

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES DE PROGRAMACIÓN
SEGUNDO SEMESTRE 2023



| Sección | Catedrático | Tutor Académico |
|---------|--------------------------------------|-----------------------------------|
| A+ | Ing. Otto Amilcar Rodríguez Acosta | Erick Daniel Antillón Chinchilla |
| A- | Inga. Vivian Damaris Campos González | Enrique Alejandro Pinula Quiñonez |
| B+ | Ing. David Estuardo Morales Ajcot | Francisco Magdiel Asicono Mateo |
| B- | Inga. Zulma Karina Aguirre Ordoñez | Elder Anibal Pum Rojas |

PROYECTO NO. 1

Aplicación Numérica con Análisis Léxico

DESCRIPCIÓN GENERAL

1. OBJETIVO GENERAL

Que el estudiante cree una herramienta la cual sea capaz de reconocer un lenguaje, dado por medio de un analizador léxico el cual cumple con las reglas establecidas, manejando la lectura y escritura de archivos para el manejo de la información. A través de un entorno gráfico.

2. OBJETIVO ESPECÍFICOS

- Implementar por medio de estados un analizador léxico.
- Utilizar funciones de manejo de cadenas de caracteres en lenguaje Python.
- Programar un Scanner para el análisis léxico.
- Construir un scanner basándose en un autómata finito determinístico.
- Crear una herramienta para interactuar de forma visual con el usuario con Tkinter
- Crear diagramas con la librería Graphviz

DESCRIPCIÓN

Se solicita la lectura de código fuente, el cual tendrá un formato JSON, creando un programa el cual sea capaz de identificar un lenguaje dado, identificando los errores léxicos y ejecutando las instrucciones correspondientes.

Se listarán una serie de instrucciones las cuales deben de ser ejecutadas, cumpliendo con el formato asignado, generándolo un resultado y graficarlos en un archivo según la jerarquía operacional de cada instrucción. **Colocando el resultado en cada nodo que aplique.**

Los errores deben ser generados en un archivo **JSON**.

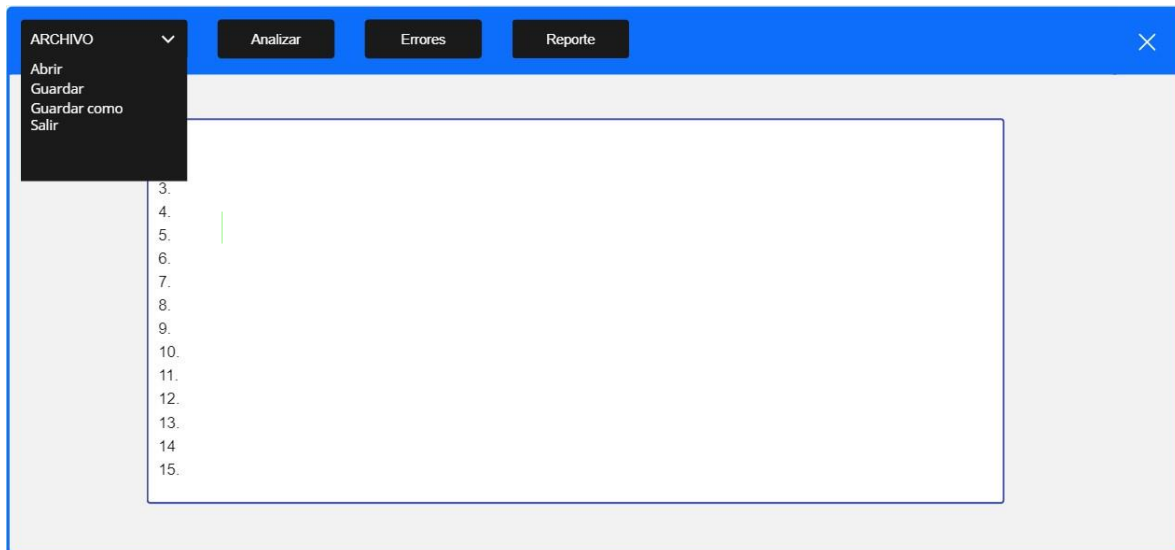
FUNCIONALIDAD

Se ha pensado en el desarrollo de una aplicación en lenguaje Python haciendo uso de la librería de Tkinter que permita reconocer las distintas instrucciones, y la ejecución de estas. Con el objetivo que se implemente el análisis léxico correspondiente.

