

Sección	Catedrática	Tutor académico
A-	Inga. Vivian Damaris Campos Gonzáles	Luisa María Ortiz Romero
B-	Inga. Zulma Karina Aguirre Ordoñez	Jonathan Leonel García Arana
A+	Ing. Otto Amílcar Rodríguez Acosta	Gerson Rubén Quiroa del Cid
B+	Ing. David Estuardo Morales Ajcót	Herberth Abisai Avila Ruiz

## **Proyecto No.1**

### CONTENIDO

Objetivos .....	2
Descripción: .....	2
Definición del lenguaje.....	3
Bloque Grafica.....	3
Bloque Continente .....	3
Bloque Destino .....	4
Ejemplo archivo de entrada .....	5
Aplicación:.....	6
Gráfico:.....	7
Interfaz gráfica .....	9
Consideraciones .....	9
Flujo de la aplicación.....	10
Restricciones y Entregables .....	10
Notas importantes .....	11

## OBJETIVOS:

- Este proyecto tiene como objetivo que el estudiante practique el análisis léxico de un lenguaje formal y que comprenda y aplique los conceptos generales sobre el lenguaje de programación Fortran, así como aplicar los conocimientos adquiridos en la clase y en el laboratorio.
- Hacer que el estudiante pueda desarrollar y practicar sus conocimientos en teoría de autómatas y pueda aplicar los métodos de implementación de analizadores lexicográficos.
- Desarrollar la habilidad para identificar y manipular tokens o caracteres en archivos de entrada, procesándolos adecuadamente para respectivo análisis y manipulándolos para llevar a cabo los requerimientos.

## DESCRIPCIÓN:

Una empresa internacional desea poner una oficina en algún lugar del mundo donde tenga menor competencia en el mercado. Para esto tienen un archivo con extensión .ORG que desean visualizar de manera gráfica y lo han elegido a usted para desarrollar una aplicación que lo haga posible. La aplicación incluirá una interfaz gráfica desarrollada en Python utilizando la biblioteca Tkinter y para el análisis se utilizará el lenguaje de programación Fortran.

El archivo .ORG tiene un lenguaje propio definido en el que se pueden visualizar los lugares propuestos, su ubicación geográfica y un porcentaje de qué tan saturado está el mercado en dicho lugar.

El gráfico de las localizaciones se realizará a partir del contenido del editor de texto, este editor se puede llenar directamente o a través de la carga de un archivo con extensión .ORG y los gráficos serán mostrados en la interfaz gráfica.

## DEFINICIÓN DEL LENGUAJE:

Para el primer proyecto deberá aplicar el **método del árbol** para generar **un autómata finito determinista (AFD)**.

El lenguaje tiene varios bloques contenidos uno dentro de otro. A continuación, se describen estos bloques

### Bloque Grafica:

Es necesario definir la gráfica que utilizaremos dentro de este bloque. Solamente se tendrá un bloque Grafica por archivo. Su estructura es la siguiente

```
Grafica: {  
    nombre: <cadena>;  
    <Bloques Continente>  
}
```

Cuando el Bloque Grafica sea analizado se generará una tabla en el archivo HTML que muestre el nombre y los resultados del análisis del área de edición. Además, se mostrará la imagen de la gráfica generada en el espacio de grafica de la interfaz. Si contiene errores solamente se mostrará el archivo HTML de errores descrito en la sección de errores.

A continuación, se detallan los elementos de la instrucción destino:

- <cadena>: es una secuencia de caracteres delimitada por comillas dobles que denota el nombre del gráfico.
- <Bloques Continente>: puede venir uno o más bloques 'Continente' detallados a continuación.

### Bloque Continente

En este bloque se describe un continente que contendrá uno o más destinos:

- <cadena>: es una secuencia de caracteres delimitada por comillas dobles que denota el nombre del continente.
- <Bloques Destino>: Indica uno o más bloques destino definidos a continuación.

La sintaxis del Bloque Continente es la siguiente:

```
Continente: {  
    nombre: <cadena>;  
    <Bloques Destino>  
}
```

### Bloque Destino

En este bloque se describe uno de los destinos que pueden ser elegidos para poner la nueva sucursal. El destino será un país con datos que se detallan a continuación:

- <Nombre>: es una cadena que denota el nombre del país.
- <Población>: indica el número de personas en el país.
- <Saturación>: es un número entero del 0 al 100 que indica el porcentaje de saturación del área.
- <Imagen>: es una cadena que indica la ruta de la imagen de la bandera del país.

