**Universidad de San Carlos de Guatemala**

**Facultad de Ingeniería**

**Escuela de Ciencias y Sistemas**

**Laboratorio Organización de Lenguajes y Compiladores 1**

**Proyecto 1: DataForge**

**Manual Técnico**

**Nombre:** Josué Daniel Rojché García

**Carné:** 201901103

**Fecha:** 10/03/2024

**Sección:** N

**Auxiliar:** Walter Alexander Guerra Duque

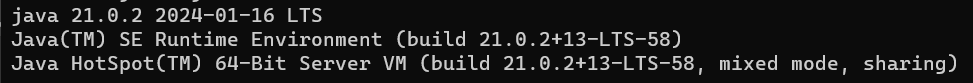
**Requerimientos del Sistema**

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, Texto, Sitio web

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, y los cálculos de funciones estadísticas se utilizaron las librerías siguientes.

Texto

Descripción generada automáticamente

**Analizador Léxico y Sintáctico**

Para conseguir la solución del programa se realizó analizador léxico, para el reconocimiento de archivos de tipo .df, por lo que se utiliza las librerías mencionadas anteriormente. En la siguiente imagen podrá observar el paquete para el analizador, con sus respectivos archivos.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Dentro del archivo Lexer.jflex se puede observar los caracteres y expresiones regulares y palabras reservadas que serán reconocidos por el lenguaje, así como el manejo de errores léxicos que puedan ocurrir.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Dentro del archivo Parser.cup se tiene el manejo de errores sintácticos, la gramática que servirá para verificar la correcta lectura del archivo y la declaración de los terminales y no terminales, también se realiza el manejo de la lógica para almacenar lo reconocido, como declaraciones, impresiones, operaciones, etc.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto, Escala de tiempo

Descripción generada automáticamente

**Manejo de Errores**

Para el manejo de los errores, se utilizó un objeto que contendría los parámetros que se observan en la imagen y estas a su vez se almacenaron dentro de una lista y así poder recorrerla para los reportes.

Texto

Descripción generada automáticamente

**Manejo de Tokens**

También se manejaron a través de objetos, y se almacenaron en una lista para poder utilizarlo en el reporte correspondiente, el objeto cuenta con los siguientes parámetros:

Texto

Descripción generada automáticamente

**Manejo de Simbolos**

Para este se manejaron dos objetos, el primero para almacenar datos con valores simples, y el segundo para manejar los que incluyen arreglos de valores, y todos estos objetos se almacenaron a su vez en hashmap y linkedlist para posteriormente poder ser utilizado para obtener los datos tanto para manejo de funcionalidades y el reporte. Para ello se muestran los dos objetos en las imágenes siguientes:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**Manejo de Operaciones**

Para las operaciones aritméticas y estadísticas se realizó una clase donde se tuviesen las funciones que reciben como parámetros los datos y los operan internamente, para así retornar los resultados, incluso se utilizó la librería commons.math3, de apache para las operaciones de funciones estadísticas como moda, mediana, varianza, etc.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

**Manejo de Gráficas**

Para tener un manejo adecuado de las gráficas, se realizó una clase para las mismas, en este se manejaron variables de listas y strings, para almacenar los valores y nombres de títulos, etc que servirían para las mismas.

Texto

Descripción generada automáticamente

Las gráficas acceden a los datos a través de un método en el cual se agregan los valores y títulos necesarios, y al crearse la grafica se almacena como imagen para así poder ser visualizado en el programa.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para el histograma fue necesario realizar otras operaciones, las cuales son la frecuencia, frecuencia acumulada y frecuencia relativa, estas a su vez se almacenan en una variable que servirá para realizar la impresión en consola y luego se crea la imagen de la gráfica.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente **Manejo de Funcionalidades**

Dentro de la clase función, se manejaron las variables, listas y hasmap necesario para el manejo de los datos que se van pasando por el analizador durante la ejecución de la entrada, como los tokens, errores y símbolos, así como el ordenamiento de los datos que se imprimen en consola.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

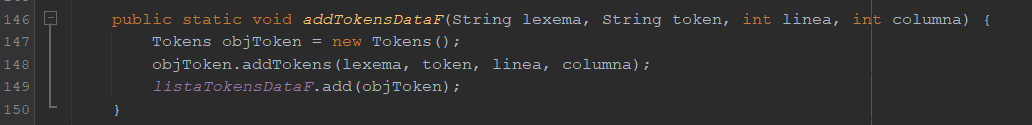
Texto

Descripción generada automáticamente

También funciones que sirven para la búsqueda de los valores de los id.

Texto

Descripción generada automáticamente



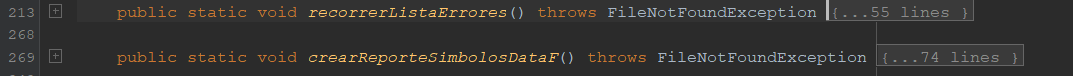
Las funciones para realizar los reportes correspondientes a símbolos, tokens y errores.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



Y el método que limpia todas las variables.

Texto

Descripción generada automáticamente

**Interfaz Grafica**

Dentro de la interfaz se manejan las funcionalidades como el nuevo archivo, abrir archivo, guardar, guardar como, eliminar pestaña, ejecutar, ver reportes. Dentro del código se utilizó JTextArea para el ingreso de los datos, y la salida de los mismos, junto con panel y scroll para poder visualizar los archivos cuando más contenido tuviese, también se utilizó JTabbedPane el cual sirve para realizar las pestañas, para las gráficas se mostraron por medio de un label el cual funciona para enviar ImageIcon y así mostrar las imágenes y para cambiarlas se utilizaron los botones siguiente y anterior. Para ello se utilizaron los siguientes métodos.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente