

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Laboratorio Lenguajes Formales y de Programación
Auxiliar Douglas Omar Arreola Martínez
Sección "B -"



"MANUAL DE USUARIO"

Luis Angel Barrera Velásquez

Carné: 202010223

Índice

| | |
|--|----|
| Objetivos | 3 |
| General:..... | 3 |
| Específicos: | 3 |
| Introducción..... | 4 |
| Descripción del Programa | 5 |
| Descripción de las Funciones del Programa..... | 6 |
| 1. Cargar..... | 6 |
| 2. Analizar..... | 7 |
| 3. Área de ingreso de texto | 7 |
| 4. Consola | 7 |
| 5. Reportes de errores | 7 |
| 6. Reporte de errores | 8 |
| Ruta de guardado de reportes HTML..... | 9 |
| Estructura del archivo LFP | 10 |

Objetivos

General:

Establecer una guía lo suficientemente estructurada para un entendimiento fácil del uso del software de generación de reportes del proyecto 2 y lograr de esta manera que el usuario de este software de uso de este de forma adecuada para que funcione correctamente y de forma optimizada.

Específicos:

- Proporcionar al usuario una explicación detallada de cada función del programa para un fácil manejo y evitar confusiones.
- Explicar el formato LFP para que el usuario pueda a través de un archivo de entrada generar reportes de los datos ingresados.
- Proporcionar reportes de errores en caso existan, para que el usuario de manera fácil y rápida pueda identificarlos y corregirlos.

Introducción

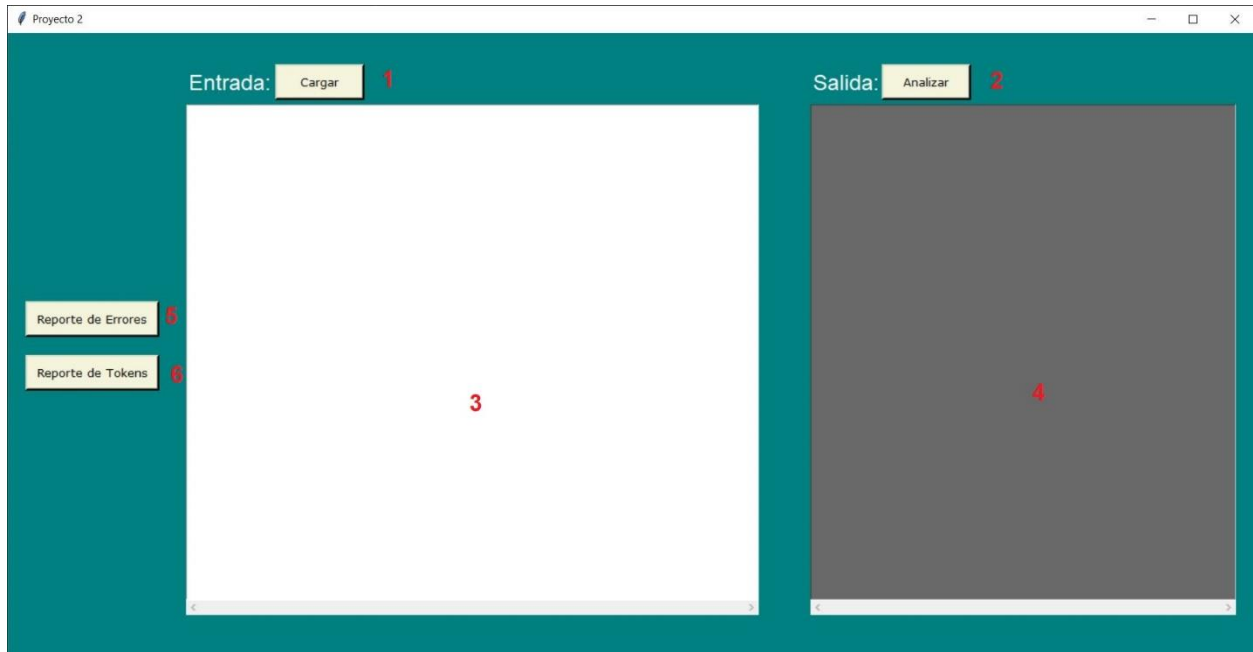
La realización de este manual es que el usuario de este programa lo conozca y se familiarice con cada una de las funciones y pueda sacar el máximo provecho de poder generar reportes tanto en consola como en archivos HTML y que tenga una experiencia agradable, como también dar a conocer el formato LFP y como se debe estructurar al momento de ingresar los datos que se desean procesar con diferentes comandos con funcionalidades del software.