Su sintaxis es la siguiente:

```
País: {  
    nombre: <cadena>;  
    población:<Numero Entero>;  
    saturación: <Numero porcentaje>;  
    bandera: <Ruta>;  
}
```

A continuación, se detallan los elementos de la instrucción destino:

- <cadena>: es una secuencia de caracteres delimitada por comillas dobles.
- <Número porcentaje>: es un número entero del 0 al 100 que termina con el carácter '%'.
- <Número entero>: es un número entero.
- <Ruta>: es una cadena que contiene la ruta, delimitada por comillas dobles.

Dentro del bloque Grafica pueden venir uno o más bloques destino. Además, los datos que vengan dentro del bloque destino pueden venir en cualquier orden.

## EJEMPLO ARCHIVO DE ENTRADA

A continuación, se muestra un archivo de entrada sencillo como ejemplo:

```
grafica: {
  nombre:"nombre de grafico";
  continente {
    nombre: "Asia";
    país: {
      nombre: "Japón";
      población: 2352342;
      saturacion:80%;
      bandera: "C:\imagen.jpg";
    }
    país: {
      bandera: "C:\imagen.jpg";
      nombre: "China";
      población: 1350000000;
      saturacion:95%;
    }
    país: {
      saturacion:40%;
      población: 2352342;
      nombre:"Korea";
      bandera: "C:\imagen.jpg";
    }
  }
  continente {
    nombre: "América";
    país: {
      nombre: "Canadá";
      población: 23423423;
      saturacion:65%;
      bandera: "C:\imagen.jpg"
    }
    país: {
      población: 17263239;
      nombre: "Guatemala";
      saturacion:40%;
      bandera: "C:\guate.jpg"
    }
    país: {
      saturacion:60%;
      nombre:"Chile";
      población: 235234234;
      bandera: "C:\imagen.jpg"
    }
  }
}
```

## APLICACIÓN:

La aplicación deberá analizar el contenido del editor de texto en ese momento y en caso de encontrar un error léxico la aplicación deberá informar al usuario y mostrar el reporte de errores en una página HTML, en este caso no se deberá generar la página de salida (Tokens) ni la gráfica resultante. La estructura para la página de errores es la siguiente:

No	Error	Descripción	Fila	Columna
1	\$	Elemento Léxico desconocido	14	20

Si, al contrario, al analizar no se encuentra ningún error léxico deberá mostrar un HTML que contenga la siguiente información:

No	Lexema	Tipo	Fila	Columna
1	Grafica	Palabra reservada	1	1
2	"Nombre"	Cadena	1	7
3	:	Dos puntos	1	10

Además de la generación de reporte de Análisis léxico se deberá crear una imagen y mostrarla en el área de Grafico de la interfaz. A continuación, se explica cómo generar dicha grafica.

En la parte de elección tiene que salir el país que sea la mejor elección para ser el país donde pongan las nuevas oficinas de la empresa.

Debe salir el nombre, su población y la imagen de la bandera.

Los criterios para la selección son los siguientes:

- Primer criterio: **Menor saturación global**. Si hay un país con el menor porcentaje de saturación sin importar el continente este será el país seleccionado.
- Segundo criterio: **Menor saturación de continente**. Si hay dos países con el mismo nivel de saturación se mostrará cualquiera de los dos países que cumplan con los criterios.

## GRAFICO:

Se generará una gráfica (en formato PNG o SVG) con la herramienta graphviz que muestre la distribución de los países dependiendo de su continente, es decir la gráfica tendrá 3 niveles.

- El primer nivel contendrá el nombre de la grafica
- El segundo nivel contendrá los nombres de los continentes obtenidos en cada uno de los países del área del editor de texto
- El tercer nivel contendrá los países con la información de cada uno de ellos descrita.

Se deberá pintar cada país dependiendo del porcentaje de saturación que tenga siguiendo la siguiente tabla:

