

## JAVA COLLECTION FRAMEWORK

En este documento se especificaran cada una de las colecciones del JCF utilizadas en la implementación.

- Stacks: Usado en la clase Predicates y. Estos fueron usados para poder almacenar los predicados. Esto fue electo por su estructura last in first out, facilita la manera de lectura del código hecho en lisp, ya que lisp tiene similitud. El primer carácter ingresado indica el resto del código.
- LinkedList: Usada en la clase Arithmetic Operation, Functions y DEFUN este es usado para después de leer cada uno de los caracteres de la instrucción ingresada. Se puedan meter dentro de esta e irlos solicitando en otros puntos del código (cuanto se realice la operación o se requiera del operando u operador). Esta fue utilizada porque se tendría una gran movilidad de datos, se harían varias inserciones y eliminaciones. Por lo mismo es más sencillo usar un linkedList que una lista, ya que no se harían muchas búsquedas entre gran cantidad de datos.

```
Predicates.java x ArithmeticOperations.java x Defun.java x Functions.java
1 import java.util.LinkedList;
2 import java.util.List;
3 import java.util.Stack;
4
5 public class ArithmeticOperations {
6
7     private LinkedList<String> operations = new LinkedList<>(); //Linked list use
8     private String numero = ""; //stores a number (when is 2 digit or longer)
9
```

```
Predicates.java x ArithmeticOperations.java x Defun.java x Functions.java
1 import java.util.Stack;
2
3 public class Predicates {
4     private Stack<String> type = new Stack<>(); //Stack used to store the predic
5     private Stack<String> predicates = new Stack<>(); //store the predicates
6     public static boolean result; //returns true or false, depending on the resu
7
8     /**
9     * Constructor
10    * @param predicate
11    */
```