Se deben de mostrar de manera funcional y agradable al usuario resumen de errores detectados, así como el resultado de las operaciones realizadas en cada una de las funciones que se describen más adelante, así como su respectivo archivo en forma gráfica (árbol de operaciones, incluyendo el resultado en cada nodo que aplique) generado con la librería Graphviz.

El código será cargado en un área de texto, se podrá modificar, guardar el archivo con el mismo nombre, o bien guardar el archivo con diferente nombre, y al momento de analizar se deberán mostrar los respectivos diagramas del árbol creado de acuerdo con la operación, siempre considerando la jerarquía operacional, además deberá mostrar los errores encontrados en un archivo JSON, indicando que provocó el error. **RESULTADOS_#Carnet**

INTERFAZ



MENU

Archivo:

- **Abrir:** Permite abrir un archivo para poder seguir editándolo en la aplicación
- **Guardar:** Permite guardar el archivo que está siendo editado con el nombre actual.
- **Guardar como:** Permite guardar el archivo que está siendo editado con otro nombre.
- **Salir:** Con esta opción se cerrará la aplicación.

Analizar: Analizará el texto y mostrará los elementos reconocidos.

Errores: Muestra los errores con el formato que más adelante se detalla del último archivo compilado.

Reporte: Generar los diagramas de las operaciones previamente analizadas

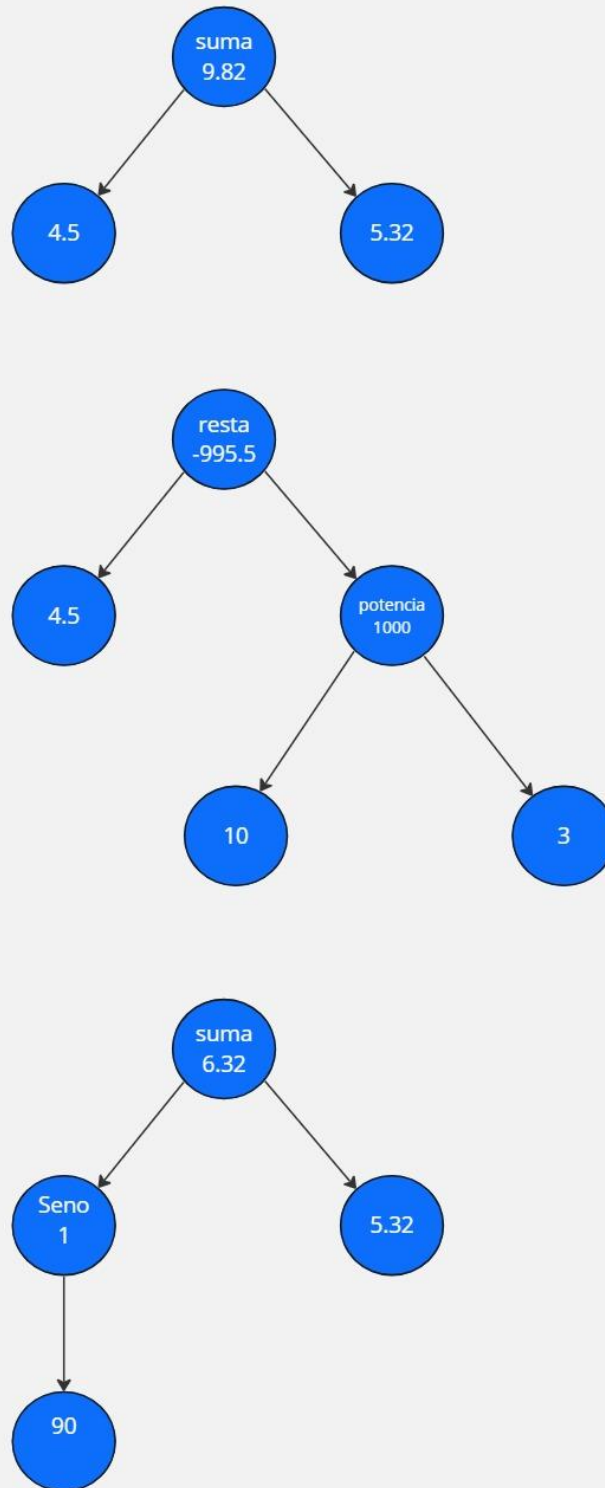
OPERACIONES VÁLIDAS

| FUNCIÓN | OPERACIÓN |
|----------------|--|
| SUMA | Suma de 2 o más números u operaciones anidadas. |
| RESTA | Resta de 2 o más números u operaciones anidadas. |
| MULTIPLICACIÓN | Multiplicación de 2 o más números u operaciones anidadas. |
| DIVISIÓN | División entre números u operaciones anidadas. |
| POTENCIA | Potencia N de un número u operación anidadas. |
| RAIZ | Raíz N de un número u operación anidadas. |
| INVERSO | Inverso de un número u operación anidadas. |
| SENO | Función trigonométrica seno de un número u operación anidadas. |
| COSENO | Función trigonométrica coseno de un número u operación anidadas. |
| TANGENTE | Función trigonométrica tangente de un número u operación anidadas. |
| MOD | Residuo entre números u operaciones anidadas. |

ESTRUCTURA DEL ARCHIVO DE ENTRADA

```
{
  "operaciones": [
    {
      "operacion": "suma",
      "valor1": 4.5,
      "valor2": 5.32
    },
    {
      "operacion": "resta",
      "valor1": 4.5,
      "valor2": [
        {
          "operacion": "potencia",
          "valor1": 10,
          "valor2": 3
        }
      ]
    },
    {
      "operacion": "suma",
      "valor1": [
        {
          "operacion": "seno",
          "valor1": 90
        }
      ],
      "valor2": 5.32
    },
    {
      "operacion": "multiplicacion",
      "valor1": 7,
      "valor2": 3
    },
    {
      "operacion": "division",