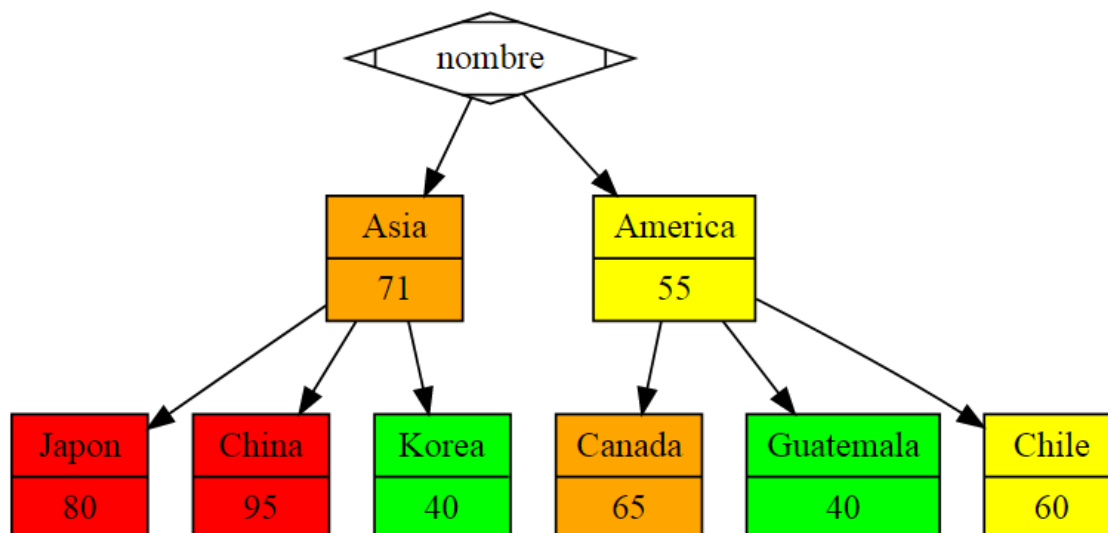
Saturación	Color de nodo
0% - 15%	Blanco
16% - 30%	Azul
31% - 45%	Verde
46% - 60%	Amarillo
61% - 75%	Anaranjado
76% - 100%	Rojo

También se deberá pintar el continente que contiene a los países haciendo un promedio de sus porcentajes de saturación y pintarlo siguiendo la tabla anteriormente descrita.

Es decir. Se sumarán los porcentajes de saturación de los países pertenecientes al continente y se dividirá en el número de países contenidos en el continente.

$$\text{Saturación continente} = \frac{\text{Suma saturación de países}}{\# \text{ de países}}$$

Este es un ejemplo de una gráfica de países en donde el primer campo es el nombre del país o continente y el segundo es el porcentaje de saturación encontrado.



Según esta grafica el país seleccionado sería Guatemala, entonces se muestra en el área de selección