El propósito del desarrollo de este software es proporcionar una herramienta de generación de reportes con el manejo de datos ingresados a través de un formato establecido el cual se explicara durante el desarrollo de este manual, esto de mucha utilidad cuando se tiene una gran cantidad de datos por ejemplo puede ser aplicado este software a una pequeña empresa, como también en caso de haber algún error en el archivo proporcionar donde están ubicados los errores y de esta forma el usuario pueda corregir donde tiene el error fácilmente.

Descripción del Programa

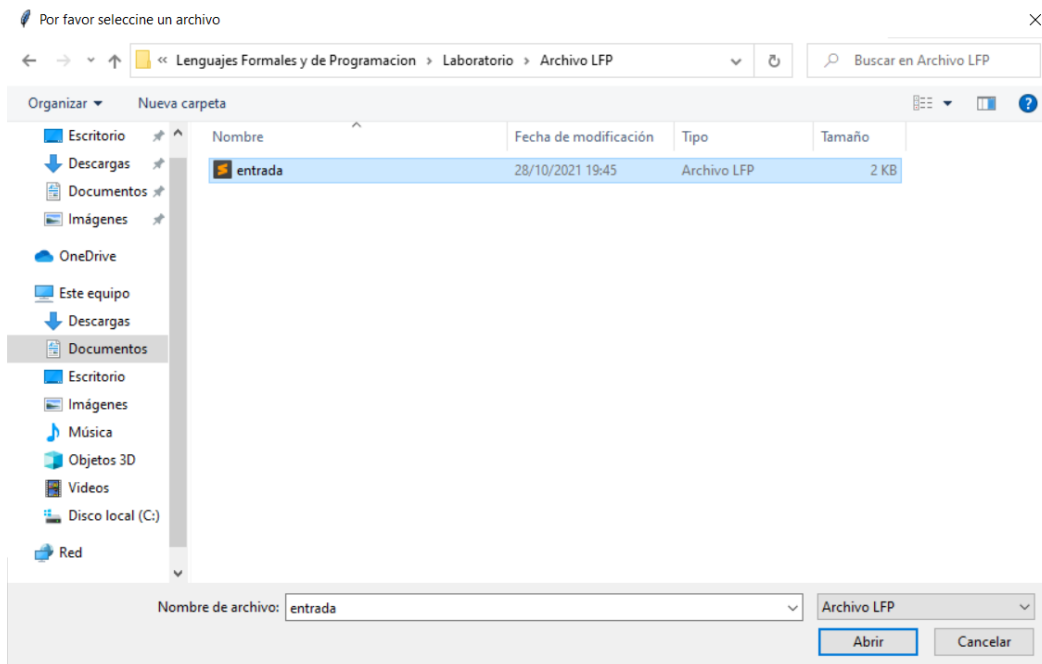
Es una aplicación que es operativa por medio de una interfaz grafica la cual su propósito es brindar al usuario una manera fácil de generar reportes de manera rápida y sencilla de entender por medio de los comandos con funcionalidades dentro del software, el software necesita que se le introduzcan claves y registros para poder realizar sus funciones, si existe algún error no ejecutara nada de sus funcionalidades por lo que también se incluye un apartado de generación de reporte de errores esto será de mucha ayuda al usuario para que pueda corregir el error del archivo de entrada o de la entrada escrita a mano de la interfaz teniendo la opción de poder generar un reporte de los datos ingresados en formato HTML.

Descripción de las Funciones del Programa



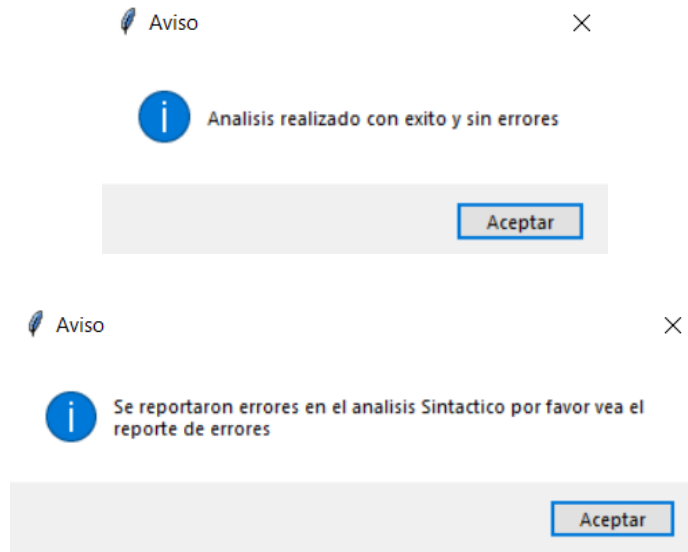
1. Cargar

Esta opción al pulsar sobre ella abrirá una ventana por la cual deberá seleccionar el archivo LFP que contiene todos los datos que serán utilizados por el software para generar reportes.