      "valor1": 15,
      "valor2": 3
    }
  ],
  "configuraciones": [
    {
      "texto": "Operaciones",
      "fondo": "azul",
      "fuente": "blanco",
      "forma": "circulo"
    }
  ]
}
```

EJEMPLO DIAGRAMA DEL ARCHIVO ANALIZADO



ESTRUTURA DEL ARCHIVO DE ERRORES

```
{
  "errores":[
    {
      "No":1,
      "descripcion":{
        "lexema": "?",
        "tipo":"error lexico",
        "columna":2,
        "fila":9
      }
    },
    {
      "No":2,
      "descripcion":{
        "lexema": "!",
        "tipo":"error lexico",
        "columna":4,
        "fila":5
      }
    }
  ]
}
```

ENTREGABLES

En UEDI entregar únicamente el link del repositorio de GitHub que debe incluir:

- Manual técnico en markdown
- Manual de usuario en markdown
- Código fuente

CONSIDERACIONES IMPORTANTES

- El proyecto se deberá realizar en forma individual.
- Se debe de crear un repositorio privado en GitHub con el siguiente nombre: LFP_S2_2023_Proyecto1_#Carnet
- El proyecto se implementará en lenguaje Python, en caso contrario no será calificado.
- No se aceptan entregas vía correo electrónico u otro medio.
- Se valorará la calidad de la información proporcionada por la aplicación cuando se produzcan errores, así como la presentación de la interfaz gráfica y amigabilidad de la aplicación.

- Sistema Operativo Libre
- Utilizar para el entorno gráfico únicamente la librería TKINTER
- Utilizar para generar los diagramas únicamente Graphviz
- Agregar al auxiliar correspondiente de su sección como colaborador a su repositorio Github para poder validar los commits hasta el día de la entrega de su proyecto en UEDI.
- Deberá mantener el repositorio y el permiso disponible hasta finalizado el semestre.
- La calificación se realizará en línea y se grabará, esto para que quede constancia de la forma en la que se calificó y como soporte en la toma de decisiones en reclamos por parte del alumno si se presenta el caso.
- La calificación es personal con una duración máxima de 30 minutos, en el horario posteriormente convenido. Durante la calificación el estudiante no podrá modificar el archivo de entrada ni el código fuente de su aplicación.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación (1 días antes) para ceder su lugar a otro estudiante, en caso contrario el estudiante solo obtendrá el 80% de su nota obtenida.
- No se dará prórroga para la entrega del proyecto 1.
- **COPIA PARCIAL O TOTAL DE LA PRÁCTICA TENDRÁ UNA NOTA DE 0 PUNTOS, Y SE NOTIFICARÁ A LA ESCUELA DE SISTEMAS PARA QUE SE APLIQUEN LAS SANCIONES CORRESPONDIENTES.**
- En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará la práctica; por lo cual, se tendrá una nota de cero puntos.
- Fecha de entrega: 22 de septiembre de 2023, antes de las 23:59, no se recibirán entregas después de la fecha y hora establecida.