

Blog sobre Java EE

Estás aquí: [Inicio](#)/[Java SE](#)/[Java 8](#)/Java Collections Remove con Java 8

Java Collections Remove con Java 8

10 octubre, 2017 por [Cecilio Álvarez Caules](#) — [Deja un comentario](#)

El concepto de Java Collections Remove es muy simple. Este interface soporta una serie de métodos para eliminar elementos de una colección. Los dos métodos más utilizados son :

1. **remove (index)**
2. **remove(Object)**

Ambos métodos nos permiten eliminar un elemento de una colección concreta . Vamos a ver un ejemplo de toma de contacto con la clase libro:

```
1  package com.arquitecturajava;
2
3  public class Libro {
4
5      private String titulo;
6      private int paginas;
7      private String categoria;
8      public String getTitulo() {
9          return titulo;
10     }
11     public void setTitulo(String titulo) {
12         this.titulo = titulo;
13     }
14     public int getPaginas() {
15         return paginas;
16     }
17     public void setPaginas(int paginas) {
18         this.paginas = paginas;
19     }
20     public String getCategoria() {
21         return categoria;
22     }
23 }
```

ARQUITECTURA JAVA

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](#)

```

35         int result = 1;
36         result = prime * result + ((categoria == null) ? 0 : categoria.hash
37         result = prime * result + paginas;
38         result = prime * result + ((titulo == null) ? 0 : titulo.hashCode()
39         return result;
40     }
41     @Override
42     public boolean equals(Object obj) {
43         if (this == obj)
44             return true;
45         if (obj == null)
46             return false;
47         if (getClass() != obj.getClass())
48             return false;
49         Libro other = (Libro) obj;
50         if (categoria == null) {
51             if (other.categoria != null)
52                 return false;
53         } else if (!categoria.equals(other.categoria))
54             return false;
55         if (paginas != other.paginas)
56             return false;
57         if (titulo == null) {
58             if (other.titulo != null)
59                 return false;
60         } else if (!titulo.equals(other.titulo))
61             return false;
62         return true;
63     }
64 }
65 }

```

Java Collections Remove Clásico

Vamos a crear una lista de libros y eliminar varios:

```

1  package com.arquitecturajava;
2
3  import java.util.ArrayList;
4  import java.util.Arrays;
5  import java.util.List;
6
7  public class Principal {
8
9      public static void main(String[] args) {
10
11         Libro l1 = new Libro("java", 500, "programacion");
12         Libro l2 = new Libro("angular", 300, "web");
13         Libro l3 = new Libro("react", 350, "web");
14

```

ARQUITECTURA JAVA

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](#)

```
27  
28     System.out.println(libros);  
29 }  
30 }
```

Acabamos de eliminar todos los libros . El primero usando remove y pasando un objeto , el segundo eliminando por posición y el tercero eliminando con removeAll, todo es correcto.

remove(index)

posicion

remove(object)

objeto con equals

removeAll()

eliminar todos

.Sin embargo en muchos casos necesitamos eliminar **un subconjunto de elementos aplicando una condicion**. ¿Cómo podemos hacer esto?. Hasta ahora siempre se ha utilizado **el interface Iterator**.

```
1  Iterator<Libro> it= libros.iterator();  
2  
3  while (it.hasNext()) {  
4  
5      if (it.next().getCategoria().equals("web")) {  
6          it.remove();  
7      }  
8  }  
9
```

Java Collections Remove con Java 8

A mi a nivel personal nunca me ha convencido demasiado, **pero era la solución existente**. A partir de Java 8 tenemos otra opción ya que podemos utilizar **expresiones lambda a través del método removeIf** que aparece en el interface de Collection .

ARQUITECTURA JAVA

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

método para eliminar
usando lambda

Así pues la o



d in

24



1

```
1 | libros.removeIf(l->l.getCategoria().equals("web"));
```

De esta forma eliminaremos los elementos que se encuentran dentro de la categoría “web” de una forma más elegante.

Otros artículos relacionados:

1. [Java 8 Lambda Expressions \(I\)](#)
2. [Java Predicate Interface y sus métodos](#)
3. [Java 8 Lambda y forEach \(II\)](#)
4. [Java Collections](#)



in

24



1



25
COMPART

Archivada en: [Java 8](#)

Deja un comentario

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

Comentario

ARQUITECTURA JAVA

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

Nombre *

Correo electrónico *

Web

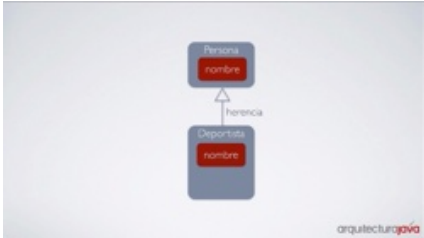
PUBLICAR COMENTARIO

BUSCAR

Buscar en este sitio ...

Mis Cursos de Java Gratuitos

Java Herencia



Java JDBC



ARQUITECTURA JAVA

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](#)

POPULAR

[Java Collections Remove con Java 8](#)

[JavaScript Console log y el manejo de Objetos](#)

[Mis Cursos de Java para desarrolladores](#)

[Spring Boot WAR sin Microservicios](#)

[Spring 5 Hello World](#)

[Cursos de Java cupones de descuento ,packs y mini curso gratuito](#)

[Java Herencia vs Interfaces](#)

[El concepto de Java Package Encapsulation](#)

[Spring Security Annotation y su configuración](#)

[JPA DTO \(Data Transfer Object\) y](#)

ARQUITECTURA JAVA

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](#)

[Usando Java Session en aplicaciones web](#)

[Java Iterator vs ForEach](#)

[Java Constructores this\(\) y super\(\)](#)

[Ejemplo de JPA , Introducción \(I\)](#)

[Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)

[Usando el patron factory](#)

[Java Collections Remove con Java 8](#)

[¿Qué es Gradle?](#)

[¿Cuales son las certificaciones Java?](#)
