

Java 8: Ms funcional que nunca

Vctor Orozco

Nabenik

April 16, 2016



Vector Orozco

- ▶ Developer (JVM/Open Source Advocate)
- ▶ Ex-JUG Leader
- ▶ Consultor independiente (Nabenik)
- ▶ @tuxtor
- ▶ The J*



Java 8



JAVA 2015
DAY
Guatemala

Java 8

<https://www.oracle.com/java8>
<https://www.oracle.com/java8launch>



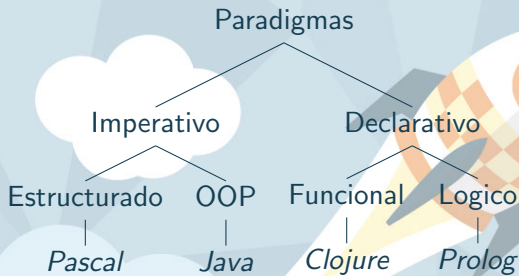
JAVA 2015
DAY
Guatemala

Java 8


- ▶ Nashorn
- ▶ Date/Time API
- ▶ Compact Profiles
- ▶ Type Annotations
- ▶ **Default methods**
- ▶ **Streams**
- ▶ **Lambda Expressions**



Paradigmas (Simplificacin)

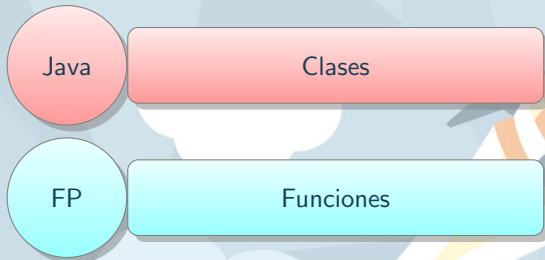


Programacin funcional

- ▶ Computacin = Evaluacin de funciones matemticas (cálculo de lambdas)
 - ▶ NO cambios en estado
 - ▶ NO mutar datos
 - ▶ Declarativo → Expresiones
- 



Java vs. Funcional (organizacin)



Java vs. Funcional (algoritmos)

Java

Imperativo, comportamiento
como una serie de pasos

FP

Declarativo, interaccin de fun-
ciones sin especificar su contenido



Java vs. Funcional (Mutabilidad y estado)

Java

Estado y comportamiento
juntos, promueve mutabilidad

FP

Evita estado, pro-
mueve inmutabilidad



Java vs. Funcional (Estilo)

Java

OOP + Patrones para abstracciones de alto nivel

FP

Es una abstracción en alto nivel por sí mismo



Java vs. Funcional (Concurrencia)

Java

Concurrencia basica con
locks y recursos compartidos

FP

Workflows paralelos sin es-
tado compartido (no locks!)



Java vs. Funcional (Codigo)

Java

Descriptivo (demasiado)

FP

Conciso y denso



Java 8

Un lenguaje de programación orientada a objetos con azúcares sintácticos funcionales.



Porque programacin funcional?

- ▶ Paralelismo!
- ▶ Multicore, multicpu
- ▶ Elegancia



- # Porque programacin funcional?
- ▶ Paralelismo!
 - ▶ Multicore, multicpu
 - ▶ Elegancia
- 
- 
- 
- 



Programacin funcional en Java 8

- ▶ Java no es un lenguaje funcional puro (Clojure)
- ▶ Otras opciones JVM (Scala, Kotlin, Ceylon)
- ▶ Java soporta programacin funcional a travs de bibliotecas



-

Bloques



Bloques funcionales en Java 8

- ▶ Interfaces funcionales
- ▶ Referencia a funciones
- ▶ Lambdas
- ▶ Funciones predefinidas en Java 8 (`java.util.function`)
- ▶ Streams API



- # Bloques funcionales en Java 8
- ▶ Interfaces funcionales
 - ▶ Referencia a funciones
 - ▶ Lambdas
 - ▶ Funciones predefinidas en Java 8 (`java.util.function`)
 - ▶ Streams API
- 
- 
- 
- 



Interfaces funcionales

- ▶ Solo un mtodo abstracto
- ▶ Interfaces ahora permiten default methods

```
^^I@FunctionalInterface
^^Ipublic interface Runnable
^^I{
^^I^^Ipublic abstract void Run();
^^I}
^^I
```



- # Interfaces funcionales

 - ▶ Solo un mtodo abstracto
 - ▶ Interfaces ahora permiten default methods

```
^^I@FunctionalInterface
^^Ipublic interface Runnable
^^I{
^^I^^Ipublic abstract void Run();
^^I}
^^I
```



Interfaces funcionales

- ▶ Solo un mtodo abstracto
- ▶ Interfaces ahora permiten default methods

```
^^I@FunctionalInterface
^^Ipublic interface Runnable
^^I{
^^I^^Ipublic abstract void Run();
^^I}
^^I
```