País seleccionado:  
Guatemala  
Habitantes: 17263239



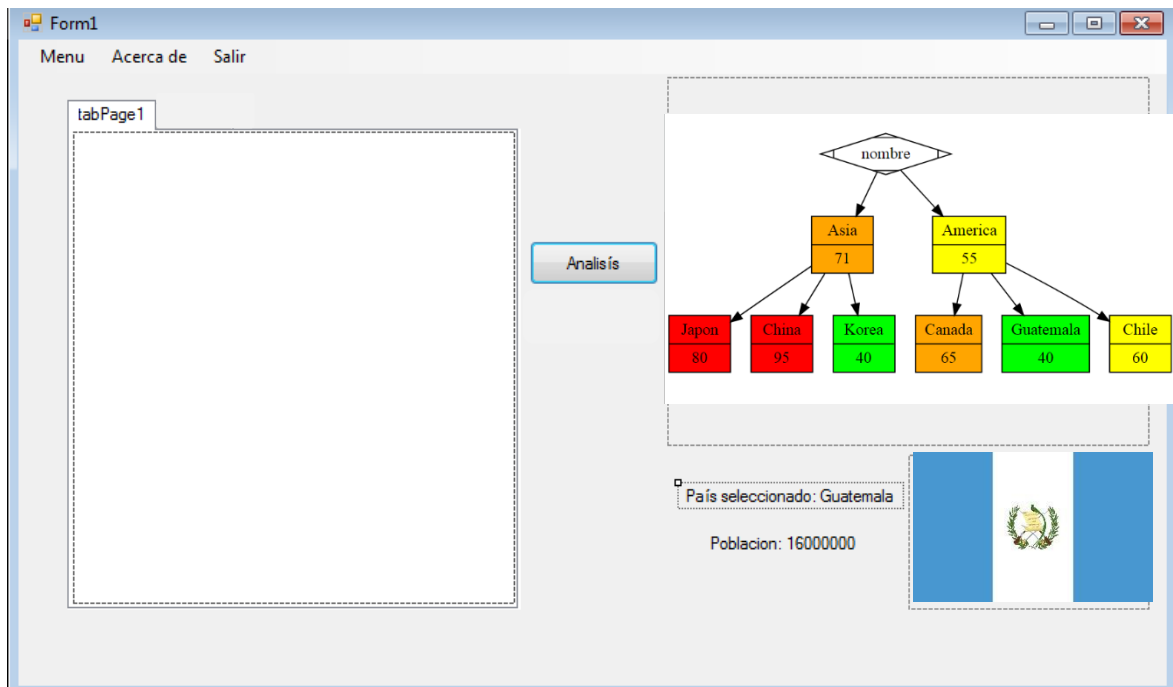


## INTERFAZ GRÁFICA:

### Componentes mínimos de la interfaz

1. Editor de texto: Debe ser un área de texto en el cual se puedan escribir nuevas instrucciones de entrada o bien cargar texto desde archivos con extensión .ORG
2. Menú: la Interfaz debe tener un menú en donde se pueda
  - Abrir: La interfaz debe proveer la capacidad de abrir archivos con extensión .ORG.
  - Guardar: La aplicación debe proveer la capacidad de guardar el texto contenido en el editor. Si no ha sido guardada nunca, debe proveer una opción para ingresar el nombre del nuevo archivo y la ubicación en la que se desea guardar
  - Guardar como: Está opción permite guardar el archivo de entrada con otro nombre, se debe preguntar el nombre del nuevo archivo.
3. Analizar: Debe realizar el análisis léxico del lenguaje que se encuentra actualmente en el editor y generar la salida correspondiente.
4. Acerca de: Debe desplegar una ventana con los datos del estudiante y del curso.
5. Área gráfica: acá se muestra la gráfica generada a partir del texto que se encuentra en el editor de texto, después de seleccionar la opción analizar. Esta grafica fue generada por Graphviz y la interfaz solo tiene que buscar el path en donde se encuentra para mostrarla, si no la encuentra no mostrar nada.
6. Área selección: Muestra el país seleccionado y su imagen. Esta grafica se busca por medio del path descrito en los atributos de País.
7. Salir: Debe terminar la ejecución de la aplicación

En el siguiente diagrama pueden ver un ejemplo de cómo se distribuye la interfaz gráfica.

