



Universidad De San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Lenguajes Formales y de programación

Sección “B-”



## **Manual de Usuario**

Proyecto 2 [LFP 2]

Autor: Sergie Daniel Arizandieta Yol

202000119

Versión: 1.0

Fecha: 28-10-2021

# **I. Objetivos**

## **1. Objeto del documento**

El documento tiene como finalidad proporcionar una guía del software con la cual puede conocer el manejo adecuado y correcto que la aplicación necesita para el funcionamiento de dicho programa.

## **2. Objetivos**

- Otorgar al usuario una explicación grafica simple y concisa de entender todas las funcionalidades que el software posee de manera lógica y del mismo modo que simula los procesos requeridos.
- Entregar al usuario las indicaciones y pasos requeridos a seguir (un algoritmo) para que la simulación de la gestión de datos sea la adecuada y cumpla su función.
- Que todo usuario que utilice el software sea capaz de cumplirse con las finalidades del mismo.

## **II. Introducción**

Este manual de usuario tiene como finalidad dar a conocer a los usuarios que utilicen el software las operaciones que brinda el software junto los pasos a seguir para el uso eficaz, para que dicho software “Analizador y reportador de datos” cumpla su objetivo. Por lo cual se dará breves explicaciones del manejo en la Interfaz de Usuario (UI) de las funcionalidades, así como el manejo de simulación y resultado, junto guía gráfica para su mayor comprensión.

Este software está orientado a la carga y gestión de datos previamente escritos en un documento para la creación de reportes en consola para dichos datos dentro de la aplicación y una página web autogeneradas para los reportes específicos tokens y errores.

### **III. Descripción del programa**

Es una aplicación enfocada a la lectura y salida de archivos a través de una Interfaz de Usuario (UI), donde dicho software se divide en tres partes, siendo la principal la lectura del archivo de entrada desplegándolo en la interfaz para que usuario pueda interactuar con dicha información, así mismo luego se procede a la lectura de dicha data siendo consumidas por un analizador léxico y consecuentemente un analizador sintáctico, siendo capaz de capturar errores en el documento y siendo capaz de consultarlos mediante reportes, ya siendo esta sección concluida se procede a los archivos de salida.

Según los requerimientos de los archivos de entrada el software procesará los datos de maneras específicas y siendo capaces de consultar los archivos de salida generados en este caso las respuesta de las instrucciones indicadas por el archivo de salida, así mismo como la exportación de reporte de la lectura tanto de tokens como de errores del archivo de entrada el cual debe cumplir con una estructura específica para el funcionamiento de este, el cual será presentado a continuación, además del árbol de derivación de la gramática.

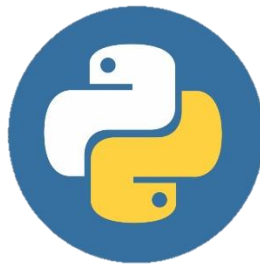
## **I. Requerimientos del sistema**

### **Software mínimo**

- 2 GB de RAM
- Windows vista
- Arquitectura 32 bits

### **Python 3**

Tener instalado Python3 ya que fue el lenguaje con el que se ejecuta “Analizador y reportador de datos”



## II. Requerimientos archivo de entrada

Para el funcionamiento de este los archivos de entrada debe cumplir con la estructura presentada en las siguientes figuras con extensión “flp”.

Se debe contener las Claves consecuente mente los Registros y para concluir las instrucciones a realizar, para las Claves la estructura es la siguiente:

### Ejemplar Claves:

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

Consecuentemente se deben ingresar los registros para las claves de la siguiente manera: donde los valores pueden ser de las siguientes formas “valor\_ingresado” o un dígito: 1,2...100 con la siguiente estructura:

### Ejemplar Registros

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

## Funciones

Siempre y cuando de hayan ingresado correctamente las claves y registros se podrán ejecutar las siguientes funcionalidades:

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 multilinea  
'''
```

## Instrucciones de Reportería:

- `imprimir(cadena)`: Imprime por consola el valor dado por la cadena en una misma línea.

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

- `imprimirln(cadena)`: Imprime por consola el valor dado por la cadena con un salto de línea

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

- `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
```

