



## **Proyecto #1**

### **Manual Técnico**

La presente aplicación se realizó con ayuda del programa Visual Studio Code 2020 en el lenguaje python, implementado en Tkinter.

Durante su elaboración, se utilizaron tres archivos .py para cada analizador y una clase main (el único form usado). A continuación una breve descripción del contenido de cada archivo .py, que sirvió como analizador léxico por orden de implementación:

- Tokens: la presente clase se usa estrictamente para catalogar cada lexema recibido durante el análisis, utilizando un “enum” para listar cada una de las posibilidades de token, y getters y setters para poder rellenar este campo. La función externa de esta clase es proporcionar un token a cada lexema aceptado, cosa que se ve reflejada en la tabla de reportes.
- Analizador Léxico: esta clase fue usada para analizar el documento de entrada caracter a caracter y concatenar lo necesario para poder generar lexemas aceptados, además de encontrar errores en el mismo.
  - La deque escanear se encarga de detectar y clasificar tokens por medio de la comparación caracter a caracter de coincidencias entre símbolos, cadenas y palabras reservadas.
  - El método AgregarToken sirve como auxiliar a la lista superior, limpiando el auxiliar léxico y agregando los lexemas ya sorteados a la lista.
- Analizador Sintáctico: esta clase fue usada para analizar el documento de entrada token a token teniendo en cuenta en el que el tipo de cada uno de estos es esperado para una traducción satisfactoria. Así como de generar la tabla de símbolos con los valores de cada variable con el paso de las distintas líneas de código.
- Token[Lenguaje]: es una clase cuyo propósito es reunir todos los puntos de información necesarios de cada token aprobado en un solo objeto, y así poder ser añadidos a una lista y posteriormente a la tabla de reportes.
- ErrorLéxico: el propósito de esta clase es reunir todos los puntos de información requeridos para completar la tabla de errores léxicos dentro de los reportes de

manera satisfactoria, para luego ser añadidos como objeto a una lista y posteriormente pasar al html.

- ErrorSintáctico: el propósito de esta clase es reunir todos los puntos de información requeridos para completar la tabla de errores sintácticos dentro de los reportes de manera satisfactoria, para luego ser añadidos como objeto a una lista y posteriormente pasar al html.

Las librerías utilizadas para el proyecto fueron las siguientes:

En el archivo main:

- Tkinter - para la interfaz grafica
- Os
- Collections
- Pathlib

En los analizadores:

- Enum
- Collections
- Pathlib
- OS
- Errno
- Subprocess
- Webbrowser
- time