

Programación Funcional en Scala

– Tema 2 –

Introducción a la Programación Funcional

Jesús López González
jesus.lopez@hablapps.com

Programación Funcional en Scala
Habla Computing

Cursos ETSII-URJC 2015

- 1 ¿Qué es la Programación Funcional?
- 2 Funciones en Scala
- 3 Tipos algebraicos de datos (ADTs)
- 4 Ejemplo: Plegando Listas
- 5 Takeaways

¿Qué es la Programación funcional?

Efectos de Lado

Se dice que una función tiene efectos de lado si, además de devolver un valor, modifica algún estado o interactúa visiblemente con el mundo externo.

Ejemplos

- Modificar una variable global
- Invocar un *setter*
- Lanzar una excepción
- Imprimir una traza por la consola
- Actualizar un dato en una BBDD
- Invocar un servicio web externo

¿Qué es la Programación funcional?

Paradigma Funcional y Pureza

La programación funcional se basa en una única premisa: programar con *funciones puras*. Una función pura es aquella que no tiene efectos de lado y que para una misma entrada siempre devolverá la misma salida (mapeo de valores).

Listing 1: Ejemplo de función pura

```
def add(a: Int, b: Int): Int = a + b
```

¿Qué es la Programación funcional?

Transparencia Referencial y Modelo de Sustitución

Se dice que una expresión es *referentially transparent* si tras reemplazarla por el valor que devuelve, no se aprecia ningún cambio en el comportamiento del programa. Esta cualidad activa un nuevo modo de evaluar nuestros programas: *el modelo de sustitución*. Y es que cuando contamos con la transparencia referencial podemos razonar sobre nuestros programas como si se tratasen de expresiones algebraicas.

Listing 2: Aplicando el modelo de sustitución sobre una expresión

```
def add(a: Int, b: Int): Int = a + b

add(add(1, 2), add(add(4, 5), 6))
add(3, add(add(4, 5), 6))
add(3, add(9, 6))
add(3, 15)
18
```

¿Qué es la Programación funcional?

Listing 3: ¿Es una función pura?

```
var lastAdd: Int = 0

def add2(a: Int, b: Int): Int = {
  val res = a + b
  lastAdd = res
  res
}

// Puedes apoyarte en el siguiente programa...

add2(1, 2)
doSomething(lastAdd)
```

¿Qué es la Programación funcional?

Listing 4: ¿Y esta función?

```
def add3(a: Int, b: Int): Int = {  
  println(s"Adding $a and $b")  
  a + b  
}
```

Listing 5: ¿Y ésta otra?

```
def add4(a: Int, b: Int): Int = {  
  var res = a + b  
  res = res - 1  
  res = res + 1  
  res  
}
```

¿Qué es la Programación funcional?

¿Por qué preocuparnos por el Paradigma Funcional?

- Eficiencia / Escalabilidad (multi-cores, Big Data)
- Modularidad (reusabilidad, composición...)
- Testability (property-based testing)
- Comprensibilidad (estructura, funcionamiento, depuración...)

¿Qué es la Programación funcional?

¿Y por qué preocuparnos por Scala?

- Multiparadigma e Impuro (*Scala as a better Java*)
- Lenguaje JVM estáticamente tipado
- Más que un lenguaje, un ecosistema (Play, Akka, Spark...)

Listing 6: Hola Mundo en Scala

```
object HolaMundo {  
  def main(args: Array[String]): Unit = {  
    println("¡Hola Mundo!")  
  }  
}
```

- 1 ¿Qué es la Programación Funcional?
- 2 **Funciones en Scala**
- 3 Tipos algebraicos de datos (ADTs)
- 4 Ejemplo: Plegando Listas
- 5 Takeaways

- 1 ¿Qué es la Programación Funcional?
- 2 Funciones en Scala
- 3 Tipos algebraicos de datos (ADTs)**
- 4 Ejemplo: Plegando Listas
- 5 Takeaways

- 1 ¿Qué es la Programación Funcional?
- 2 Funciones en Scala
- 3 Tipos algebraicos de datos (ADTs)
- 4 Ejemplo: Plegando Listas
- 5 Takeaways

- 1 ¿Qué es la Programación Funcional?
- 2 Funciones en Scala
- 3 Tipos algebraicos de datos (ADTs)
- 4 Ejemplo: Plegando Listas
- 5 Takeaways