



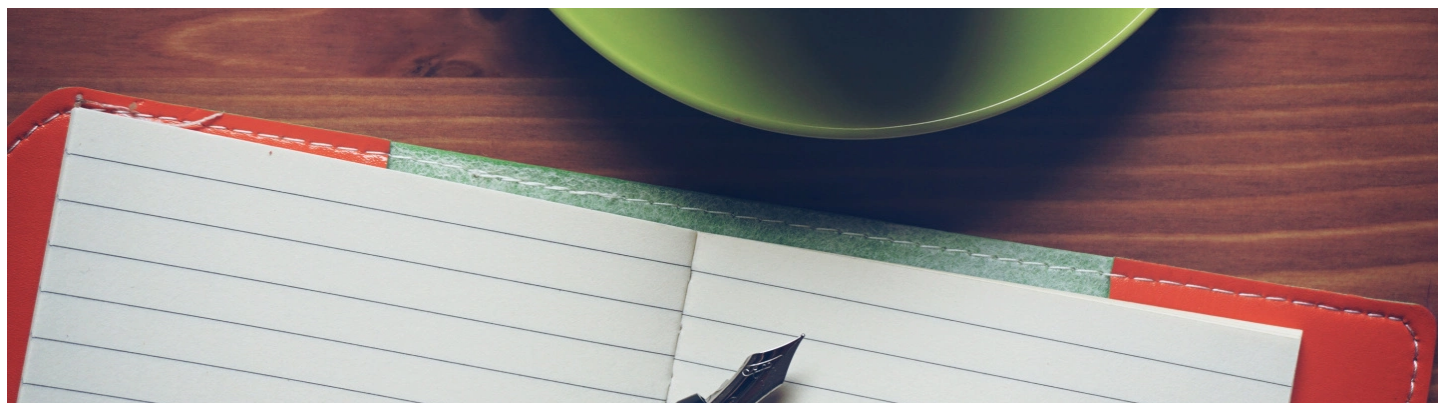
devs4j

EL MEJOR SITIO WEB SOBRE PROGRAMACIÓN EN ESPAÑOL.

[HOME](#)

[ABOUT](#)

[CONTACT](#)



Anuncios

Earn money from your WordPress site [SIGN UP >](#)

WordAds

[Report this ad](#)

Java 8 Streams – Remueve valores duplicados en una lista

📅 [HACE 5 DÍAS](#) 💬 [DEJA UN COMENTARIO](#)

Rate This

En el ejemplo del día de hoy hablaremos de un problema común, remover datos duplicados de una lista, como sabemos existen muchos algoritmos para hacerlo, en este post hablaremos de dos formas utilizando Streams.

1 Utilizando distinct()

El método `distinct` nos permitirá remover datos duplicados de una lista.

```
1 public static void main(String[] args) {  
2     List numbers = Arrays.asList(7, 7, 7, 7);  
3     numbers = numbers.stream().distinct().collect(toList());  
4     System.out.println(numbers);  
5 }
```

En el código anterior podemos ver que generamos una lista con muchos números duplicados, generamos un stream y hacemos uso de **`distinct()`** lo cual removerá datos duplicados y lo devolvemos como lista, con lo anterior tendremos la siguiente salida:

```
1 | [7, 2, 3, 100, 200]
```

Como se puede ver, aunque en la lista original tenemos repetidos los números multiples veces, al utilizar **distinct()** todos los elementos duplicados fueron removidos y tendremos resultados únicos.

2 Utilizando set

Como sabemos existe una estructura de datos en Java llamada Set la cuál solo almacena datos únicos y es muy utilizada para remover datos duplicados, veamos un ejemplo:

```
1 | public static void main(String[] args) {  
2 |     List numbers = Arrays.asList(7, 7, 7, 7);  
3 |     Set nums = numbers.stream().collect(Collectors.toSet());  
4 |     System.out.println(nums);  
5 | }
```

En el código anterior podemos ver que nuestra entrada es una lista y lo único que hacemos es a través del método **collect(..)** la transformamos a un Set y con esto aseguramos que la salida será solo datos únicos, veamos la salida:

```
1 | [2, 3, 100, 7, 200]
```

Del mismo modo que con el método **distinct** el resultado es un conjunto de números únicos en la salida, es importante considerar el tipo de colección que queremos utilizar en nuestro problema ya que eso determinará cuál de los dos mecanismos utilizaremos.

Para enterarte sobre futuros posts te recomendamos seguirnos en nuestras redes sociales: https://twitter.com/geeks_mx (https://twitter.com/geeks_mx) y <https://www.facebook.com/geeksJavaMexico/> (<https://www.facebook.com/geeksJavaMexico/>).

Autor: Alejandro Agapito Bautista

Twitter: [@raidentrance](https://twitter.com/raidentrance)

(<https://geeksjavamexico.wordpress.com/mentions/raidentrance/>)

Contacto:raidentrance@gmail.com

Anuncios

**Earn money from
your WordPress site**

WordAds

START EARNING

[Report this ad](#)

AUTOMATTIC

**We're hiring PHP
developers anywhere in
the world. Join us!**

APPLY



[Report this ad](#)

