

# Programación funcional para la JVM

---

Víctor Orozco - @tuxtor

1 de julio de 2017

Nabenik

# Acerca de



¿Como me salto la teoria  
de la computación para  
programar  
funcionalmente? -  
Estudiante 1

¿Como me salto la teoria  
de la computación para  
programar  
funcionalmente? -  
Developer 1

$$\begin{aligned}x[t/x] &= t \\y[t/x] &= y \text{ if } x \neq y \\c[t/x] &= c \\(s_1 s_2)[t/x] &= s_1[t/x] s_2[t/x] \\(\lambda x. s)[t/x] &= \lambda x. s \\(\lambda y. s)[t/x] &= \lambda y. (s[t/x]) \text{ if } x \neq y \text{ and either } x \notin FV(s) \text{ or } y \notin FV(t) \\(\lambda y. s)[t/x] &= \lambda z. (s[z/y][t/x]) \text{ otherwise, where } z \notin FV(s) \cup FV(t)\end{aligned}$$

The only difference is in the last two lines. We substitute as before in the two safe situations where either  $x$  isn't free in  $s$ , so the substitution is trivial, or where  $y$  isn't free in  $t$ , so variable capture won't occur (at this level). However where these conditions fail, we first rename  $y$  to a new variable  $z$ , chosen not to be free in either  $s$  or  $t$ , then proceed as before. For definiteness, the variable  $z$  can be chosen in some canonical way, e.g. the lexicographically first name not occurring as a free variable in either  $s$  or  $t$ .<sup>3</sup>

¿Como aprendo  
programación funcional?

¿Que es y como  
implemento programación  
funcional?

# Outline

Java 8

FP

Bloques

Demo Java y Kotlin

Fin



# Java 8

---

# Java 8

2014-03-18 - 3 años!!

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

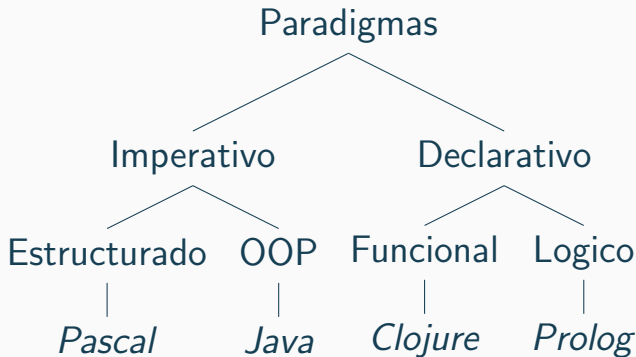
<https://www.oracle.com/java8launch>

# Java 8

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



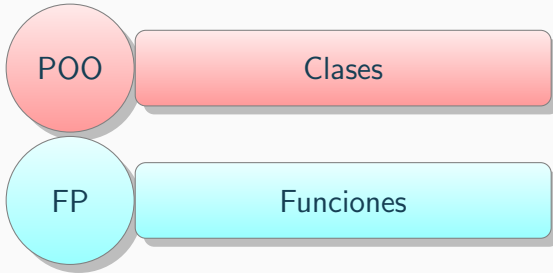
# Paradigmas (Simplificación)



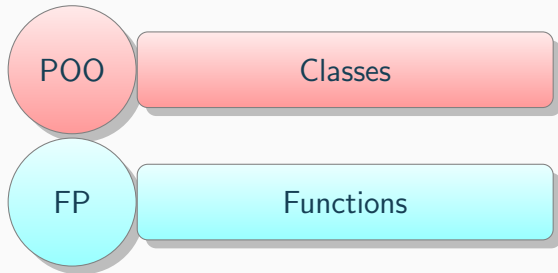
# Programación funcional

- Computación = Evaluación de funciones matemáticas (calculo de lambdas)
- NO cambios en estado
- NO mutar datos
- Declarativo  $\rightarrow$  Expresiones

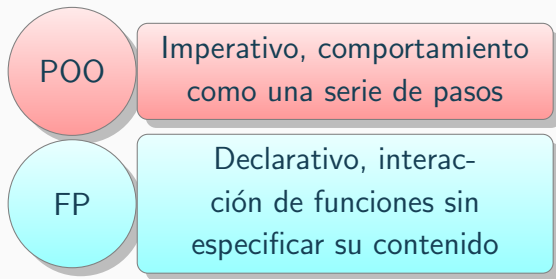
# PОО vs. Funcional (organización)



# PОО vs. Functional (organization - think about)

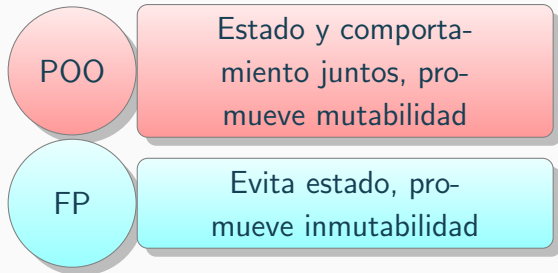


# PОО vs. Funcional (algoritmos)





# PОО vs. Funcional (Mutabilidad y estado)



# PОО vs. Funcional (Estilo)

PОО

OOP + Patrones para  
abstracciones de alto nivel

FP

Es una abstracción en  
alto nivel por si mismo

# PОО vs. Funcional (Concurrencia)

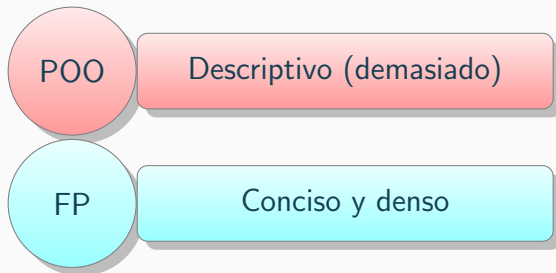
PОО

Concurrencia basica con  
locks y recursos compartidos

FP

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

# P OO vs. Funcional (Código)



**FP**

---

# ¿Porque programación funcional?

- Paralelismo
- Multicore, multicpu
- Elegancia

# Programación funcional en la JVM

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

# Java 8

Un lenguaje de programación multiparadigma con presencia fuerte de POO con complementos funcionales y orientados a aspectos





# Kotlin

Un lenguaje de programación multiparadigma con presencia fuerte de programación funcional con complementos POO



# 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 Kotlin

- Funciones y referencia a funciones
- Sintaxis Kotlin
- Lambdas
- Funciones predefinidas en Kotlin
- Streams API

# Streams API

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



# Demo Java y Kotlin

---

**Fin**

---

# Gracias

- [me@vorozco.com](mailto: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.