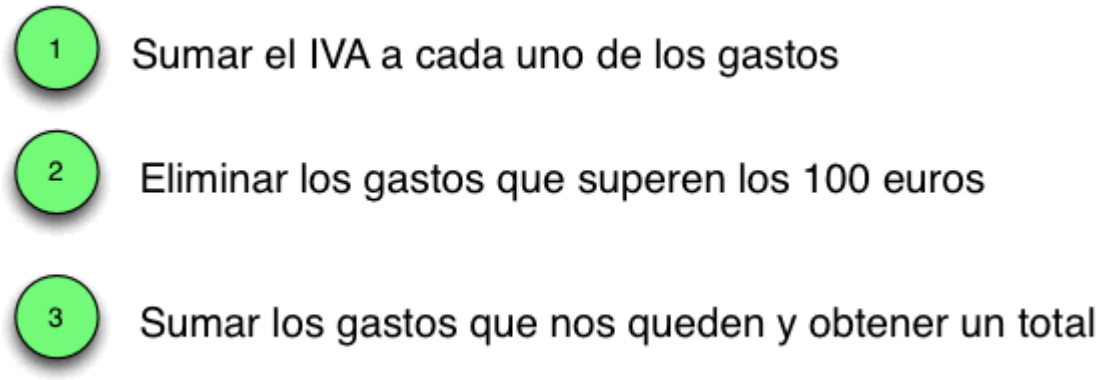
Tabla de Contenidos


- Solución Clásica
- Programación y Personas
- Java y Streams
- Otros Artículos relacionados


Cada día la programación funcional está más de moda y Java 8 tiene un buen soporte para ello .Sin embargo muchas veces no tenemos muy claro como aplicarla. Vamos a construir un ejemplo sencillo. Supongamos que tenemos una lista de gastos de viaje diarios y la empresa se hará cargo de todos ellos. Eso sí , si una vez sumado a los gastos el IVA el importe supera los 100 euros la empresa no lo pagará ya que considerará que nos hemos excedido en lo que gastamos en un día. Las operaciones que tendremos que realizar resumiendo son:




# Solución Clásica

Bueno vamos a solventar este problema y para ello vamos a construir un Array de Objetos de tipo Gasto y programar la lógica necesaria para que las operaciones nos funcionen correctamente :

```
package com.arquitecturajava.streams;

import java.util.ArrayList;

public class Principal {

 public static void main(String[] args) {

 ArrayList&amp;amp;lt;Gasto&amp;amp;gt; lista= new
ArrayList&amp;amp;lt;Gasto&amp;amp;gt;();

 lista.add(new Gasto("A",80));
 lista.add(new Gasto("B",50));
 lista.add(new Gasto("C",70));
 lista.add(new Gasto("D",95));

 double totalPago=0;

 for (Gasto g:lista) {

 if (g.getImporte()*1.21&amp;amp;lt;100) {

 totalPago=totalPago+ g.getImporte()*1.21;

 }
 }

 System.out.println(totalPago);

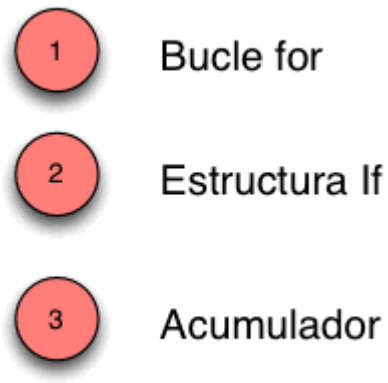
 }
```

}

Revisando el código nos podemos dar cuenta que la lógica de negocio era bastante sencilla de aplicar . Usamos un bucle for, recorremos el array y con una estructura if  que chequea si nos pasamos de 100 euros decidimos si acumulamos el importe o no el programa nos imprimirá