2. Analizar

Esta opción procesará el archivo que fue ingresado en la opción anterior mostrando al finalizar un mensaje si se finalizó el análisis con éxito o no.



3. Área de ingreso de texto

Esta sección se puede introducir de forma manual todos las claves y registros así como también los comandos con funcionalidades dentro del software, también si se carga un archivo el contenido se mostrara en esta área y se podrá editar si lo desea el usuario.

4. Consola

Esta sección es muy parecida a la anterior pero con la diferencia que esta únicamente mostrara texto y no se podrá editar su contenido, en esta área se mostrara la ejecución de los comandos ingresados como también algunos mensaje acerca de errores de ejecución del sistema.

5. Reportes de errores

Esta funcionalidad genera un reporte en formato HTML con todos los tokens encontrados durante el análisis del archivo de entrada o del texto ingresado manualmente en el área de ingreso de texto.

Reportes

Tabla de Tokens

| No. | TOKEN | LEXEMA | FILA | COLUMNA |
|-----|------------|--------------|------|---------|
| 1 | claves | Claves | 9 | 2 |
| 2 | igual | = | 9 | 8 |
| 3 | Corchete_a | [| 9 | 10 |
| 4 | cadena | dorsal | 9 | 12 |
| 5 | coma | , | 9 | 19 |
| 6 | cadena | nombre | 9 | 22 |
| 7 | coma | , | 9 | 29 |
| 8 | cadena | apellido | 9 | 32 |
| 9 | coma | , | 9 | 41 |
| 10 | cadena | nacionalidad | 9 | 44 |
| 11 | coma | , | 9 | 57 |
| 12 | cadena | edad | 9 | 60 |
| 13 | Corchete_c |] | 9 | 65 |

6. Reporte de errores

Esta funcionalidad genera un reporte en formato HTML donde se muestran todos los errores ocurridos durante la ejecución del análisis para que sea más fácil identificar los errores y aplicar correcciones para que el software se ejecute de forma adecuada y eficiente.

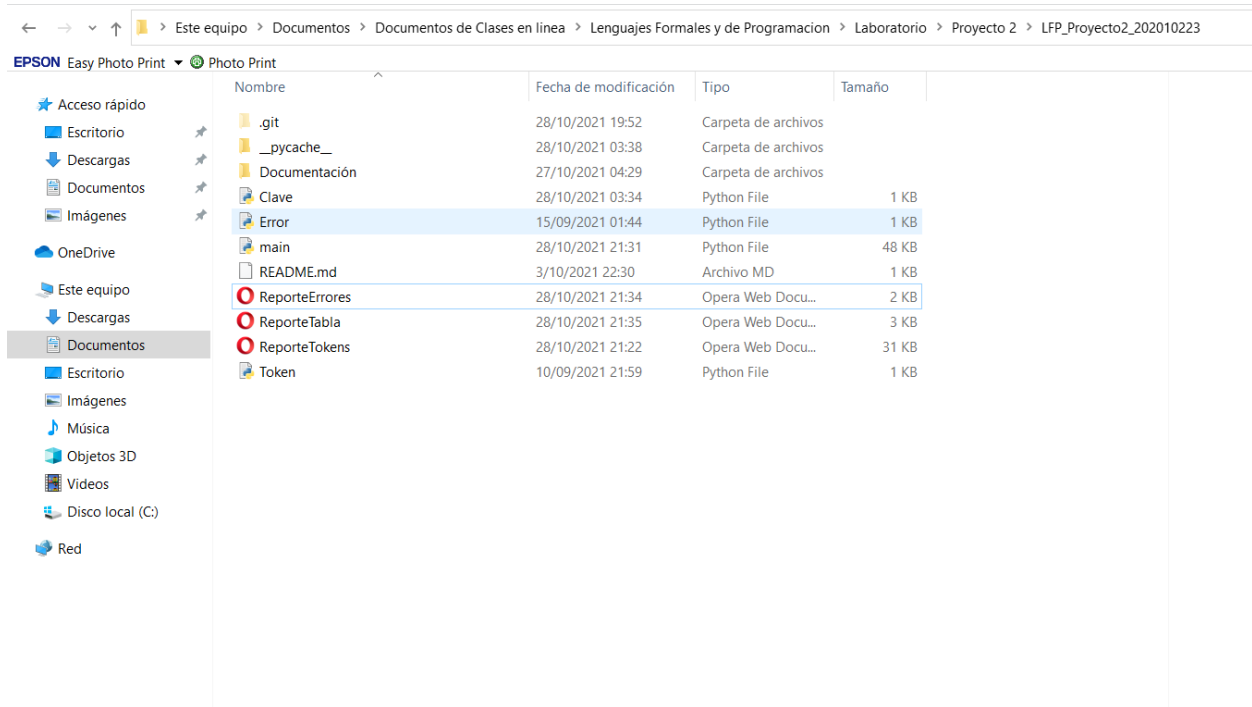
Reportes

Tabla de Errores

| No. | FILA | COLUMNA | CARACTER | OBSERVACION |
|-----|------|---------|----------|------------------------------|
| 1 | 7 | 2 | claves | se esperaba token punto_coma |

