**Universidad de San Carlos de Guatemala**

**Facultad de Ingeniería**

**Escuela de Ciencias y Sistemas**

**Laboratorio Organización de Lenguajes y Compiladores 1**

**Proyecto 1: StatPy Convertor**

**Manual Técnico**

**Nombre:** Josué Daniel Rojché García

**Carné:** 201901103

**Fecha:** 16/09/2023

**Sección:** N

**Auxiliar:** Walter Alexander Guerra Duque

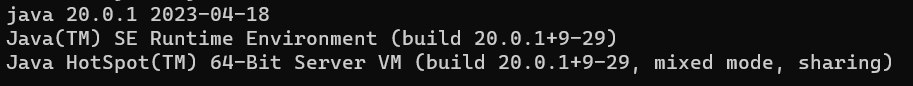
**Requerimientos del Sistema**

En la realización del software el IDE utilizado fue APACHE NetBeans, con las especificaciones que se observan en la siguiente imagen.

Interfaz de usuario gráfica

Descripción generada automáticamente

También se utilizó la versión del jdk de java como se observa en la siguiente imagen.



Para la realización del analizador léxico y sintáctico se utilizaron las librerías jflex, y cup, las versiones las puede visualizar en la siguiente imagen.

Texto

Descripción generada automáticamente

Para la realización de las gráficas estadísticas, se utilizaron las librerías siguientes.



**Analizador Léxico**

Para conseguir la solución del programa se realizaron dos analizadores léxicos diferentes, uno para el reconocimiento de archivos de tipo .json y otro para el reconocimiento de archivos .sp, por lo que se utiliza las librerías mencionadas anteriormente.

Para diferencias cual es el analizador correspondiente para cada uno, se utilizaron dos paquetes diferentes como se muestra en la siguiente imagen.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El archivo leer tiene la siguiente estructura, en la parte inicial hasta la parte anterior del archivo se debe mantener y no editar nada a excepción del pakage, luego añadimos las expresiones regulares que servirán para manejar las cadenas de texto, los números decimales, comentarios de línea y multilínea.

Texto

Descripción generada automáticamente

En reglas léxicas se declaran los caracteres que serán utilizados para analizar y también los correspondientes a las expresiones regulares, los nombres de las expresiones regulares deben ser igual en las reglas léxicas, y se debe agregar un nombre de token a cada uno de los utilizados, el método func.Funcion.addTokensJSON, sirve para agregar a la tabla de tokens cada uno de los tokens con forme el lexema, descripción, fila y columna. En la parte de errores léxicos se agrega el envio de los errores hacia una función que se encarga de almacenar en un LinkedList cada uno de estos errores que se encuentre al analizar.

Captura de pantalla de computadora

Descripción generada automáticamente

De la misma forma, se realiza la estructura para el analizador de statpy. La diferencia será que se tendrán diferentes expresiones regulares.

Texto

Descripción generada automáticamente

También se tendrán diferentes reglas lexicas, en este caso tendremos diferentes caracteres que servirán para validar que cumpla con la regla del lenguaje. Y básicamente se agregan los nombres de los tokens y a la función que recibe como parámetros los tokens, el lexema, fila y columna para almacenarlas en una LinkendList.

Texto

Descripción generada automáticamente

Luego de los caracteres se utilizan las palabras reservadas que tendrá el lenguaje, se agrega el nombre del token y luego se envían los tokens a la lista de tokens a través de la función.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza mediaMas adelante se utiliza también las variables utilizadas para las expresiones regulares, y para los comentarios multilínea, luego se agrega las agregaciones para las funciones de errores léxicos, el cual se encarga de agregar a la lista de errores.

**Analizador Sintáctico**

Texto

Descripción generada automáticamenteEste es el archivo Parser.cup, en este básicamente vamos a tener la siguiente estructura par manejar las producciones.

Luego hacemos la declaración de los no terminales y de las terminales que usaremos para las producciones.

Texto

Descripción generada automáticamente

En las producciones utilizamos las no terminales y las terminales declaradas anteriormente para poder crear la gramática de manera funcional. También dentro de la terminal obtenemos el valor del terminal asignándole una variable y luego dentro de las llaves, mandamos a llamar la función addlistarchJson, esto nos servirá Texto

Descripción generada automáticamentepara validar el ingreso de los datos a una linkedlist y luego a un hasmap.

La misma estructura se utiliza para manejar el sintactico de statpy.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ahora en la clase función se declararon las listas y hasmap y métodos utilizados para agregar a las listas y escribir el archivo html para las tablas de errores y de tokens.

Texto

Descripción generada automáticamente

Texto, Escala de tiempo

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para almacenar los símbolos se utilizó un objeto con los respectivos parámetros.

Texto

Descripción generada automáticamente

También una clase para manejar los objetos de tokens, con sus parámetros.

Captura de pantalla de computadora

Descripción generada automáticamente

Luego tenemos la clase para el objeto de errores, con sus respectivos parámetros

Texto

Descripción generada automáticamente

Para la traducción se realizó una declaración de un LinkedList para almacenar las traducciones, luego un contador para tabulaciones, y otras variables utilizadas para manejar los parámetros del for, también se agregó el método que se utiliza para agregar cada tabulación de acuerdo a su contador.

Texto

Descripción generada automáticamente

En el método main, se tienen los analizadores y la llamada al método que ejecuta la ventana grafica.



Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Dentro de la clase de la interfaz grafica, simplemente se tienen los métodos para manejar las acciones que se requieren para el buen funcionamiento de la aplicación de escritorio. Por lo que se muestran en las siguientes imágenes.

Texto

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Proceso de armado de la gramática en excalidraw.com

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente