Scala

Java

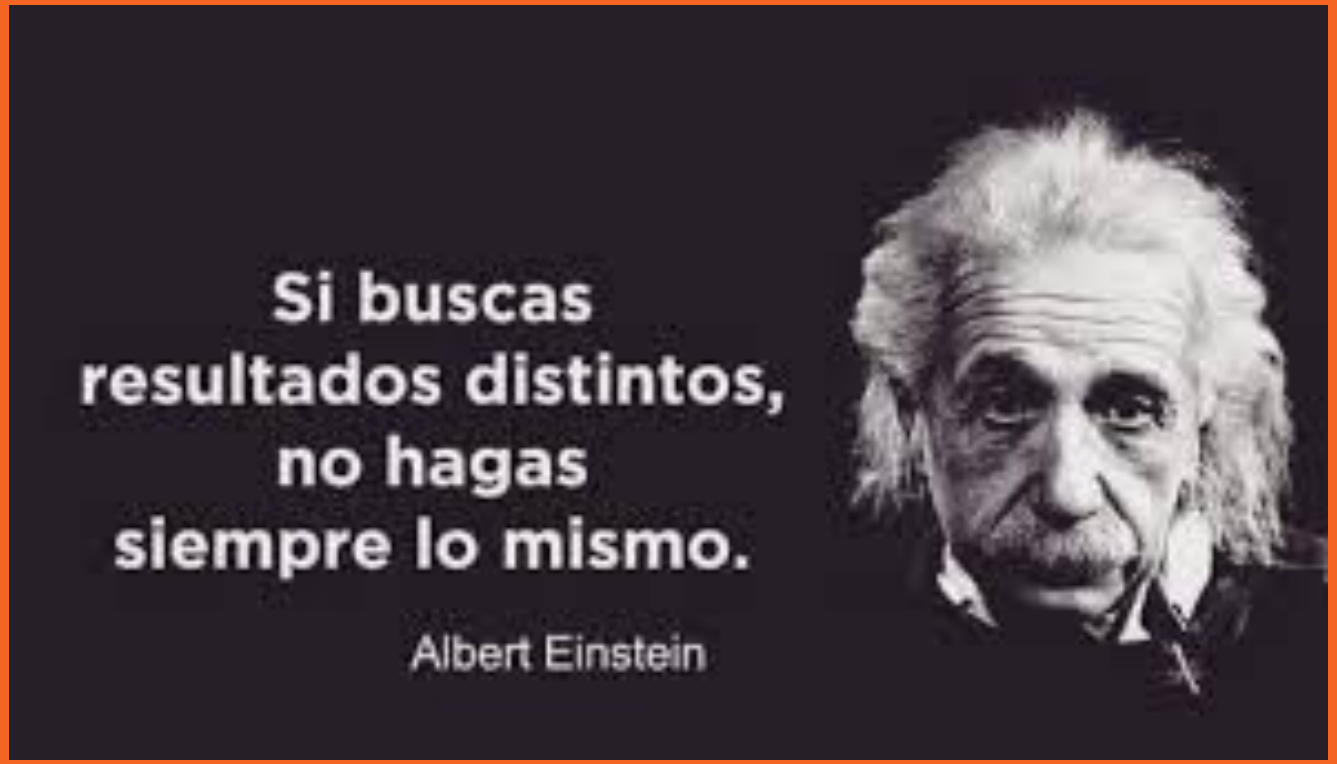
Plastilina vs Rastis

Scala

Scala over Java

Scala

Java


Si buscas
resultados distintos,
no hagas
siempre lo mismo.
Albert Einstein

expressions

statements

Las expresiones

- Son declarativas y devuelven un valor
- Se pueden combinar
- Son inmutables e idempotentes
- Su evaluación puede ser reordenada por el compilador
- Tienen transparencia referencial (lazy evaluation, substitution model (pensar como ecuaciones))
- Bloques (closures)

funcional

imperativo

- Una función solo tiene una responsabilidad
- Una función no tiene efecto de lado
- Una función tiene transparencia referencial
- Una función puede ser fácilmente paralelizable
- Higher Order Functions
- Función Partida (o parcial)
- Inmutabilidad
- Types, Monadas, Aplicativos, Functores, etc

Functor

- Es un patrón de diseño funcional
- Caja que lleva algo adentro que soporta poder cambiar el tipo de lo que lleva adentro
- Todo tipo que implemente map es un functor
- Por que necesitamos saber esto?
  - List, Future, Option, Either son functores

Functor

MAP

.map(transformarAPerro)