Ruta de guardado de reportes HTML

Para facilitar encontrar estos reportes el software genera los reportes en la misma carpeta donde se encuentra ubicado y se esta ejecutando, de esta manera el usuario puede ubicar el archivo que solicito de manera fácil y sencilla. Cada reporte tiene nombre diferente para no ser confundido tal como se muestra a continuación la carpeta donde se esta ejecutando el programa:



Estructura del archivo LFP

Sección de Claves: En esta sección se declaran los claves o campos por los que están contruidos los registros, su estructura está formada por la palabra reservada Claves, seguido de signo igual, corchete de apertura, lista de claves y corchete de cierre.

Lista de claves está formada por cadenas de caracteres encerradas entre comillas y separadas por coma.

```
Claves = [  
    "clave_1", "clave_2", "clave_3", "clave_4"  
]
```

```
Claves = [  
    "codigo", "producto", "precio_compra",  
    "precio_venta", "stock"  
]
```

Sección de Registros: En esta sección se detallan los registros que se quieren analizar y sigue la estructura dada por palabra reservada Registros, signo igual, corchete de apertura, lista de registros y corchete de cierre.

Lista de registros: Cada registro está encerrado entre llave de apertura y llave de cierre y sus valores están separados por comas, estos valores pueden ser cadenas de texto, enteros o decimales.

```
Registros = [  
    {valor1, valor2, valor3, valor4}  
    {valor1, valor2, valor3, valor4}  
    {valor1, valor2, valor3, valor4}  
    {valor1, valor2, valor3, valor4}  
]
```

```
Registros = [
    {1, "Barbacoa", 10.50, 20.00, 6}
    {2, "Salsa", 13.00, 16.00, 7}
    {3, "Mayonesa", 15.00, 18.00, 8}
    {4, "Mostaza", 14.00, 16.00, 4}
]
```

Comentarios:

Comentarios de una línea: Se representan con un numeral y finalizan con un salto de línea.

```
# Comentarios
```

Comentarios multilínea: Inicia con tres comillas simples y finaliza con tres comillas simples.

```
'''
comentario multilínea
'''
```

Instrucciones de Reportería:

imprimir(cadena): Imprime por consola el valor dado por la cadena.

```
imprimir("Reporte de ");
imprimir("Abarrotería");
>>> Reporte de Abarrotería
```

imprimirln(cadena):

```
imprimirln("Reporte de ");
imprimirln("Abarrotería");
>>> Reporte de Abarrotería
>>> Abarrotería
```

conteo(): Imprime por consola la cantidad de registros en el arreglo de registros.

```
conteo();  
>>> 46
```

promedio("campo"): Imprime por consola el promedio del campo dado.

```
promedio("stock");  
>>> 6.25
```

contarsi("campo", valor): Imprime por consola la cantidad de registros en la que el campo dado sea igual al valor dado.

```
contarsi("stock", 0);  
>>> 0  
  
contarsi("stock", 1);  
>>> 18  
  
contarsi("stock", 2);  
>>> 7
```

datos(): Imprime por consola los registros leídos.

sumar("campo"): Suma todos los valores del campo dado.

```
datos();  
>>> codigo  producto  precio_compra  precio_venta  stock  
>>> 1      Barbacoa   10.50         20.00         6  
>>> 2      Salsa      13.00         16.00         7  
>>> 3      Mayonesa   15.00         18.00         8  
>>> 4      Mostaza    14.00         16.00         4
```

max("campo"): Encuentra el valor máximo del campo dado.

```
max("precio_venta");  
>>> 20.00
```

min("campo"): Encuentra el valor mínimo del campo dado.

```
min("precio_compra");  
>>> 10.50
```

exportarReporte("titulo"): Genera un archivo html con una tabla en donde se encuentren los registros leídos y con el título como parámetro.

```
exportarReporte("Reporte HTML de abarrotería");
```

| Reporte HTML de abarrotería | | | | |
|-----------------------------|----------|---------------|--------------|-------|
| codigo | producto | precio_compra | precio_venta | stock |
| 1 | Barbacoa | 10.50 | 20.00 | 6 |
| 2 | Salsa | 13.00 | 16.00 | 7 |
| 3 | Mayonesa | 15.00 | 18.00 | 8 |
| 4 | Mostaza | 14.00 | 16.00 | 4 |