

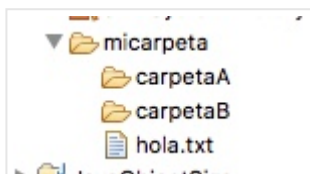


Navigation ☰

## Java List Directory en Java 8 con Streams

por **Cecilio Álvarez Caules** sobre 8 Septiembre, 2017 en **Java Lambda y Streams**

El uso de Java List Directory es muy común . Probablemente será una de las búsquedas de google más habituales en el mundo Java. Recordemos que el api de Java IO esta soportado desde la primera versión de Java por lo tanto el código para realizar esta operación es un clásico. Supongamos que disponemos de una carpeta con los siguiente ficheros y queremos listarlos por la consola.



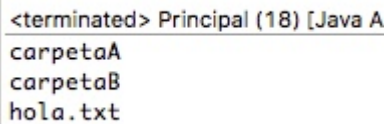
Para realizar un listado de forma clásica usaremos el siguiente bloque de código:

```
1 package com.arquitecturajava;
2
3 import java.io.File;
4
5 public class Principal {
6
7     public Principal() {
8         // TODO Auto-generated constructor stub
9     }
10
11     public static void main(String[] args) {
12
13         File micarpeta= new File("micarpeta");
14
15         File[] listaFicheros=micarpeta.listFiles();
16
17         for (int i=0;i<listaFicheros.length;i++) {
18
19
20
21
22             System.out.println(listaFicheros[i].getName());
23
24         }
```

```

25
26     }
27
28 }
```

El resultado se imprime en la consola:



```

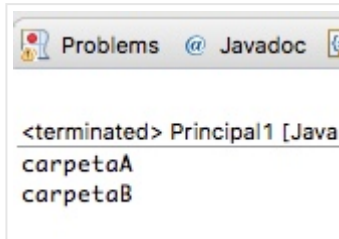
<terminated> Principal (18) [Java A
carpetaA
carpetaB
hola.txt
```

De la misma forma podemos hacer algo similar y quedarnos únicamente con el listado de carpetas apoyándonos **en el método isDirectory de la clase File**.

```

1  File micarpeta= new File("micarpeta");
2  File[] listaFicheros=micarpeta.listFiles();
3
4  for (int i=0;i<listaFicheros.length;i++) {
5
6
7      if (listaFicheros[i].isDirectory()) {
8
9          System.out.println(listaFicheros[i].getName());
10     }
11 }
12
```

El resultado solo muestra las carpetas:



```

<terminated> Principal1 [Java
carpetaA
carpetaB
```

## Java List Directory Streams

El código es razonable. Ahora bien **¿Se podría hacer de una forma más elegante utilizando Java8?**  
 .La verdad es que sí , **sin ninguna duda**. Vamos a verlo:

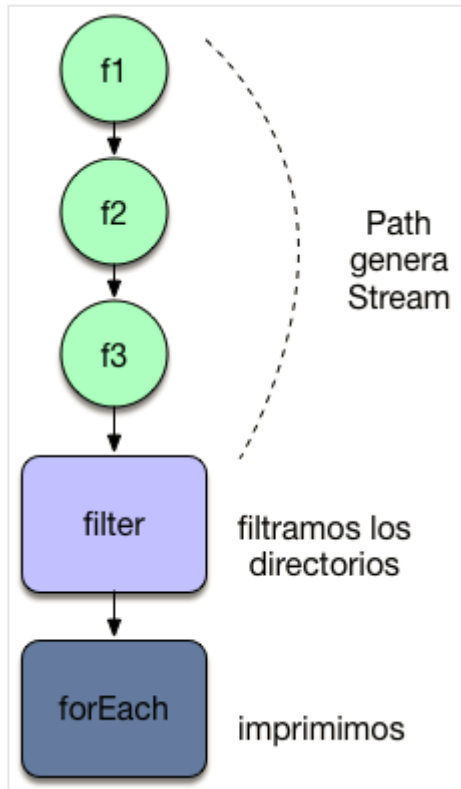
```

1  package com.arquitecturajava;
2
3  import java.io.IOException;
4  import java.nio.file.Files;
5  import java.nio.file.Paths;
6
7  public class Principal2 {
8
9      public static void main(String[] args) throws IOException {
10
11
12          Files.list(Paths.get("micarpeta")).forEach(System.out::println);
13      }
14 }
```

El concepto de **Java List Directory** se simplifica claramente y el resultado es idéntico.

```
<terminated> Principal (18) [Java A  
carpetaA  
carpetaB  
hola.txt
```

Acabamos de usar **un Stream de Java 8 para recorrer un directorio**. De la misma forma podríamos utilizar el método `filter` y un predicado para filtrar los elementos del directorio y mostrar únicamente las carpeta.



