

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS  
LENGUAJES FORMALES Y DE PROGRAMACION  
CATEDRÁTICA: ING. ASUNCION MARIANA SIC SOR  
AUXILIAR: ENRIQUE ALEJADRO PINLA QUIÑONEZ



# **LABORATORIO LENGUAJES FORMALES Y DE PROGRAMACION**

## **PROYECTO 2**

JOSÉ ROLANDO YAQUIAN PAZ

202201185

GUATEMALA, JUNIO 2024

# Introducción

Este manual técnico proporciona una guía detallada sobre el uso del software, desde la instalación y configuración inicial, hasta las instrucciones específicas para la carga, ordenamiento y gestión de datos. El objetivo es garantizar que cada usuario pueda utilizar el programa de manera efectiva, maximizando los beneficios de la digitalización y optimizando la operatividad interna de la empresa.

Para facilitar esta transformación, se ha desarrollado un programa de software especializado, diseñado para recibir y gestionar listas de datos proporcionadas por cada departamento de la empresa. Este software se basa en una sintaxis previamente definida, garantizando una estructura homogénea y simplificada para la carga y manipulación de datos.

# Objetivos

## General

Transformar la información física almacenada en la empresa a un formato digital, mejorando la accesibilidad y la seguridad de los datos.

## Específicos

- Desarrollo de un Programa de Software:

Crear un programa que permita la carga de listas de datos utilizando una sintaxis definida y homogénea.

- Funcionalidad de Ordenamiento de Datos:

Implementar algoritmos de ordenamiento que permitan clasificar los datos de manera eficiente y según criterios específicos.

Facilitar el acceso rápido y organizado a los datos digitalizados.

- Seguridad y Confidencialidad:

Desarrollar medidas de seguridad que protejan los datos durante el proceso de digitalización y en su almacenamiento digital.

- Optimización del Acceso a Datos:

Crear interfaces de usuario intuitivas que permitan búsquedas y consultas rápidas en la base de datos digital.

# Funcionalidades principales

La aplicación ofrece las siguientes funcionalidades clave:

## 1. Cargar archivo:

Permite seleccionar y cargar archivos con la extensión ".lfp" a la memoria del sistema, preparando el archivo para su posterior ejecución y análisis.

## 2. Ejecutar archivo:

Analiza el archivo de entrada previamente cargado y genera los archivos CSV de salida.

## 3. Archivos CSV:

Crea los archivos CSV después de haber analizado el archivo y haber ejecutado correctamente las instrucciones que venían en el archivo.

## 4. Reportes:

La aplicación genera y exporta reportes en formato HTML, permitiendo al usuario seleccionar la ubicación de almacenamiento. Los reportes disponibles incluyen:

- i. Reporte de tokens: Detalla el nombre del token, lexema, fila y columna.
- ii. Reporte de errores: Se crea una lista con los errores léxicos y sintácticos.}

# Lógica del programa

- **Interfaz de Usuario (UI) y Entrada de Datos:**

Formulario de Carga: Proporcionar una interfaz gráfica o de línea de comandos donde los usuarios puedan ingresar datos siguiendo una sintaxis definida.

- **Procesamiento de Datos:**

Parser: Un módulo que analiza la sintaxis de los datos ingresados, asegurando que cada elemento cumpla con las especificaciones y transformándolo en una estructura de datos interna adecuada.

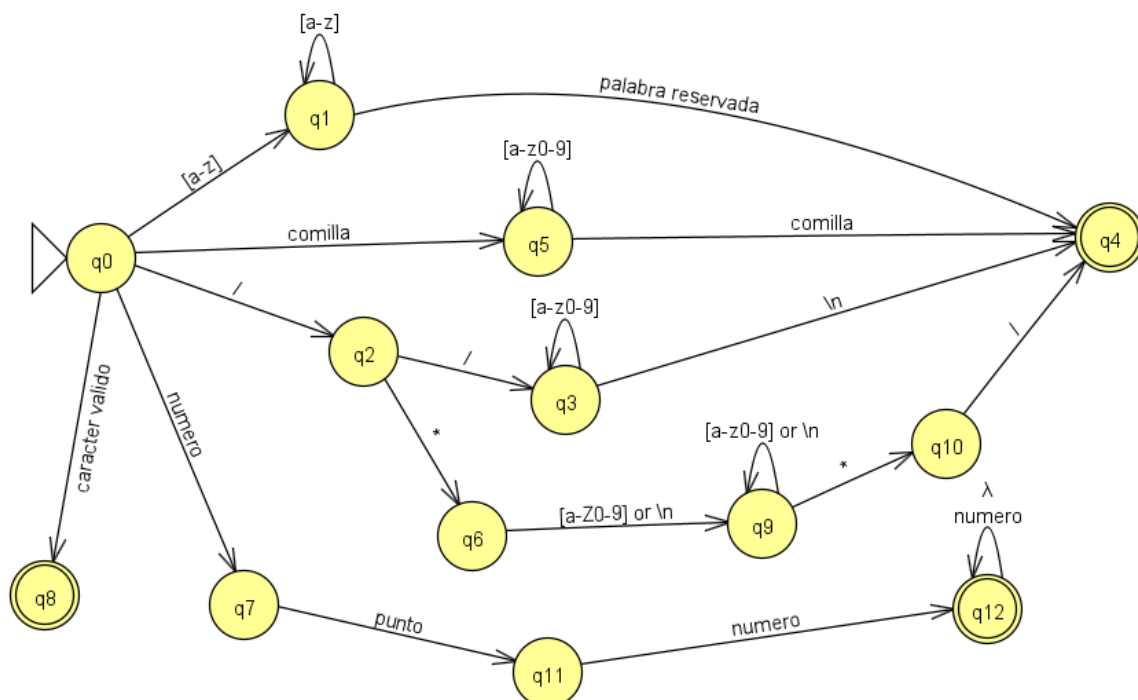
- **Almacenamiento de Datos:**

Mejora la eficiencia y la seguridad para guardar los datos de la empresa.

- **Ordenamiento de Datos:**

Algoritmos de Ordenamiento: Desarrollar y aplicar algoritmos de para organizar los datos según criterios específicos proporcionados por los usuarios.

AFD:



Gramática:

terminales: tk\_palabraArray

tk\_palabraNew

tk\_numero

tk\_signo

tk\_igual

tk\_id

tk\_string

tk\_metodo

tk\_punto

tk\_palabraSort

tk\_palabraSave

tk\_parentesisAbre

tk\_parentesisCierra

tk\_corcheteA

tk\_corcheteC

tk\_coma

tk\_PyC

tk\_palabraAsc

tk\_booleano

No terminales: <inicio>

<instruccion>

<instrucciones>

<declaracion>

<instruccionID>

<accionArreglo>

<listaElementos>

<masElementos>

<elemento>

<ordenamiento>

<guardar>

Inicio: <inicio>

producciones:

<inicio> ::= <instrucciones>

<instrucciones> ::= <instruccion> <instrucciones>

| epsilon

<instruccion> ::= <declaracion>

| <instruccionID>

<instruccionID> ::= tk\_id tk\_punto <accionArreglo>

<accionArreglo> ::= <ordenamiento>

| <guardar>

$\langle \text{declaracion} \rangle ::= \text{tk\_palabraArray} \text{ tk\_id} \text{ tk\_igual} \text{ tk\_palabraNew} \text{ tk\_palabraArray}$   
 $\text{tk\_corcheteA} \langle \text{listaElementos} \rangle \text{ tk\_corcheteC} \text{ tk\_PyC}$

$\langle \text{listaElementos} \rangle ::= \langle \text{elemento} \rangle \langle \text{masElementos} \rangle$

$\mid \text{epsilon}$

$\langle \text{masElementos} \rangle ::= \text{tk\_coma} \langle \text{elemento} \rangle \langle \text{masElementos} \rangle$

$\mid \text{epsilon}$

$\langle \text{elemento} \rangle ::= \text{tk\_string}$

$\mid \text{tk\_numero}$

$\langle \text{ordenamiento} \rangle ::= \text{tk\_palabraSort} \text{ tk\_parentesisAbre} \text{ tk\_palabraAsc} \text{ tk\_igual}$   
 $\text{tk\_booleano} \text{ tk\_parentesisCierra} \text{ tk\_PyC}$

$\langle \text{guardar} \rangle ::= \text{tk\_palabraSave} \text{ tk\_parentesisAbre} \text{ tk\_string} \text{ tk\_parentesisCierra}$   
 $\text{tk\_PyC}$



# Entorno de desarrollo

## **Lenguaje de programación:**

1. Python 3.12.1

## **Librerías utilizadas:**

1. Tkinter para la interfaz grafica
2. Graphviz para la generación de los grafos

## **Procesador:**

1. AMD Ryzen 5 5500U with Radeon Graphics 2.10 GHz

## **RAM instalada:**

1. 8.00 GB (7.35 GB usable)

## **Tipo de sistema:**

1. Sistema operativo Windows 11 de 64 bits, procesador basado en x64