



Facultad de Ingeniería
Ingeniería en Ciencias y Sistemas
Lenguajes Formales y de Programación sección "A"

Manual Técnico "Grafos Guatemala"

Nombre
Edy Rolando Rojas González

Carné
201730511

16 de junio de 2024

ÍNDICE

ÍNDICE.....	2
INTRODUCCIÓN.....	3
OBJETIVOS.....	4
Objetivo General.....	4
Objetivos Específicos.....	4
DESCRIPCIÓN DE LA SOLUCIÓN.....	5
LÓGICA DE SOLUCIÓN.....	6

INTRODUCCIÓN

Este manual técnico proporciona información técnica de la aplicación desarrollada para “Grafos de Guatemala”, incluye información detallada del lenguaje utilizado para el desarrollo, las librerías utilizadas, la lógica general, las plataformas y herramientas utilizadas en el desarrollo, y las versiones en el que fueron desarrolladas.

La aplicación tiene como fin procesar las instrucciones del archivo de entrada seleccionado con extensión “.code”, visualizar las imágenes de los grafos resultantes del procesamiento y poder guardar el reporte de tokens y el reporte de errores en formatos HTML.

El procesamiento del archivo de entrada se realiza a través de un analizador léxico a la medida que se rige bajo lo requerimientos del enunciado, se utiliza un autómata finito determinista (AFD) para poder identificar los tokens y los errores para posteriormente poder generar los reportes y los grafos que son renderizados en imágenes con extensión “.png”.

OBJETIVOS

Objetivo General

- Proporcionar la documentación necesaria, diagramas, clases utilizadas, librerías utilizadas y aspectos técnicos relevantes para facilitar el mantenimiento y modificaciones futuras realizadas por terceros.

Objetivos Específicos

- Compartir una descripción del SO, IDE y versión de lenguaje utilizado durante del desarrollo.
- Otorgar una visión general de la aplicación a través de un diagrama de flujo.
- Presentar una descripción y explicación del AFD desarrollado.

DESCRIPCIÓN DE LA SOLUCIÓN

Para el desarrollo de esta aplicación se recolectaron los requerimientos funcionales y no funcionales descritos por el cliente.

Después de entendido los requerimientos y tomando las consideraciones de prioridad se listo lo siguiente.

- Requerimientos funcionales
 - Se debe poder seleccionar un archivo de entrada de extensión .code.
 - Se procesa la información del archivo de entrada identificando los tokens y errores, estos se almacenan en una lista posteriormente para poder generar reportes en HTML.
 - Seguido del procesamiento del archivo de entrada se generan los grafos en imágenes con extensión .png las cuales deben poder ser visualizadas en la interfaz gráfica.
 - Reporte de tokens, este debe contener:
 - Nombre del token
 - Lexema
 - Fila.
 - Columna
 - Reporte de errores, este debe contener:
 - Token que estaba siendo reconocido
 - Lexema
 - Fila
 - Columna
 - Carácter de error
- Requerimientos no funcionales
 - Desarrollar una interfaz gráfica intuitiva para cargar archivo, ejecutar archivo, visualizar las imágenes de los grafos y generar los reportes para poder ser guardados en la ubicación seleccionada.
 - Utilizar Graphviz para graficar los grafos.
 - Utilizar el lenguaje Python.

LÓGICA DE SOLUCIÓN

El flujo general de la aplicación es el siguiente:

El usuario presiona el botón de “Cargar Archivo” y selecciona el archivo de entrada con extension .code, seguidamente se presiona el botón de “Ejecutar Archivo” y el archivo cargado en memoria se analiza y el AFD procesa, identifica y almacena lo siguiente:

- Nombre del token
- Lexema
- Fila.
- Columna

En el caso de exista un error el AFD lo identifica y guarda en el listado de errores:

- Token que estaba siendo reconocido
- Lexema
- Fila
- Columna
- Carácter de error

Con el archivo procesado se pasa a la parte de generar los grafos, estos son exportados a .png cargados al combobox para poder seleccionar la imagen y ser cargada a la interfaz gráfica para su visualización.

Procesado y generadas la imágenes para su visualización el usuario puede generar los reportes, al dar clic en el botón de “Reporte de Tokens” se le solicitará al usuario seleccionar una ubicación para guardar el reporte en HTML, este reporte en HTML cuenta con estilos CSS para una visualización más amigable. El procedimiento anterior es el mismo para reporte de errores con la diferencia que se debe dar clic al botón de “Reporte de Errores”.

Al finalizar el procedimiento es posible cargar un nuevo archivo para ser procesado y las variables son reiniciadas.

En el siguiente gráfico se muestra el diagrama de flujo general de la aplicación.