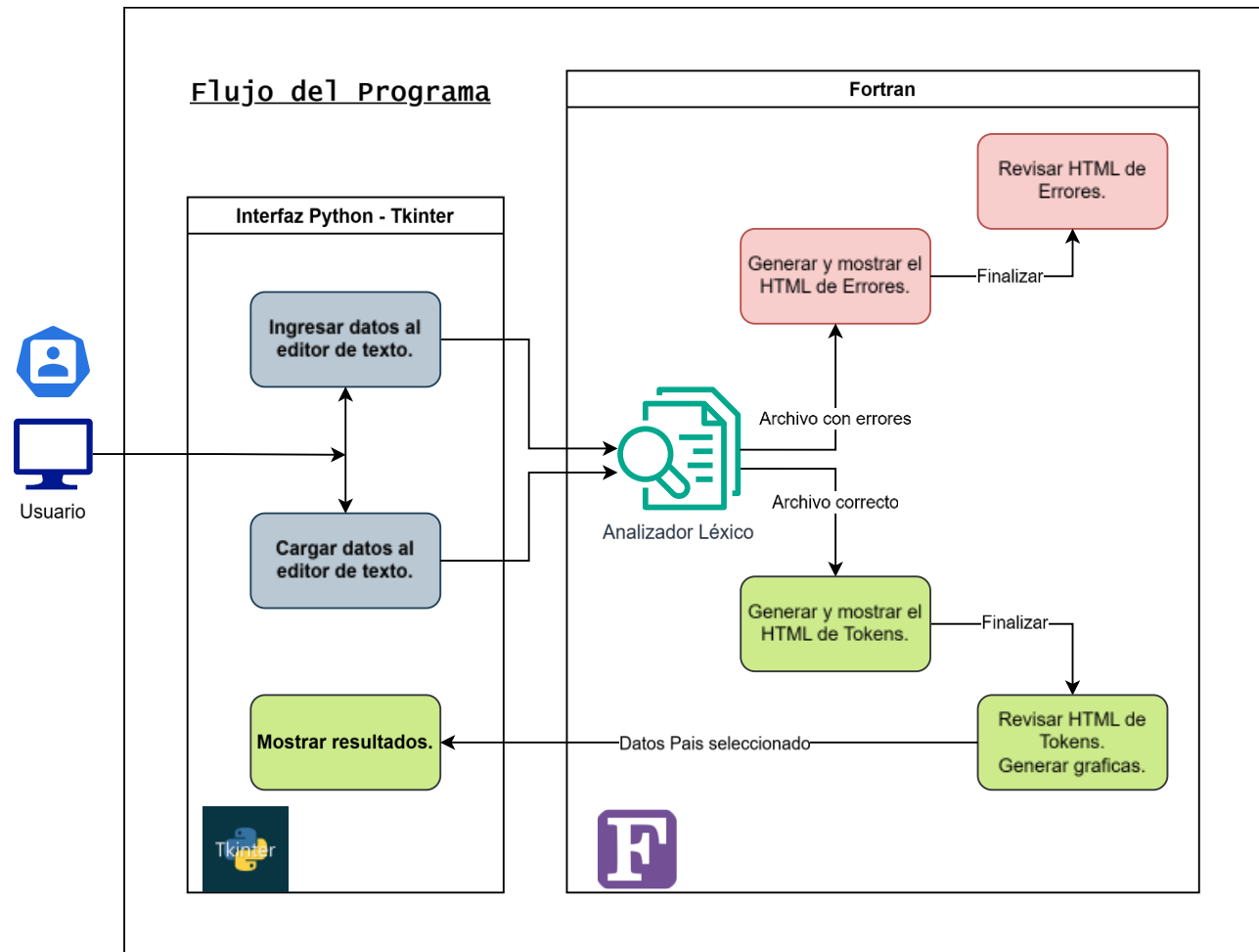


Esta es solamente una guía, no es necesario hacer la interfaz igual.

### CONSIDERACIONES

- El porcentaje de saturación solo aceptara números enteros entre 0-100.
- Al calcular el porcentaje de saturación del continente se redondeará al entero más cercano.
- Respetar el correcto funcionamiento del analizador léxico al reportar errores.

## FLUJO DE LA APLICACIÓN



## RESTRICCIONES:

- Prohibido el uso de expresiones regulares: No está permitido utilizar expresiones regulares para la manipulación y procesamiento de los datos en los archivos de entrada. Los estudiantes deben de implementar sus propias funciones para identificar y manipular tokens o caracteres.
- **Se calificará sobre el ejecutable**, sin ayuda de ningún tipo de herramienta o IDE.

## ENTREGABLES:

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

- Manual de Usuario. Tiene que explicar los pasos, no se permiten solo capturas de pantalla. En formato Markdown (.md).
- Manual Técnico, debe explicar la lógica que usó para el analizador léxico. No se permite solo capturas de pantalla del código. En formato Markdown (.md).
- Código Fuente de la aplicación.

### **Documentación que entregar de forma física el día de la calificación:**

- Hoja de calificación (Original y una copia)

### **Notas importantes:**

- El proyecto se debe desarrollar de forma individual.
- No se debe tomar en cuenta el uso de números negativos.
- Se debe de crear un repositorio privado en GitHub con el siguiente nombre: [LFP]2S24Proyectos\_<#carnet> y que contenga la carpeta Proyecto1.
- Agregar a su respectivo auxiliar a su repositorio de GitHub: Herberth3
- Este proyecto se deberá desarrollar utilizando el lenguaje de programación Fortran e interfaz en Tkinter.
- El proyecto será anulado si el estudiante utiliza Python para implementar el analizador léxico o el escáner.
- No está permitido el uso de librerías externas que ayuden a la lectura del archivo ni a la manipulación de la información a excepción de la librería Graphviz para generar los gráficos.
- Para la interfaz gráfica es obligatorio utilizar la biblioteca Tkinter de Python. No se permitirá el uso de otras bibliotecas o frameworks para esta parte del desarrollo.
- El proceso de obtener tokens se debe hacer a través de la implementación del autómata finito determinista desarrollado por el propio estudiante.
- No se puede agregar o quitar algún símbolo en el archivo de entrada. El proyecto deberá funcionar con los archivos de prueba que se disponga para la calificación, sin modificación.
- La calificación del proyecto será personal y PRESENCIAL y se estará firmando o escribiendo la frase "Estoy conforme con mi nota" en la hoja de calificación física y durará como máximo 20 minutos, en un horario que posteriormente será establecido. Se debe tomar en cuenta que durante la calificación no podrán estar terceras personas alrededor, de lo contrario no se calificará la práctica.
- No se aceptan entregas vía correo electrónico u otro medio.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación (1 día 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.
- **COPIA PARCIAL O TOTAL DEL PROYECTO 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 el proyecto; por lo cual, se tendrá una nota de cero puntos.**
- Fecha de entrega: 19 de septiembre de 2024, antes de las 23:59, no se recibirán entregas después de la fecha y hora establecida.

### **Anexos**

- Sitio oficial de Graphviz: <http://www.graphviz.org/>
- Código de graphviz del ejemplo: [Ejemplo](#)