UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
LENGUAJES FORMALES DE PROGRAMACIÓN

MANUAL TÉCNICO

DIEGO ALESSANDRO GONZALEZ BRAVO / 202230939

Introducción

El proyecto desarrollado consiste en un analizador léxico que procesa código fuente en HTML, CSS, y JavaScript. El objetivo principal es identificar tokens clave en cada uno de estos lenguajes y proporcionar la capacidad de optimizar el código eliminando elementos repetidos. El sistema puede generar reportes detallados de los tokens encontrados y presentar los resultados de forma clara en una interfaz gráfica. También incluye una función de optimización de código, donde se eliminan líneas repetidas o innecesarias para hacer el código más limpio y eficiente.

Herramientas de desarrollo.
Las herramientas (IDES, lenguajes de programación) utilizadas para el desarrollo de la aplicacion Manejo de tarjetas son:

- JAVA
    - Version openjdk version "21.0.4"
- Apache NetBeans
    - Version 22
- Sistema operativo de desarrollo
    - Ubuntu 24.04 LTS

Descripción del Análisis Léxico

El análisis léxico es una etapa fundamental de la compilación de lenguajes de programación. El analizador léxico desarrollado para este proyecto tiene la capacidad de procesar tres tipos de lenguajes: HTML, CSS, y JavaScript. El analizador convierte el código fuente en una secuencia de tokens, donde cada token representa una unidad sintáctica básica del lenguaje (como etiquetas, propiedades CSS, funciones JS, etc.).

Componentes Principales

- Clase Token: La clase Token es el núcleo del analizador léxico. Cada instancia de esta clase representa un token específico del código fuente y contiene información sobre:
    - Valor: El valor literal del token (por ejemplo, una etiqueta HTML o una propiedad CSS).
    - Lenguaje: Indica si el token pertenece a HTML, CSS, o JavaScript.
    - Tipo: El tipo específico del token (por ejemplo, si es una etiqueta de apertura en HTML, una propiedad en CSS, etc.).
- Enumeraciones (Enums) para HTML, CSS, y JavaScript: Se utilizan tres enumeraciones (enum) para identificar los diferentes tipos de tokens que pueden aparecer en HTML, CSS, y JavaScript, respectivamente. Cada enum contiene los tokens permitidos para cada lenguaje.

Proceso del Análisis Léxico

1. Entrada del código: El código fuente se pasa al analizador como una cadena de texto. Dependiendo del tipo de código, el analizador invoca la clase correspondiente (HTML, CSS, o JS) para identificar los tokens.
2. Identificación de Tokens: Se realiza un recorrido del código línea por línea, detectando las palabras clave o patrones que coinciden con los valores de los enum del lenguaje. Estos tokens se almacenan en una lista de objetos Token.
3. Generación del Reporte: Una vez analizado el código, se puede generar un reporte de los tokens identificados. Este reporte se genera en formato HTML, incluyendo detalles como el valor del token, su expresión regular, el lenguaje al que pertenece, y su tipo.

Optimización del Código

La optimización del código consiste en procesar bloques de código fuente y eliminar líneas repetidas para mejorar la eficiencia y reducir el tamaño del archivo.

Funcionamiento de la Optimización

El optimizador recibe como entrada una lista de cadenas (List<String>) que representa las líneas del código a procesar. El objetivo es identificar y eliminar cualquier línea duplicada, manteniendo solo la primera aparición de cada línea.

Algoritmo de Optimización

1. Entrada: Se recibe una lista de cadenas que contiene el código fuente a optimizar.
2. Uso de un LinkedHashSet: Para garantizar que no haya repeticiones, se utiliza un LinkedHashSet. Esta estructura de datos permite almacenar los elementos en el orden de inserción y asegura que cada elemento es único.
3. Eliminación de duplicados: A medida que se recorren las líneas de código, el optimizador verifica si la línea ya ha sido procesada. Si no lo ha sido, se agrega al conjunto. Esto asegura que solo se retienen las líneas necesarias.
4. Salida: El código optimizado se devuelve como un solo String en el que cada línea se presenta una única vez.

Implementación Técnica

Análisis Léxico

El análisis léxico está implementado utilizando una estructura de clase central Token y métodos especializados para identificar tokens en HTML, CSS, y JavaScript. Las clases tienen la siguiente estructura:

```
public class Token {

    private String value;

    private TokenTypeHTML typeHTML;

    private TokenTypeCSS typeCSS;

    private TokenTypeJS typeJS;

}
```

Este sistema proporciona una herramienta útil para analizar y optimizar código en HTML, CSS, y JavaScript. A través del análisis léxico, es posible identificar tokens clave y generar reportes detallados. Además, la optimización del código garantiza que se eliminen repeticiones, lo que mejora la eficiencia y legibilidad del código.

Diagrama de clases: