

Programación Funcional en Scala

– Tema 3 –

Constructores de Tipos, Type Classes y Funtores

Jesús López González
jesus.lopez@hablapps.com

Programación Funcional en Scala
Habla Computing

Cursos ETSII-URJC 2015

1 Higher Kinded Types

2 Implicits

3 Type Classes

4 Funtores

5 Takeaways

Constructor de Tipos

Un constructor de tipos es un tipo que recibe argumentos tipo para construir un nuevo tipo. Ejemplos de constructores de tipo pueden ser *List* (recibe un argumento tipo), *Option* (recibe un argumento tipo) o *Map* (recibe dos argumentos tipo).

Listing 1: Usando constructores de tipo

```
scala> val opt: Option[Int] = Some(3)
opt: Option[Int] = Some(3)

scala> val lst: List[String] = List("how", "are", "you")
lst: List[String] = List(how, are, you)

scala> val map: Map[Int, String] = Map(1 -> "one", 2 -> "two")
map: Map[Int,String] = Map(1 -> one, 2 -> two)
```

Kinds

- Proper: $(*)$ Ej: String, Int, List[Int], etc.
- First-order: $(* \rightarrow *)$ Ej: Option[_], List[_], etc.
- Higher-order: $((* \rightarrow *) \rightarrow *)$ Ej: Functor[F[_]], etc.

Listing 2: Usando constructores de tipo

```
scala> def show[C[_], _], A, B](arg: C[A, B]): String =  
  arg.toString  
show: [C[_], _], A, B](arg: C[A,B])String  
  
scala> show(Map(1 -> "one", 2 -> "two"))  
res0: String = Map(1 -> one, 2 -> two)  
  
scala> show(List(1, 2, 3))  
<console>:9: error: inferred kinds of the type arguments  
  (scala.collection.LinearSeqOptimized,Int,List[Int]) do not  
  conform to the expected kinds of the type parameters (type  
  C,type A,type B).  
scala.collection.LinearSeqOptimized's type parameters do not  
  match type C's expected parameters:  
type Repr's bounds <:  
  scala.collection.LinearSeqOptimized[A,Repr] are stricter  
  than type _'s declared bounds >: Nothing <: Any
```

1 Higher Kinded Types

2 **Implicits**

3 Type Classes

4 Funtores

5 Takeaways

1 Higher Kinded Types

2 Implicits

3 Type Classes

4 Funtores

5 Takeaways

1 Higher Kinded Types

2 Implicits

3 Type Classes

4 Funtores

5 Takeaways

1 Higher Kinded Types

2 Implicits

3 Type Classes

4 Funtores

5 Takeaways