- `datos()`: Imprime por consola los registros leídos

```
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
```

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

titulo				
codigos	producto	precio_compra	precio_venta	stock
1	Barbacoa	10.50	20.00	6
2	Salsa	12.00	16.00	7
3	Mayonesa	15.00	18.00	4
4	Mostaza	14.00	16.00	4



## Ejemplar de una entrada con todas las funciones

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

Registros = [
    {1, "Barbacoa", 10.50, 20.00, 6}
    {2, "Salsa", 12.00, 16.00, 7}
    {3, "Mayonesa", 15.00, 18.00, 4}
    {4, "Mostaza", 14.00, 16.00, 4}
]
```

```
imprimir("Plantilla actual de: ");
imprimirln("FC Bayern Munchen");

imprimirln("conteos");
conteo();

imprimirln("promedio");
promedio("edad");

imprimirln("contar si");
contarsi("nacionalidad", "Alemán");

imprimirln("datos");
datos();

imprimirln("sumar edades");
sumar("edad");

imprimirln("mayor edad");
max("edad");

imprimirln("menor edad");
min("edad");

exportarReporte("titulo");
```

### Respuesta:

```
Plantilla actual de:
FC Bayern Munchen
conteos
20
promedio
No se encontro registro
contar si
No se encontro registro
datos
codigos  producto  precio_compra  precio_venta  stock
0      1  Barbacoa      10.50      20.00      6
1      2    Salsa      12.00      16.00      7
2      3  Mayonesa      15.00      18.00      4
3      4   Mostaza      14.00      16.00      4