Un Functor es una monada :D


SAY MONAD
ONE MORE TIME
memegenerator.net

Monads

- Functor hereda de Monada
- Agrega operación de
  - flatMap
- Por que necesitamos saber esto?
  - List, Future, Option, Either son monadas
- En resumen, le pusieron nombre a operaciones que se usan frecuentemente en distintos tipos de datos cuyo comportamiento es el mismo (patrón de diseño)

inmutabilidad

mutabilidad

- “El ser es inmutable” (Parmenides dixit)
- Simplifica los problemas donde la concurrencia y el paralelismo son requeridos
- Algebraic Data Types (case classes)

scala syntax

java syntax

- class
- object (as singleton)
- companion object (apply como factory method)
- trait
- case class
- match case
- for comprehension + yield
- implicit
- type alias

pattern match

polimorfismo

- ‘match case’ como un switch con esteroides
- ‘unapply’ detrás de las cortinas
- exhaustive match case en tiempo de compilación
- polimorfismo cuando realmente tenga sentido

Errors as Values

throw Exceptions

- Either[Left[Throwable]],Right[T]]
- Try[Success[T], Failure[Throwable])
- Se manejan como cualquier monada y se puede hacer pattern match para definir que hacer de manera más expresiva

Scala Async

- Future as Monads
- Composable
- Pattern Matchable

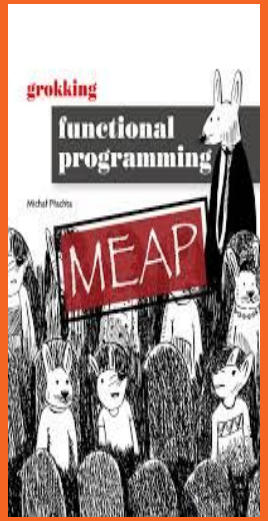
Conclusiones

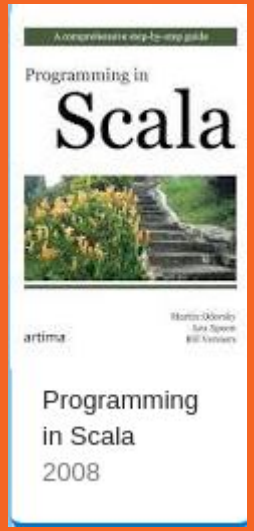
- Scala une lo mejor de 2 mundos (objetos + funcional)
- Oderksy lo escribió tomando lo mejor de haskell, python, java y smalltalk
- Expresión mata galan
- Simplifica y reduce la cantidad de código
- Te obliga a pensar y no codear como robot
- Es maleable como la plastilina
- Tiene 16 años en el mercado, está maduro
- Tiene un ecosistema activo y muy variado
- La comunidad sigue en crecimiento y cada vez hay más adeptos a nivel corporaciones (fintech)

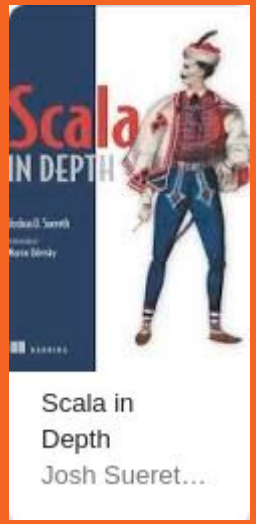
Referencias

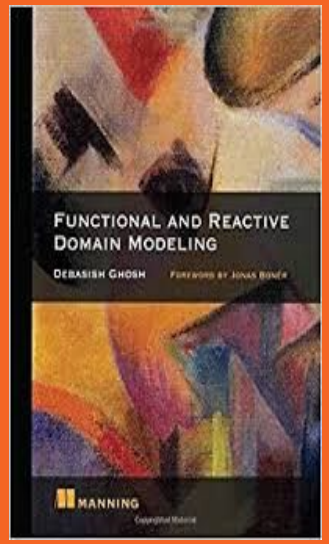
- REPO
- Scala Lang
- SBT Simple Build Tool (como maven pero sin tanta paja)
- Typelevel
- Buen blog
- Colecciones

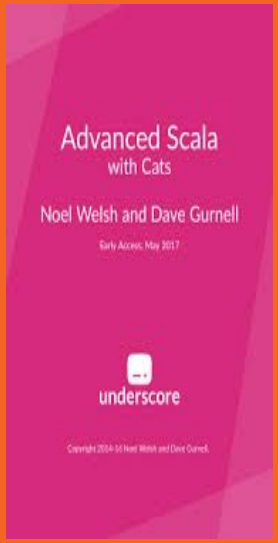
Libros recomendados











There is no try

¿Preguntas?