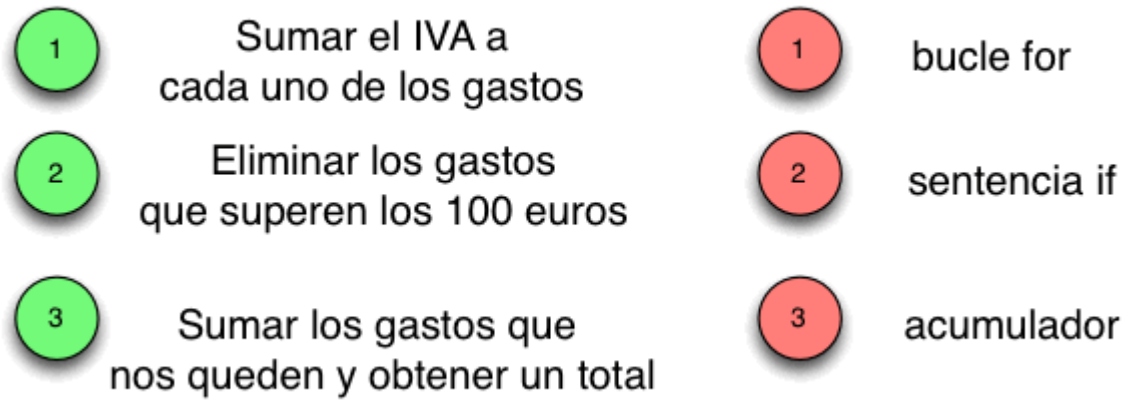
242.0

Hemos realizado las siguientes operaciones :




# Programación y Personas

A veces los programadores nos alejamos mucho del pensamiento clásico humano que es mucho más lineal . Es decir es mas del estilo incrementamos a todos el IVA , eliminamos los gastos que pasen de 100 euros  ,sumamos el resto.  De ahí que aprender a programar nunca haya sido algo “facil” sino mas bien todo lo contrario ya que los conceptos son difíciles de encajar:




La programación funcional nos puede ayudar a acercarnos mas a un pensamiento humano clásico y hacer que los dos enfoques que tenemos encajen mejor.

# Java y Streams

Para poder solventar el problema que tenemos de una forma más amigable vamos a usar el concepto de Java 8 Stream y programación funcional. Un Stream no es ni mas ni menos que un conjunto de funciones que se ejecutan de forma anidada.

```
package com.arquitecturajava.streams;

import java.util.ArrayList;

public class PrincipalFuncional {

public static void main(String[] args) {

ArrayList&amp;amp;lt;Gasto&amp;amp;gt; lista= new
ArrayList&amp;amp;lt;Gasto&amp;amp;gt;();
```

```
lista.add(new Gasto("A",80));
lista.add(new Gasto("B",50));
lista.add(new Gasto("C",70));
lista.add(new Gasto("D",95));

double resultado=lista.stream()
.mapToDouble(gasto-&amp;amp;gt;gasto.getImporte()*1.21)
.filter(gasto-&amp;amp;gt;gasto&amp;amp;lt;100)
.sum();

System.out.println(resultado);

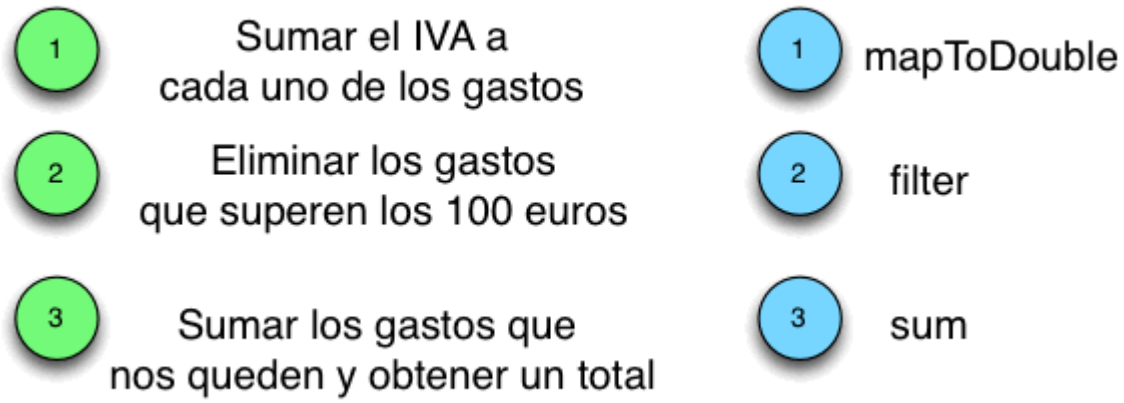
}

}

&amp;amp;amp;nbsp;
```

Como podemos ver hemos usado en este nuevo código un stream y lo hemos dividido en tres pasos:

1. Sumamos el IVA a cada uno de nuestros gastos
2. Filtramos aquellos gastos que se pasan del tope
3. Sumamos los que restan
   De esta forma habremos resuelto nuestro problema de una forma mucho más funcional y mucho más cercana a como las personas enfocamos.




Poco a poco tenemos que irnos haciendo con este tipo de programación que nos será muy util

**CURSO Java Herencia**
**GRATIS**
**APUNTATE!!**

## Otros Artículos relacionados

- Java Lambda ForEach
- Introducción a Java Lambda
- Java Generics