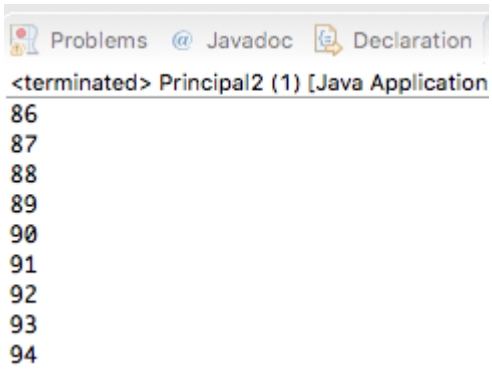
El concepto de Java 8 IntStream hace referencia a Streams compuestos por números enteros. Es algo que en muchas ocasiones puede sernos util a la hora de trabajar ya que aporta algunos métodos interesantes, vamos a verlos:

```
package com.arquitecturajava;

import java.util.stream.IntStream;

public class Principal {

	public static void main(String[] args) {

		IntStream it= IntStream.of(1,2,3,4,5,6);
		it.forEach(System.out::println);

	}
}
```

Nuestro primer ejemplo crea un Stream de seis números y lo imprime por pantalla. Podemos hacer lo mismo de una forma más compacta usando el método range() e imprimir una lista de 100 números por ejemplo:

```
package com.arquitecturajava;

import java.util.stream.IntStream;
```

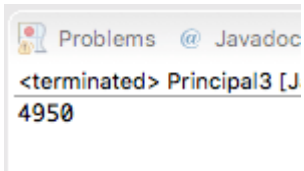
```
public class Principal2 {

public static void main(String[] args) {

 IntStream it= IntStream.range(0, 100);
 it.forEach(System.out::println);

}

}
```



De esta forma es muy sencillo realizar cálculos sobre la suma de n términos, bastaría con modificar el código e invocar al método sum():

```
 System.out.println(IntStream.range(0, 100).sum());
```

El resultado imprimirá :



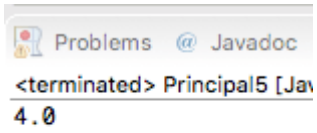
Usando IntStream

El código es mucho más compacto que usar el clásico bucle for. A veces se nos olvida que los IntStreams pueden simplificar sobre manera operaciones cotidianas. Supongamos que tenemos un array de cadenas y queremos obtener la media de la longitud de todas las cadenas . Se trata de un programa sencillo de implementar en Java clásico:

```
package com.arquitecturajava;

public class Principal4 {

	public static void main(String[] args) {

		String[] cadenas= {"hello","my","name",
"is","cecilio"};

		double total=0;
		double media=0;

		for (int i=0;i&amp;amp;lt;cadenas.length;i++) {

			total+=cadenas[i].length();
		}

		media=total/cadenas.length;
		System.out.println(media);

	}
```

```
}
```

El resultado por consola sera:

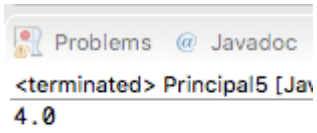


El código es muy muy sencillo de entender pero son bastantes lineas. ¿ Podríamos haber usado Streams numéricos para simplificarlo? . La respuesta es SI y el código queda mucho más elegante.

```
package com.arquitecturajava;

import java.util.Arrays;

public class Principal5 {

	public static void main(String[] args) {

		String[] cadenas = { "hello", "my", "name", "is",
"cecilio" };

System.out.println(Arrays.stream(cadenas).mapToInt(String::length).ave
rage().getAsDouble());

	}
```

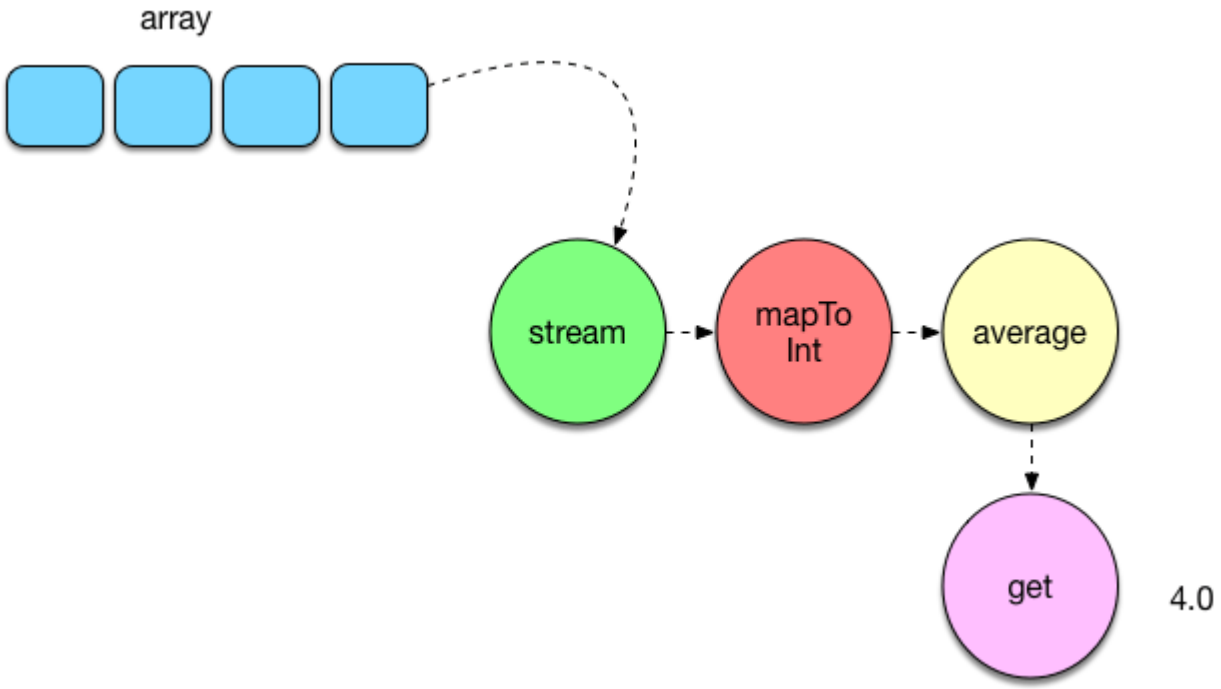
```
}
```

El resultado será idéntico.



Hemos realizado varias operaciones.

1. Hemos convertido el array en un stream
2. Hemos mapeado la propiedad length para convertir el Stream en un Stream de números.
3. Después hemos invocado la propiedad average para calcular la media
4. Finalmente se usa el método get para obtener un resultado.



Acabamos de usar Java 8 para reducir un bucle for y una serie de transformaciones a una única linea.

Otros artículos relacionados :

Java Streams

Java 8 Lambda Expressions (I)

Java Reference methods