# Manual de usuario

# Pruebas de funcionalidad

# Descripción del problema

El avance del proyecto a diseñar es la fase de Análisis Léxico para gramática. Algunos aspectos relevantes del problema son los siguientes: leer un archivo fuente (.txt), escribir en un archivo todos los tokens encontrados (.html), identificador asociado con el lexema y debe de reportar y manejar los errores léxicos encontrados.

La herramienta que se tiene que usar es JFlex para realizar el análisis léxico junto con la librería java-cup. Se deben crear las definiciones regulares correspondientes para identificar los diferentes tokens con la siguiente información: tipo de token, lexema, línea de aparición y columna.

# Diseño del programa

La solución propuesta para el programa es la siguiente:

Se contará con 3 paquetes diferentes, los cuales son: Analizadores, Generadores y Programa. Analizadores cuenta con los archivos .jflex y .cup, adicionalmente en este paquete se crearán los archivos .java para las clases sym, Parser (java-cup) y Lexer (JFlex). Generadores cuenta con 2 archivos .java, los cuales se encargan de generar las diferentes clases para el funcionamiento correcto del programa. Por último, Programa cuenta con la clase App.java, para correr el programa (leer archivo fuente, guardar resultados en archivo .html y mostrar errores de tokens en consola), adicionalmente, se cuenta con la clase HTMLTable.java para generar el archivo .html.

El archivo lexer.jflex cuenta con todas las definiciones regulares para identificar los diferentes tokens. Al mismo tiempo, el archivo parser.cup cuenta con todos los nombres de los diferentes terminales.

Por último, al ser un proyecto creado en Maven todas las instalaciones de las dependencias se almacenan en un archivo pom.xml. De esta manera, se lleva un control exacto de las versiones de las librerías que se están utilizando y el compilar es más amigable.

# Librerías usadas

Las librerías que se utilizaron fueron JFlex para realizar el análisis léxico, adicionalmente se utilizó java-cup para hacer el parsing del archivo y trabajar junto con JFlex. Las librerías mencionadas anteriormente se instalaron con Maven, por lo tanto, se cuenta con un pom.xml para mantener un control de las diferentes dependencias y simplificar la compilación al usuario.

# Análisis de resultados