sumar edades
No se encontro registro
mayor edad
No se encontro registro
menor edad
No se encontro registro
Se genero la exportacion del Reporte
```

titulo				
codigos	producto	precio_compra	precio_venta	stock
1	Barbacoa	10.50	20.00	6
2	Salsa	12.00	16.00	7
3	Mayonesa	15.00	18.00	4
4	Mostaza	14.00	16.00	4

# III. Operaciones del sistema

## 1. Pantalla Unica

### a. Carga

Se desplegará la única pantalla , ver Figura 1, con todas las opciones del sistema las cuales son:

- 1) Abrir: permite seleccionar el archivo a procesar, como se presenta en la Figura 2, de tipo LFP, dicha data se agregará en un editor de texto a) para poder editar dicha data.
- 2) Analizar: permite le permite segunda la data de la sección del editor de texto a) procesar la data y desplegar la respuesta en el visualizador de texto b).
- 3) Generar: permite generar un archivo solicitado según la opción seleccionada en las opciones del combobox c)
  - a. Los reportes de Tokens se generan la carpeta del proyecto específicamente dentro de la carpeta './Tokens.HTML'
  - b. Los reportes de Errores se generan la carpeta del proyecto específicamente dentro de la carpeta './Errores.HTML'
  - c. El árbol de derivacion se generan la carpeta del proyecto específicamente dentro de la carpeta './Diagramas/Arbol\_ReporteArbol\_Derivacion.jpg'
- 4) Salir: permite cerrar la aplicación.

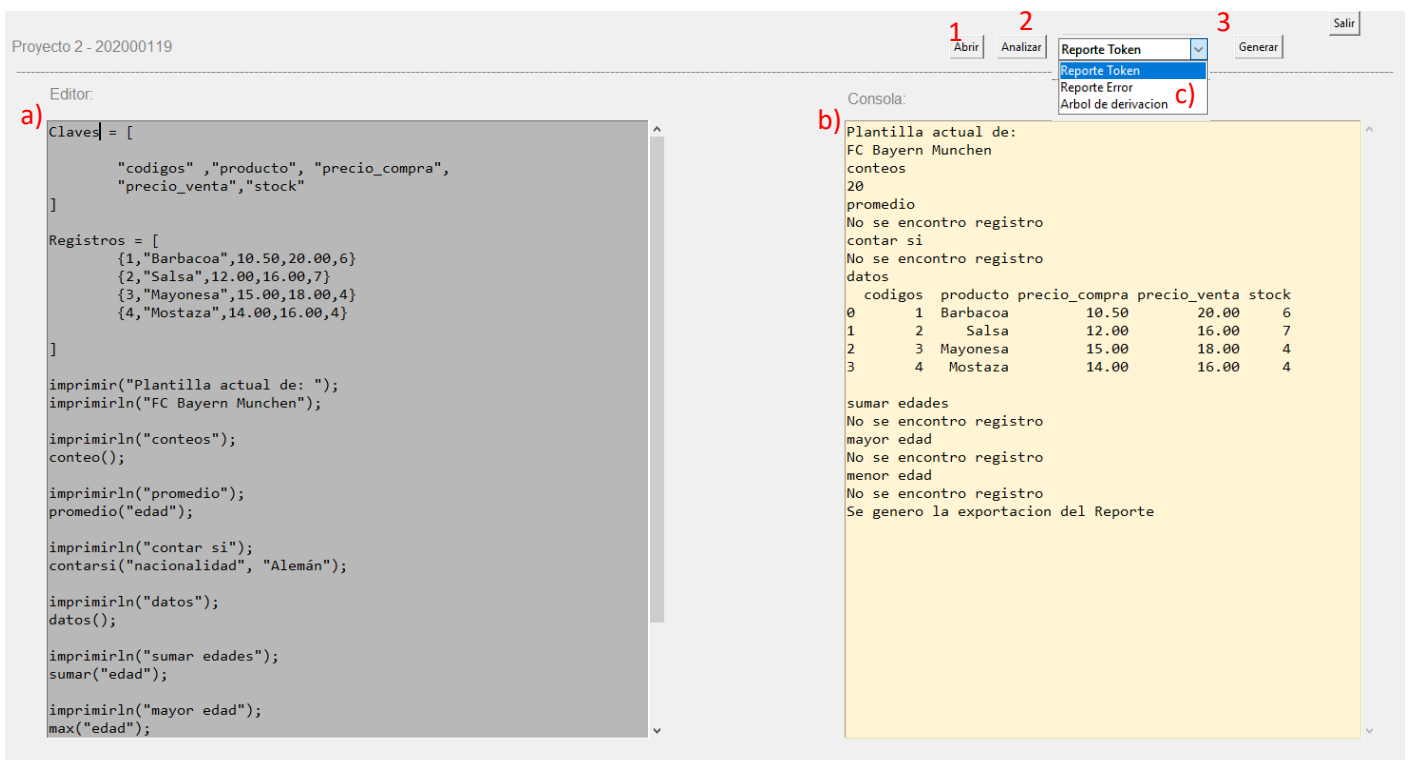


Figura 1: interfaz del sistema

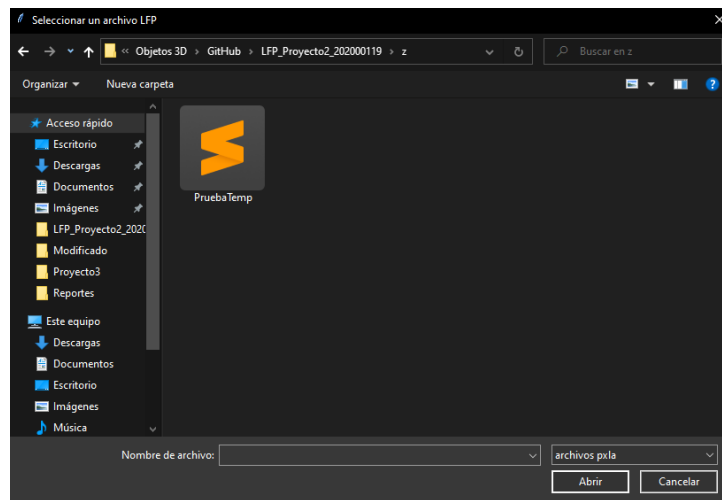


Figura 2: interfaz para seleccionar archivo de extensión “LFP”