S4N

SEATTLE / SÃO PAULO / PANAMÁ
BOGOTÁ / MEDELLÍN

PROGRAMACIÓN FUNCIONAL USANDO JAVA

Por Laura Camacho

Medellín - 11 de junio de 2020

S4N

Agenda

¿Por qué esta charla?

- ¿Por qué programación funcional?
- ¿Por qué programación funcional en Java?
- Conceptos básicos

Funciones de primera clase

- ¿Qué es una función?
- ¿Qué son funciones de primera clase?
  - *Funciones anónimas*
  - *Funciones de orden superior*
- ¿Por qué son importantes?
- ¿Cómo lograrlo en Java?

Inmutabilidad

- Mutabilidad / Inmutabilidad
- ¿Cómo lograrlo en Java?

Cierre

Preguntas

01

¿Por qué esta charla?

¿Por qué programación funcional?

- Está basado en definiciones matemáticas
- Nos ayuda a escribir código más formal con el que lograremos, entre otras cosas:
  - Mantenibilidad
  - Escalabilidad
- Es necesario aprender un nuevo método de solución de problemas

¿Por qué programación funcional en Java?

- La mayoría de las personas que utilizan Java lo hacen en el estilo imperativo y orientado a objetos
- Aplicar correctamente nuevas características que provee el lenguaje
- Podemos sacar provecho del diseño orientado a objetos en una construcción funcional
- Hacer software de calidad

S4N

Conceptos básicos

- **Funciones de primera clase (First-class functions)**
- Funciones puras (Pure functions)
- Recursividad (Recursion)
- **Variables inmutables (Immutable variables)**
- Evaluación no estricta (Nonstrict evaluation)
- Declaraciones (Statements)
- Coincidencia de patrones (Pattern matching)

02

Funciones de primera clase

¿Qué es una función?

Convenciones Parámetros Nombre Cuerpo Retorno

¿Qué son funciones de primera clase (First-class functions)?

- Son funciones que son tratadas como objetos
  - Las podemos pasar como parámetro a otra función
  - Una función puede ser retornada por otra función
  - Almacenarlas en una variable

Funciones anónimas (Anonymous Functions)

Funciones con alcance limitado, que necesitan existir por un corto periodo de tiempo

- **Funciones lambda (Lambda Functions)**
  Son funciones sin nombre

- **Closures**
  Son funciones sin nombre que referencian variables que están fuera del alcance de la función

Funciones de orden superior (Higher-Order Functions)

Una función es de orden superior si recibe como parámetro una función y/o la retorna

¿Por qué son importantes?

¡Podemos **componer** funciones!

$$f(x) \text{ y } g(x)$$

$$(g \circ f)(x)$$

$$g[f(x)]$$

¿Cómo lograrlo en Java?

- **Método:**
  - Suite de pruebas unitarias
    - JDK 8
    - Junit 4
- **Caso de estudio:**
  - Preparar alimentos fritos

