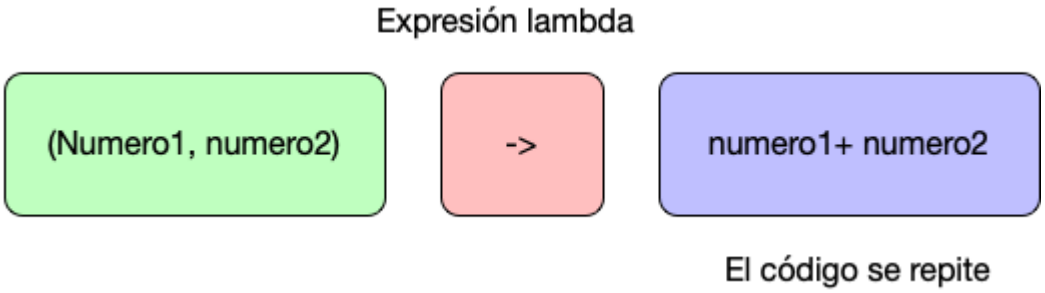
¿Para que sirven los :: en Java? . Esta es una pregunta muy recurrente para mucha gente que empieza formarse con programación funcional en el lenguaje. El operador :: o de doble dos puntos es un operador muy habitual en programación funcional que hace referencia a como aumentar la reutilización de una expresión lambda o función.



Java y funciones

Vamos a ver un ejemplo sencillo de como utilizarle. Para ello imaginémonos que disponemos de una lista de numeros y queremos sumarlos con programación funcional a través de un stream. La operación más sencilla antes de sumarlos es simplemente mostrarlos .

```
package com.arquitecturajava;

import java.util.Optional;
import java.util.stream.Stream;

public class Principal2 {

	public static void main(String[] args) {

		Stream<Integer> lista = Stream.of(5, 7, 2, 1);

		lista.forEach((n)->System.out.println(n));

	}
```

```
}
```

Hemos recorrido con una operación foreach todos los elementos

Java y Reducciones

Ahora podemos acceder a otra operación más avanzada que en vez de recorrerlos los sume. Esta operación es conocida como una operación de reducción y permite aplicar una función a un flujo o stream de datos para obtener un único resultado final . Vamos a hacerlo en código:

```
                Stream<Integer> lista = Stream.of(5, 7, 2, 1);

                Optional<Integer> oResultado = lista.reduce((numero1,
numero2) -> numero1 + numero2);

                if (oResultado.isPresent()) {

                        System.out.println(oResultado.get());
                }
```

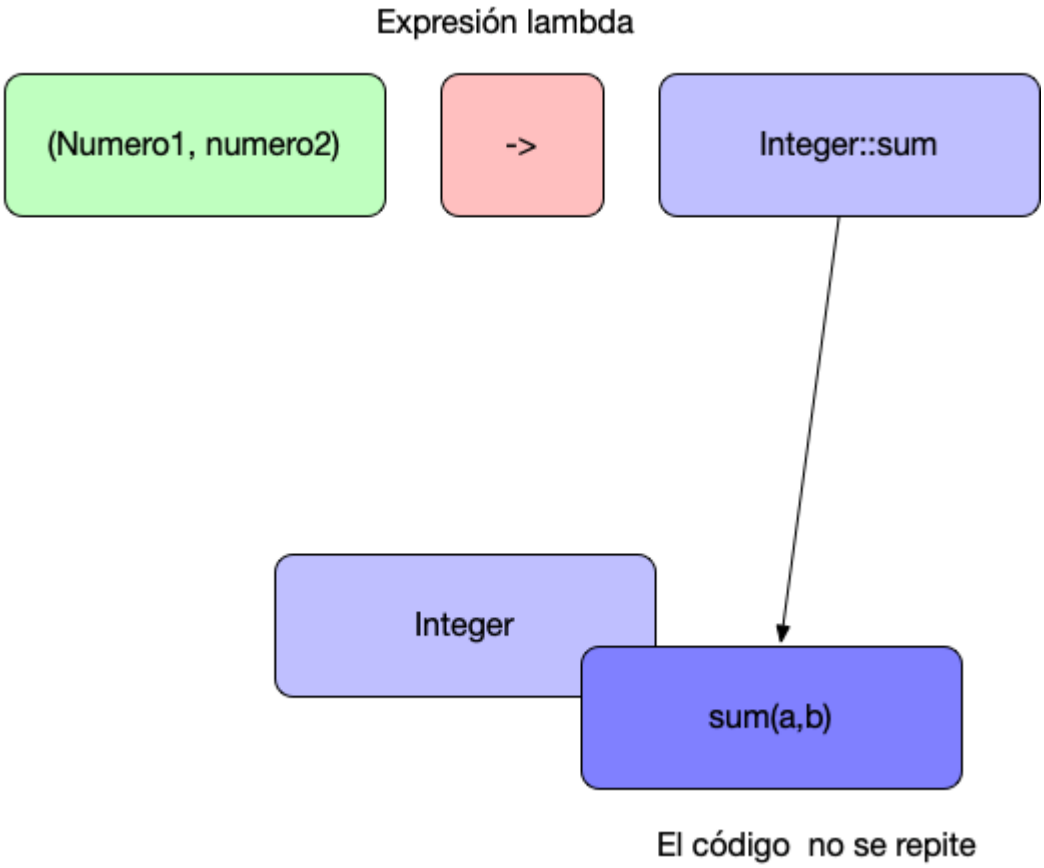
Esto imprimirá el resultado :

```
15
```

Esta claro que esta operación simplifica mucho y podemos construir un código mucho más limpio que suma todos . Sin embargo una operación de reducción hay que definirla para cada grupo de datos que manejamos esto implica generar mas funciones cada vez que realicemos una suma.

Usando :: en Java

Para simplificar esta operación a nivel de Java podemos en vez de definir una operación de reducción . Referenciar a otra función que ya exista en alguna clase que construyamos o en alguna clase que aporte el JDK. En este caso al tratarse de una operación tan sencilla ese método existe ya en el JDK y pertenece a la clase Integer . Concretamente el método sum (a,b) que permite sumar dos elementos .



Por lo tanto podemos simplificar todavía las cosas más y sumar los dos elementos que tenemos usando un método de referencia sin tener que añadir código extra usando :: en Java y referenciando el método existente.

```
Stream<Integer> lista2 = Stream.of(5, 7, 2, 1);
Optional<Integer> oResultado2 =
```

```
lista2.reduce(Integer::sum);

                if (oResultado2.isPresent()) {

                        System.out.println(oResultado.get());
                }
```

Usemos más los métodos de referencia usando el operador :: .

Otros artículos relacionados

- Kotlin vs Java y el manejo de clases
- Java Stream Sum y Business Objects
- Java Lambda reduce y wrappers
- Curso Java EE desde Cero