Veamoslo:

```
1 package com.arquitecturajava;  
2  
3 import java.io.IOException;  
4 import java.nio.file.Files;  
5 import java.nio.file.Paths;  
6  
7 public class Principal2 {  
8  
9     public Principal2() {  
10         // TODO Auto-generated constructor stub  
11     }  
12  
13     public static void main(String[] args) throws IOException {  
14  
15         Files.list(Paths.get("micarpeta")).filter(Files::isDirectory).forEach(S  
16     }  
17  
18  
19 }
```

Con estos ejemplos podemos ver como Java 8 simplifica de forma **clara alguna de las operaciones más clásicas de Java IO**.

Otros artículos relacionados:

1. [Java Stream Filter y Predicates](#)
2. [Java 8 Lambda Expressions \(I\)](#)
3. [Java Lambda reduce y wrappers](#)
4. [Java Stream String y Java 8](#)
5. [Java Streams Oracle](#)

It's only fair to share...    

## Subscribe

Síguenos en LinkedIn y Twitter o suscríbete al RSS.

### Related Posts:

[Java Herencia vs Interfaces](#)  
[Spring Security Annotation y su configuración](#)  
[Java Stream Filter y Predicates](#)  
[Java Herencia y sus limitaciones](#)  
[Java Stream String y Java 8](#)

◀ Cursos de Java cupones de descuento ,packs y mini curso gratuito

**Sin comentarios todavía.**

Deja un comentario

Name (required)

Email (will not be published) (required)

Website

[Submit Comment](#)

## Buscar

## Translate:

Con la tecnología de  Traductor de Google

POPULAR

LATEST



### Mis Cursos de Java para desarrolladores

27 JULIO, 2017



### Command Pattern en Java y la gestion de tareas

26 MAYO, 2017



### Spring @import , organizando Spring framework

9 AGOSTO, 2017



### Java LinQ con JinQ y Java Persistence API

17 MAYO, 2017



### Cursos de Java cupones de descuento ,packs y mini curso gratuito

7 SEPTIEMBRE, 2017



### Java Herencia vs Interfaces

1 SEPTIEMBRE, 2017



### El concepto de EJB Event y como desacoplar servicios

25 JULIO, 2017

**JPA Database Schema y automatización**

31 MAYO, 2017

**Java 8 Factory Pattern y su implementación**

26 JUNIO, 2017

**Spring Security Annotation y su configuración**

30 AGOSTO, 2017

## POPULAR

**¿Cuales son las certificaciones Java?**

16 ENERO, 2015

**¿Tiene futuro JSF?**

21 ABRIL, 2015

**Libro:Arquitectura Java JPA (Domain Driven Design)**

8 MAYO, 2014

**Ejemplo de JPA , Introducción (I)**

17 SEPTIEMBRE, 2013

**Introducción a EJB 3.1 (I)**

2 AGOSTO, 2013

**JPA (III) EntityManager métodos**

4 OCTUBRE, 2013

**Usando Java Session en aplicaciones web**

15 OCTUBRE, 2014

**El futuro Java EE 7 vs Spring Framework**

6 MAYO, 2015

**Java 8 Lambda Expressions (I)**

15 JULIO, 2014

**Libro: Arquitecturas Web y Angular js**

8 SEPTIEMBRE, 2014

## Contacto

[contacto@arquitecturajava.com](mailto:contacto@arquitecturajava.com)

## Redes Sociales

Síguenos en LinkedIn y Twitter o suscríbete al RSS.

© 2017 Arquitectura Java. All Rights Reserved.

Diseñado por [Clickea](#)