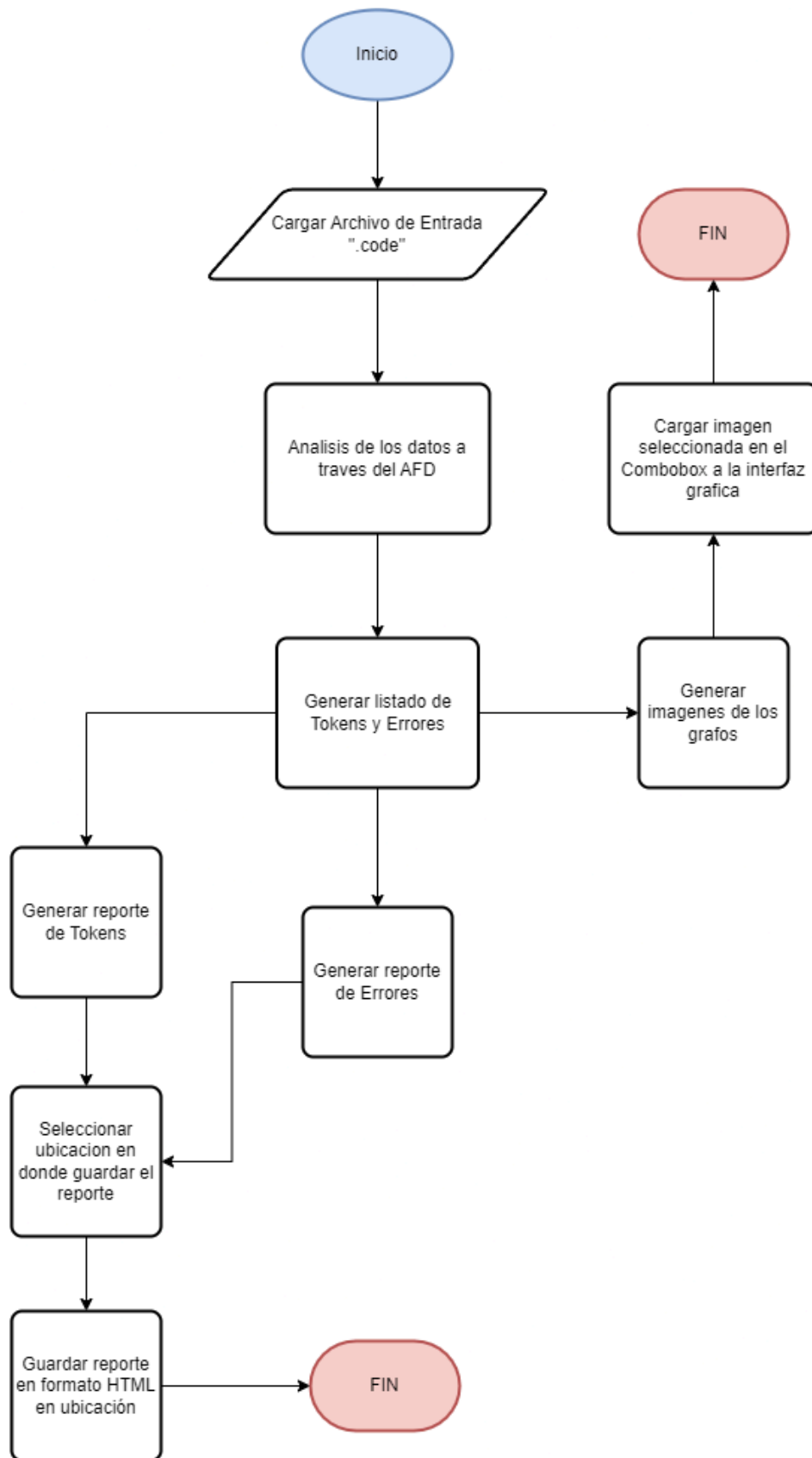
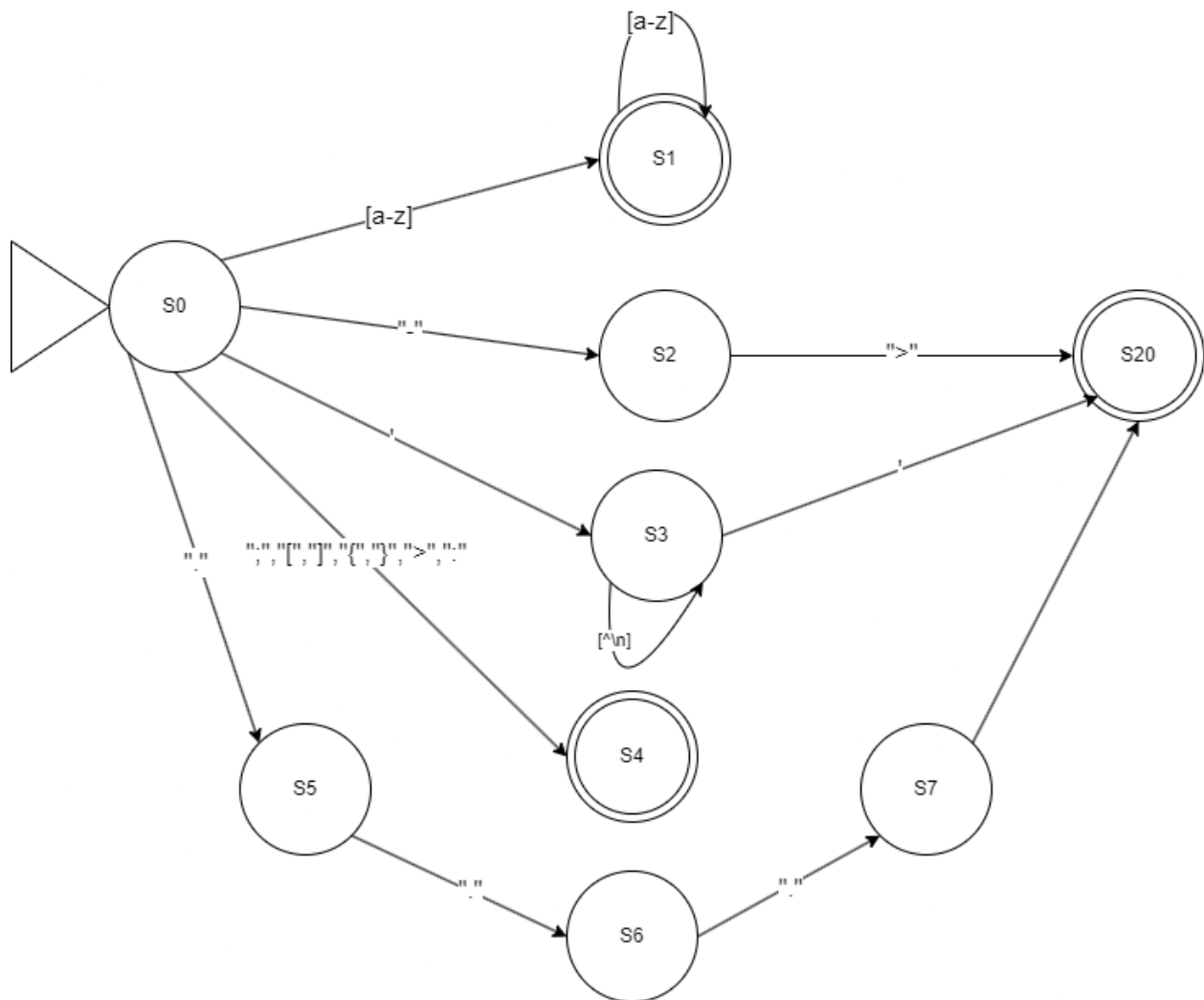