Referencias a funciones

- ▶ Permiten utilizar una función dentro de una expresión lambda
- ▶ Permiten invocar métodos existentes



- # Referencias a funciones
- ▶ Permiten utilizar una función dentro de una expresión lambda
 - ▶ Permiten invocar métodos existentes
- 
- 
- 
- 



Expresion lambda

- ▶ Funcin annima sin asociar a un identificador
- ▶ Usadas para pasar comportamiento a funciones high-order
- ▶ Usadas para construir el resultado de una funcin high-order que necesita retornar una funcin



- # Expresion lambda
- ▶ Funcin annima sin asociar a un identificador
 - ▶ Usadas para pasar comportamiento a funciones high-order
 - ▶ Usadas para construir el resultado de una funcin high-order que necesita retornar una funcin
- 
- 
- 
- 



Expresion lambda (anatomia)

(parametros) → comportamiento

```
^^I(Integer i) -> {System.out.println(i);};
```

```
^^Ii -> System.out.println(i);
```

```
^^Ii -> i*2;
```

```
^^I
```



Streams API



Funciones predefinidas

- ▶ Java soporta programación funcional con bibliotecas
- ▶ Más de 40 interfaces funcionales en Java 8
- ▶ Raramente se deben crear nuevas
- ▶ Streams API



- # Funciones predefinidas
- ▶ Java soporta programación funcional con bibliotecas
 - ▶ Más de 40 interfaces funcionales en Java 8
 - ▶ Raramente se deben crear nuevas
 - ▶ Streams API
- 
- 
- 
- 



Streams API - Listas

```
List<String> jedis =  
    ^IArrays.asList("ObiWan", "Anakin", "Yoda");  
  
jedis.forEach(s -> System.out.println(s);
```



Streams API - Listas

```
List<String> jedis =  
    ^IArrays.asList("ObiWan", "Anakin", "Yoda");  
  
jedis.forEach(System.out::println);
```



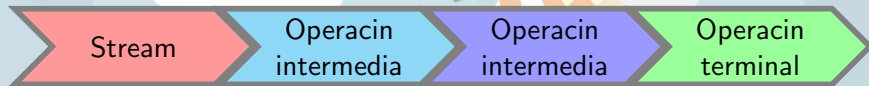
Streams API - Operaciones con predicados

```
List<String> jedis =  
    ^IArrays.asList("ObiWan", "Anakin", "Yoda");  
  
Predicate<String> darthRemover =  
    ^IIs -> "Anakin".equals(s);  
  
jedis.removeIf(darthRemover);
```



Streams API

- ▶ Map-Reduce
- ▶ Monads = Serie de pasos / funciones anidadas



Streams API - Map-reduce

```
winners
^^I.stream() //Steam
^^I.filter( //Intermedio +- MAP
^^I^^Ip -> p.getOrderId()
^^I^^I^^I.equals(winner.getOrderId()))
^^I^^I)
^^I.findFirst(); //Reduce
```



Streams API - Map-reduce

```
winners
^^I.stream() //Steam
^^I.filter( //Intermedio +- MAP
^^I^^Ip -> p.getOrderId()
^^I^^I^^I.equals(winner.getOrderId())
^^I^^I)
^^I.collect(Collectors.toList()); //Reduce - collector
```



Ejemplo

<http://github.com/tuxtor/fpjavademo>



Programacin funcional - Bueno

- ▶ Divertido
- ▶ Declarativo
- ▶ Menos cdigo, cdigo ms legible



Programacin funcional - Malo

- Performance - invokedynamic (debatible)
- Debug
- Flexibilidad



- # Programacin funcional - Malo
- Performance - invokedynamic (debatible)
 - Debug
 - Flexibilidad
- 
- 
- 
- 



Lecturas recomendadas

- ▶ JDK 8 Lamdas & Streams MOOC -
<https://www.youtube.com/playlist?list=PLMod1hYilvSZL1xclvHcsV>
- ▶ Functional Programming in Java: Harnessing the Power Of
Java 8 Lambda Expressions -
<http://www.amazon.com/Functional-Programming-Java-Harnessing-Expressions/dp/1937785467>



- # Lecturas recomendadas
- ▶ JDK 8 Lamdas & Streams MOOC -
<https://www.youtube.com/playlist?list=PLMod1hYilvSZL1xclvHcsV>
 - ▶ Functional Programming in Java: Harnessing the Power Of
Java 8 Lambda Expressions -
<http://www.amazon.com/Functional-Programming-Java-Harnessing-Expressions/dp/1937785467>
- 
- 
- 
- 



Gracias

- ▶ me@vorozco.com
- ▶ <http://vorozco.com>
- ▶ <http://github.com/tuxtor/slides>



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Guatemala License.