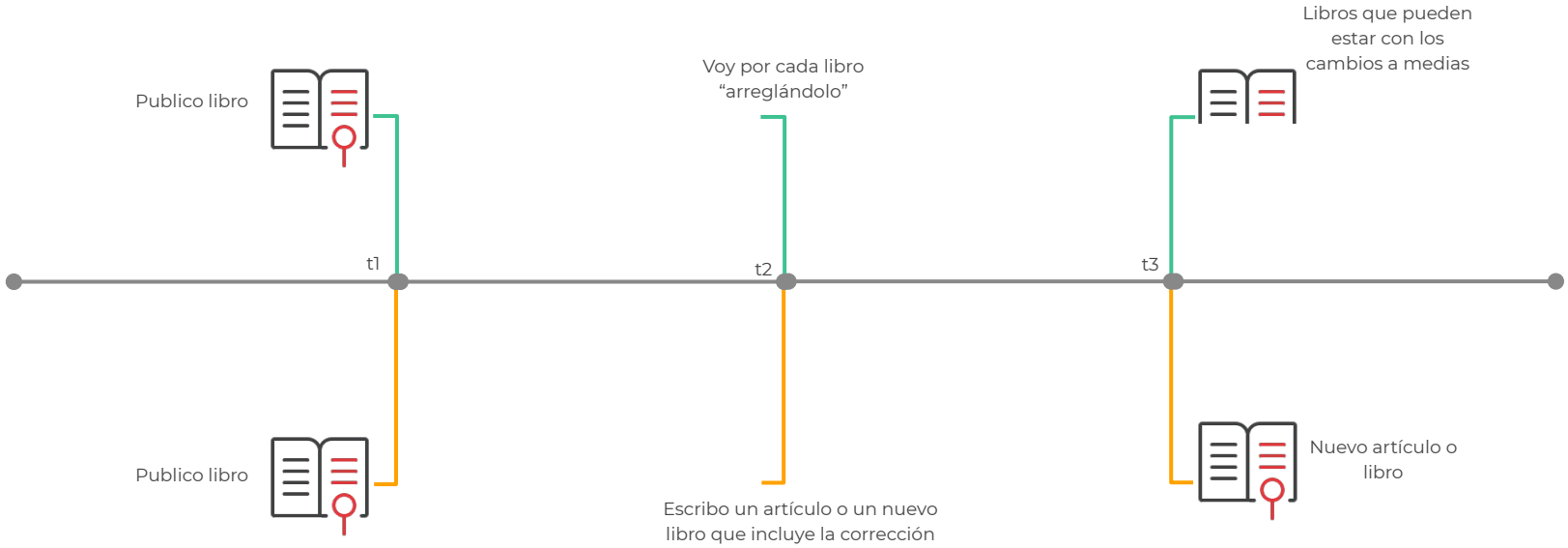
En resumen

- Las funciones de primera clase son funciones que son tratadas como objetos
- Algunos de sus subconjuntos son las funciones anónimas y funciones de orden superior
- La composición de funciones es extremadamente poderosa

03

Inmutabilidad

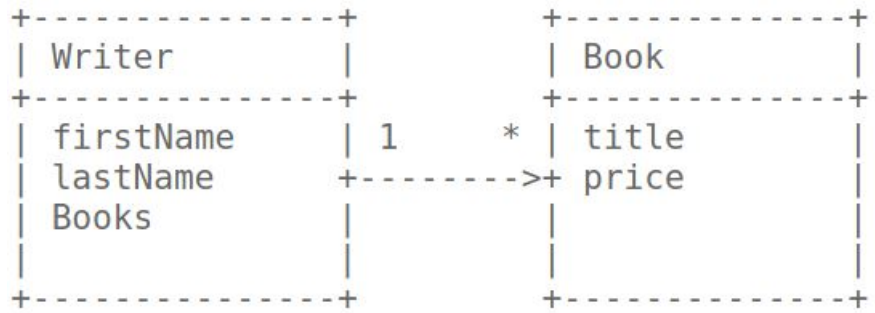
Mutabilidad / Inmutabilidad



Convenciones Mutable Inmutable Línea de tiempo

¿Cómo lograrlo en Java?

- **Método:**
  - Suite de pruebas unitarias
    - JDK 8
    - Junit 4
- **Caso de estudio:** Escritor y sus libros



En resumen

- Los objetos inmutables no cambian su estado interno en el tiempo
- Cambiar la referencia de una variable se convierte en un proceso atómico
- Los objetos inmutables se pueden compartir de manera segura entre múltiples subprocesos y además son libres de efectos secundarios

04

Cierre

En resumen

- **Funciones de primera clase (First-class functions)**
- **Variables inmutables (Immutable variables)**
- Funciones puras (Pure functions)
- Recursividad (Recursion)
- Evaluación no estricta (Nonstrict evaluation)
- Declaraciones (Statements)
- Coincidencia de patrones (Pattern matching)

- Entender cuál es el problema que necesitamos solucionar
- Entender el método de solución que estamos aplicando y porqué lo estamos aplicando.
- ¡La herramienta al final del día no es lo importante! ¯_(ツ)_/¯
- Necesitamos garantizar calidad en los productos que hacemos

S4N

Referencias

- Libros:
  - Becoming Functional - Joshua Backfield
  - Functional Thinking - Neal Ford
  - Java 8 in Action - Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft
- Papers:
  - Why Functional Programming Matters - John Hughes
  - Can Programming Be Liberated From the von Neumann Style? A Functional Style and Its Algebra of Programs - John Backus
- Librerías:
  - Immutables
  - Vavr (la primera versión se llamaba javaslang)

- Links:
  - https://www.baeldung.com/java-immutable-object
  - https://allegro.tech/2020/04/immutability-in-java.html
  - http://cr.openjdk.java.net/~briangoetz/lambda/sotc3.html
  - http://openjdk.java.net/projects/lambda/
  - https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html
  - https://docs.oracle.com/javase/tutorial/essential/concurrency/immutable.html

¡Muchas gracias!

S4N | Laura Camacho
lauracamacho@s4n.co

¿Preguntas?