Java Function Composition (Funciones Composición y Java) . ¿ Cómo podemos componer funciones en Java ? . A día de hoy y a través de Java 8 nos encontramos que disponemos del concepto de función y que podemos construir funciones en Java usando expresiones Lambda. Es una operación realmente sencilla vamos a verlo con un ejemplo con la función duplicar.

```
package com.arquitecturajava;

import java.util.function.Function;

public class Principal {

  public static void main(String[] args) {
    Function<Double, Double> duplicar = x -> x * 2;
    double resultado=duplicar.apply(2.0);
    System.out.println(resultado);

  }

}
```

En este caso lo que estamos es diseñando una función muy sencilla aquella que recibe un valor numérico de tipo double y lo duplica. Para ello vamos a usar el interface Function de Java 8 que se encarga de definir las funciones que reciben un parámetro y devuelven un resultado . Al ejecutar el código nos imprimirá un 4 por la consola.



Acabamos de construir nuestra primera función y la hemos ejecutado . Ahora bien las funciones soportan composición a nivel del lenguaje Java veamos cómo combinar esta función con otra.

Funciones Composición y Java

Para combinar dos funciones en Java necesitamos evidentemente añadir una función adicional:

```
package com.arquitecturajava;

import java.util.function.Function;

public class Principal {

  public static void main(String[] args) {
    Function<Double, Double> duplicar = x -> x * 2;
    double resultado=duplicar.apply(2.0);
    System.out.println(resultado);

    Function<Double,Double> sumar2= x->x+2;
    double resultado2=sumar2.apply(resultado);
    System.out.println(resultado2);
  }

}
```

Ahora tenemos dos funciones si las ejecutamos una después de otra tendremos por la consola los siguientes mensajes:

```
package com.arquitecturajava;

import java.util.function.Function;

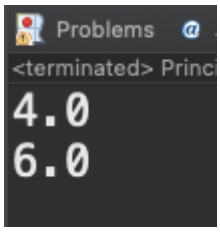
public class Principal {
```

```
  public static void main(String[] args) {
    Function<Double, Double> duplicar = x -> x * 2;
    double resultado=duplicar.apply(2.0);
    System.out.println(resultado);

    Function<Double,Double> sumar2= x->x+2;
    double resultado2=sumar2.apply(resultado);
    System.out.println(resultado2);
  }

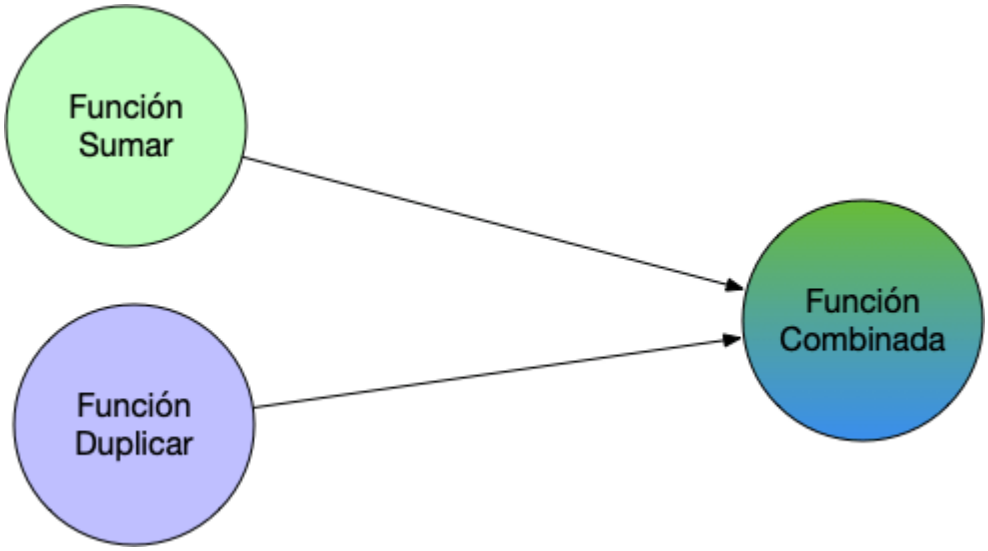
}
```

El resultado en consola es :



Java Function Composition

Es momento de ver como podemos combinar ambas funciones para que se ejecuten de forma combinada .



Para ello modificaremos el código y usaremos el método andThen del interface Function que nos permite generar una nueva Función combinando el comportamiento de las dos funciones anteriores.

```
package com.arquitecturajava;

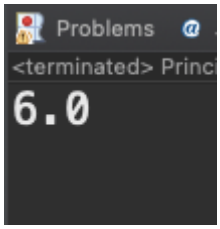
import java.util.function.Function;

public class Principal {

  public static void main(String[] args) {
    Function<Double, Double> duplicar = x -> x * 2;
    Function<Double,Double> sumar2= x->x+2;
    Function<Double, Double> composicion = duplicar.andThen(sumar2);
    double resultado2=composicion.apply(2.0);
    System.out.println(resultado2);
  }

}
```

El resultado por la consola:



Aprendamos los conceptos de programación funcional que nos ayudarán en la gestión de nuestro día a día y permitirán un código más flexible:

Otros artículos relacionados

1. ¿Que es un Java Stream?
2. ¿Qué es un Java Lambda?
3. ¿Qué es un POJO en Java?
4. ¿Qué es un Java Optional?