

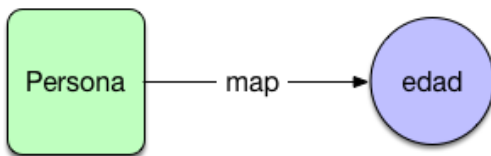
Blog sobre Java EE

Estás aquí: [Inicio](#)/[Java SE](#)/[Java Lambda y Streams](#)/[Java Stream map y estadísticas](#)

Java Stream map y estadísticas

18 noviembre 2017 por [Cecilio Álvarez Caules](#) — 2 comentarios

El uso de **Java Stream map** es una de las operaciones más comunes cuando trabajamos **con un flujo de Streams** . El método **map** nos permite realizar una transformación **rápida de los datos y muy directa sobre el flujo original**.

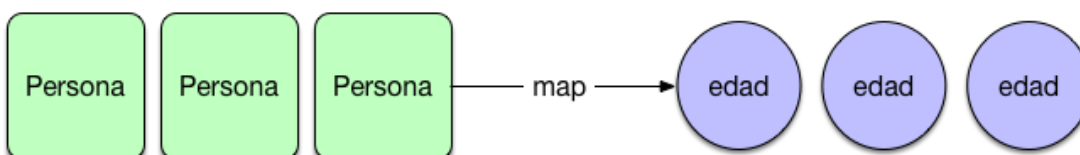


Vamos a ver algunos ejemplos que nos ayuden a clarificar como se utiliza **Java Stream map**. Para ello nos vamos a apoyar en una lista de Personas .

```
1 Persona p1= new Persona("pedro",20,"perez");  
2 Persona p2= new Persona("juan",25,"perez");  
3 Persona p3= new Persona("ana",30,"perez");  
4 List<Persona> lista= new ArrayList<Persona>();  
5 lista.add(p1);  
6 lista.add(p2);  
7 lista.add(p3);
```

Java Stream Map

La primera operación que vamos a realizar es convertir nuestra lista de personas a una lista de números enteros.



Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)
[ACEPTAR](#)

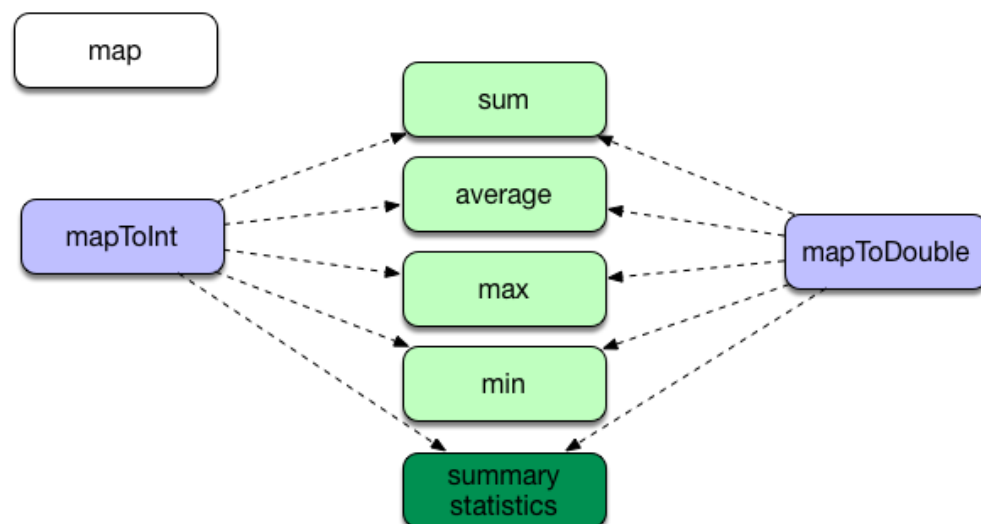
<terminated> Pr

75

Sin embargo esta aunque este una solución **válida no es la mejor solución**

Stream Map vs MapInt y MapDouble

El `map` viene con dos métodos adicionales orientados a trabajar con datos numéricos. Estos métodos son `mapToInt` y `mapToDouble`. Si cambiamos nuestro método de `map` a `mapToInt` o `mapToDouble` se nos abrirá la posibilidad de acceder a métodos adicionales muy orientados a estadísticas.