Diagrama de transiciones del AFD.



ENTORNO DE DESARROLLO

Para el desarrollo de la aplicación fue utilizado PyCharm como IDE debido a que es la herramienta por excelencia para el desarrollo en Python, otorgando una gran cantidad de herramientas y plugins ya configurados, proporciona sugerencias inteligentes y autocompletado de código, lo que facilita la escritura rápida y precisa.

- Version del IDE utilizada:
 - PyCharm 2024.1.3 (Professional Edition)
 - Build #PY-241.17890.14, built on June 4, 2024
 - Runtime version: 17.0.11+1-b1207.24 amd64
 - VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o.
 - Windows 11.0
 - GC: G1 Young Generation, G1 Old Generation
 - Memory: 4064M
 - Cores: 12
 - Registry:
 - ide.experimental.ui=true
 - Non-Bundled Plugins:
 - com.jetbrains.space (241.17890.8)
 - com.intellij.ml.llm (241.17890.15)
- Lenguaje de programación:
 - Python en su versión 3.12.4
- Sistema operativo
 - Sistema operativo de 64 bits
 - Nombre del dispositivo ryuk
 - Procesador 12th Gen Intel(R) Core(TM) i5-12400 2.50 GHz
 - RAM instalada 32.0 GB (31.7 GB usable)
 - Identificador de dispositivo D0A7005E-E4EC-4772-B8C9-3A6345BC06CE
 - Id. del producto 00330-80000-00000-AA025
 - Tipo de sistema Sistema operativo de 64 bits, procesador basado en x64
 - Lápiz y entrada táctil La entrada táctil o manuscrita no está disponible para esta pantalla
 - Edición Windows 11 Pro

- Versión 23H2
- Instalado el 24/05/2023
- Versión del sistema operativo 22631.3737
- Experiencia Windows Feature Experience Pack 1000.22700.1009.0

LIBRERÍAS UTILIZADAS

Las librerías utilizadas para el desarrollo del proyecto son las siguientes:

- `os`, para acceder a las carpetas
- `tkinter`, para la creación de interfaz gráfica
- `time`, para generar un id único en los reportes
- `jinja2`, como gestor de plantillas HTML
- `graphviz`, para generar los grafos e imágenes