

# Informe TP2 - Intérprete Cálculo Lambda

## Grupo 12 - Los devaluados

En este trabajo se presentaba el problema de realizar un intérprete de cálculo lambda en Scala usando el paradigma funcional.

El intérprete consta de 3 módulos principales: el Lexer, el Parser y el Reductor. Todos los módulos se construyeron a partir del uso del Pattern Matching, ya que esta poderosa característica de la programación funcional nos permite hacer un desarrollo de una forma flexible y de fácil análisis, convirtiendo el procesamiento de las expresiones lambda a un mecanismo sencillo pero con la capacidad de procesar cadenas complejas de caracteres, siguiendo una convención de expresiones regulares predefinidas.

Utilizamos recursividad para ir iterando sobre las expresiones a analizar y procesar, sin utilizar ciclos, junto con Guards para evitar la utilización de flujos de control, respetando principios de este paradigma.

La hipótesis tomada (por enunciado) fue que las aplicaciones deben ir exclusivamente entre paréntesis y que se considera la expresión como incorrecta en el caso de que se encuentren en otro lado.

Como conclusión, pudimos apreciar las capacidades que nos brinda la programación funcional para modificar un dato a través de un pipeline de funciones. A la vez, apreciamos la gran diferencia que existe entre el paradigma orientado a objetos, que se basa en la interacción entre objetos y el encapsulamiento y el funcional, que se enfoca en la inmutabilidad de la información y la aplicación y composición de funciones.

Además, pusimos en práctica el concepto ya conocido de recursión y el nuevo concepto de pattern matching que resultan ser muy útiles y demuestran ser viables para su uso en software escalable.