Veamos su código:

```

1  int total=lista.stream().mapToInt(Persona::getEdad).sum();
2  System.out.println(total);
3  lista.stream().mapToDouble(Persona::getEdad).average().ifPresent(System.out::println);
4  lista.stream().mapToInt(Persona::getEdad).max().ifPresent(System.out::println);
5  lista.stream().mapToInt(Persona::getEdad).min();

```

Podemos ver el resultado por la consola.

Problems @ Javadoc D

<terminated> Principal1 [Java App]

75

75

25.0

30

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)[ACEPTAR](#)

```
2 | System.out.println(estadisticas.getAverage());
```

Otros artículos relacionados:

18

[Stream Filter y Predicates](#)[2. Java Stream String y Java 8](#)

18

[3. Java Stream Collectors y su uso](#)[IntStream](#)[5. Java Lambda reduce y wrappers](#)[6. Java List Directory en Java 8 con Streams](#)[7. El concepto de Java constructor reference](#)

Archivada en: [Java Lambda y Streams](#)

Etiquetada con: [Java8Tips](#)

Comentarios



[AntuanSoft](#) dice

8 noviembre, 2017 en 11:10

Que poquito me gustan los lamdas y la programación funcional mezclada con la POO, para el mantenimiento del código y la legibilidad del mismo no me parecen lo mejor.

Pero gracias a tus post voy viendo lo útil y rápido que es, aunque nunca he sido amigo de hacer muchas cosas en una sólo linea de código ni una clase con centenas de líneas de código, sobre todo por legibilidad ya que el programador que venga detrás puede no entenderlo bien.

Gracias Cecilio.

[Responder](#)

De nada 🙏 . El tema de la programación funcional es poco a poco . Con tiempo la integraremos

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

Responder

18 [JAVA SE](#) [JAVA SE +](#) [SPRING](#) [JAVA EE](#) [JAVASCRIPT](#) [FRAMEWORKS JS](#) [ARQUITECTURA](#) [MIS LIBROS](#)

18

Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario

COMPARTIR

Nombre *

Correo electrónico *

Web

PUBLICAR COMENTARIO

BUSCAR

Buscar en este sitio ...

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

plugin cookies

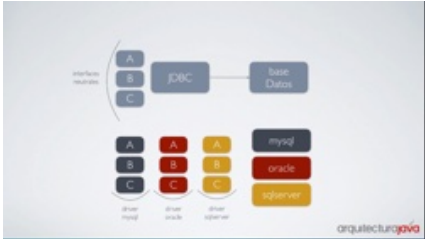
ACEPTAR

arquitecturajava

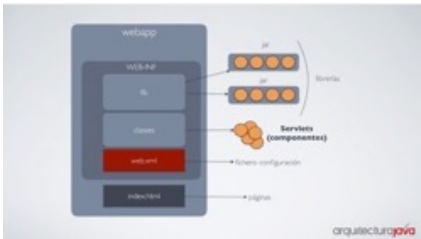
18

18

Java JDBC



Servlets



POPULAR

- Java Stream Context y simplificación de Streams
- Spring GetMapping ,PostMapping etc
- Java Stream map y estadísticas
- Spring 5 Hello World
- Spring Boot WAR sin Microservicios

Continuando con el desarrollo de Spring

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR

18

18

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

- ¿Qué es Spring Boot?
- Java Stream map y estadísticas
- Usando Java Session en aplicaciones web
- Java Constructores this() y super()
- Java Iterator vs ForEach
- Ejemplo de JPA , Introducción (I)
- ¿Cuales son las certificaciones Java?
- Usando el patron factory
- ¿Qué es Gradle?
- Ejemplo de Java Singleton (Patrones y ClassLoaders)
- Introducción a Servicios REST
- Uso de Java Generics (I)
- Java Stream Context y simplificación de Streams
- REST JSON y Java
- Mis Libros
- Spring GetMapping ,PostMapping etc
- Comparando java == vs equals
- Spring MVC Configuración (I)
- Java Override y encapsulación
- ¿Qué es un Microservicio?
- ¿Tiene futuro JSF?
- Eclipse Git , Repositorios locales y remotos

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

[plugin cookies](#)

Copyright © 2017 · [eleven40 Pro Theme](#) en [Genesis Framework](#) · [WordPress](#) · [Acceder](#)

18

18