Universidad del Valle de Guatemala Mario Barrientos

Programación y Tecnologías Integradoras Carné 13039

Douglas Barrios 11/09/17

**Demostración Final de Scala**

**Dificultades Encontradas**

Entre las dificultades encontradas se puede destacar el adaptarse a una nueva sintaxis que, a pesar de que se parece mucho a Java, representa cierto reto llegar a dominar. En Scala se puede hacer una cosa de varias formas, sin embargo, hay cierta dificultad en poder determinar cuál es la mejor forma de hacerlo, algo que únicamente la experiencia brinda.

Por otro lado, los programadores que estamos acostumbrados a utilizar Java, estamos acostumbrados a usar ciertos tipos de estructuras de datos al momento de desarrollo. Sin embargo, implementar las mismas estructuras en Scala puede resultar muy difícil, e incluso en la mayor parte de ocasiones es mejor utilizar otras estructuras que tiene Scala (por ejemplo RDDs o DataFrames para el manejo de muchos datos de una tabla) que al principio podrá resultar difícil aprender cómo funciona cada una de ellas pero que al final resultarán en mayores ventajas desde el punto de vista de rendimiento.

Además, si ya se posee alguna experiencia programando en otros lenguajes de programación, probablemente se esté acostumbrado a utilizar los mismos procedimientos para operar datos dentro de una estructura (lista, array, etc.), sin embargo, Scala tiene muchas funciones incorporadas para operar datos entre cada estructura: desde operaciones matemáticas simples a operaciones que requieran una función definida. Llegar a encontrar estas funciones más que una dificultad fue un reto que al final de cuentas optimiza el código desarrollado. Por tanto, se recomienda hacer una lectura rápida de la documentación: http://docs.scala-lang.org/cheatsheets/index.html.

**Casos de Uso**

Scala puede ser utilizado en una gran variedad de formas y para una gran variedad de aplicaciones. Los principales casos de uso con los cuales se utiliza Scala incluyen utilizar en software con programación concurrente o que posea características de concurrencia, algo que Java no posee y no ayuda tanto como lo hace Scala. Por otro lado, Scala puede ser utilizada como herramienta para fácil manipulación de estructuras de datos, así como también el análisis de los datos independientemente del objetivo del análisis (machine learning, por ejemplo). En comparación con Java, Scala permite hacer cosas que en Java es más complicado de hacer en tan solo una línea de código, haciendo que el rendimiento sea el óptimo.

En términos generales, Scala se puede utilizar para los mismos casos de uso de Java, puede hacer todo lo que Java hace, pero normalmente con una sintaxis mucho más limpia y óptima. Entre algunos casos de uso concretos podemos encontrar:

* Entornos Big Data: para herramientas como Spark, las ventajas de utilizar Java (o incluso Python) son abrumadoras: en la versión Scala de un programa, será unas 5-10 veces más corto que el programa de Java equivalente, y será como mínimo tres veces más rápido.
* Creación de DSLs: Scala es adecuado para el desarrollo de una DSL, gracias a características tales como coincidencia de patrones, la flexibilidad sintáctica y la sobrecarga de operadores, así como también la facilidad que tiene de hacer operaciones concurrentes.

**Curva de Aprendizaje Esperada**

Con respecto a la curva de aprendizaje esperada, esta depende de la persona que empiece a utilizar esta herramienta. Si esta persona posee experiencia en Java o algún otro lenguaje de programación orientado a objetos, la curva de aprendizaje será menor en comparación a una persona que utilice Scala como primer lenguaje de programación. Es más fácil empezar a usar Scala a partir de dominar Java que viceversa. Se hacen tantas comparaciones con Java pues Scala es sintácticamente similar a Java, lo cual reduce la curva de aprendizaje inicial para los programadores.

Algo importante de destacar es que Scala es un lenguaje difícil de dominar. Un programador excelente será más productivo en Scala que en Java, pero un programador medio probablemente será menos productivo al pasar de Java a Scala (y no sólo en los primeros meses). A diferencia de java, Scala tiene una sintaxis flexible y normalmente ofrece muchas maneras de lograr el mismo resultado final. Pero esto no es sólo una ventaja, ya que en lugar de hacer Scala más fácil de usar para los programadores promedio hace que no haya muchos patrones de implementación que existen en otros lenguajes como Java.

**Fuente:**

Como referencia, todo lo desarrollado para esta tecnología de acuerdo a los requerimientos de cada una de las fases se encuentra en el siguiente repositorio: https://github.com/mariobsdd/tec-scala