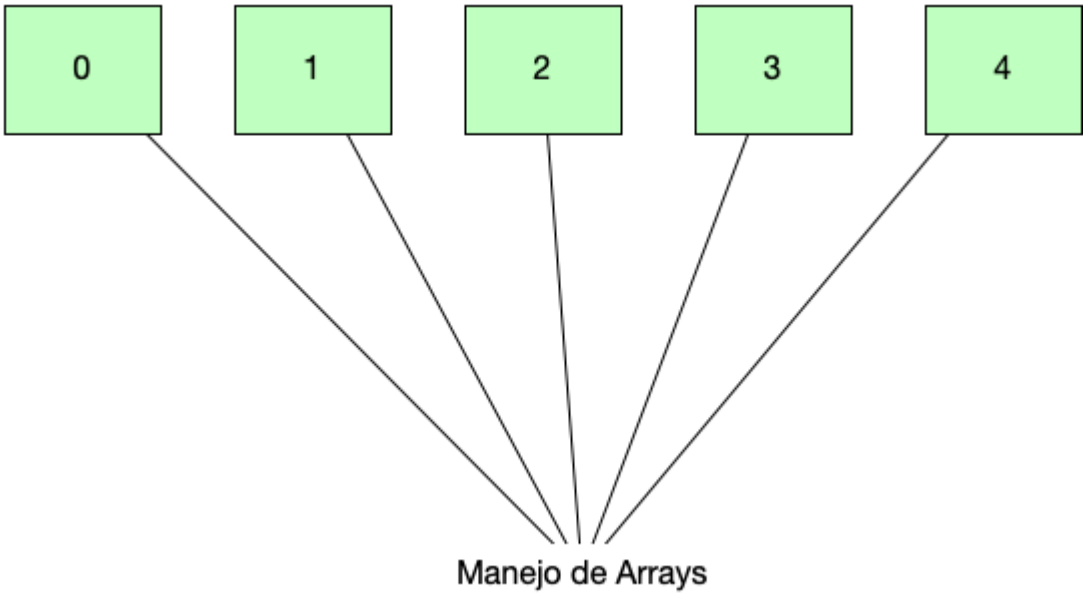
Java Array for , o como recorrer un Array en Java es una de las cosas más sencillas que el lenguaje soporta ya que simplemente se trata de iterar entre una lista de elementos de un Array.



Pero incluso las cosas más sencillas pueden tener sus trucos o curiosidades . Vamos a ver algunos ejemplos sencillos. Para ello partiremos del típico bucle for .

```
public class Principal {

	public static void main(String[] args) {
		String[] lista= new String []{"hola",
"que","tal","estas"};
		for (int i=0;i<lista.length;i++) {
			System.out.println(lista[i]);
		}
	}

}
```

Si le recorremos pues obtendremos el resultado de una forma muy directa:

```
hola
que
tal
estas
```

¿Es la mejor opción? . Bueno es una de las opciones más clásicas sino la más clásica. Con esta opción podemos si queremos imprimir también la posición en la que nos encontramos si nos hiciera falta. Sin embargo un Array permite algunas otras opciones curiosas. Un array es un objeto no una clase , es algo antiguo que se heredo de C++ y tiene sus peculiaridades.

Java Array for y forEach

Un Java Array se puede recorrer como si se tratara de una colección aunque no lo sea ya que el lenguaje es muy flexible y podemos recorrerlo con una estructura foreach.

```
package com.arquitecturajava;

public class Principal2 {

	public static void main(String[] args) {
		String[] lista= new String []{"hola",
"que","tal","estas"};
		for (String s : lista) {
			System.out.println(s);
		}
	}

}
```

El resultado será idéntico , esta opción terminará siendo en muchos casos la preferida de todo el mundo por temas de homogeneidad. ¿Hay alguna opción más? . Si por supuesto

Java Arrays y Streams

Un Java Array se puede convertir en un Stream aboyándonos en la clase Java Arrays (en plural) y realizar las operaciones que necesitemos con él.

```
package com.arquitecturajava;

import java.util.Arrays;

public class Principal3 {

	public static void main(String[] args) {
		String[] lista= new String []{"hola",
"que","tal","estas"};
		Arrays.
		stream(lista).filter((s)->s.startsWith("h"))
		.forEach(System.out::println);
	}

}
```

El resultado se filtrará y solo aparecerá la palabra hola en la consola:

```
hola
```

Acabamos de ver el uso de Java Array for . Los Arrays son unos objetos peculiares del lenguaje Java que siempre nos dan alguna sorpresa pero que nos permiten transformarlos de formas muy similares a las colecciones si nos conocemos donde tocar a nivel del JDK.

Otros artículos relacionados

- Java List y sus mejoras en Java 8

- Java 9 Factory Methods (List,Set,Map)
- Java Ordered vs Sortered y sus diferencias
- Java Iterator vs ForEach
- Java Arrays