

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LENGUAJES FORMALES DE PROGRAMACIÓN
PRIMER SEMESTRE 2024



Sección	Catedrático	Tutor Académico
A+	Ing. Otto Amilcar Rodríguez Acosta	Gerson Rubén Quiroa del Cid
A-	Inga. Vivian Damaris Campos González	Enrique Alejandro Pinula Quiñonez
B+	Ing. David Estuardo Morales Ajcot	Francisco Magdiel Asicono Mateo
B-	Inga. Zulma Karina Aguirre Ordoñez	Elder Anibal Pum Rojas

PROYECTO 1 TRADUCTOR DE HTML

Objetivos:

- Dar continuidad a la implementación de soluciones de software, así como conceptos generales sobre lenguajes formales tales como alfabeto, símbolos, cadenas y reglas.
- Implementar un software en Python que permitan plasmar los conocimientos sobre el analizador léxico, adquiridos en la unidad 2 del programa del curso de lenguajes formales y de programación.
- Implementar la programación orientada a objetos en Python.

Descripción:

El proyecto consiste en realizar un editor web simple el cual **generará** código HTML, la sintaxis que se usará en el editor se explica más adelante. Se debe poder modificar desde el editor el código que se cargara para generar código HTML.

Características del programa:

La aplicación de escritorio deberá estar ejecutándose y al ser consultada, mostrará una pantalla con dos cuadros de texto y dos botones, el primer botón será para “abrir” un archivo y el segundo botón para “traducir”. Cuando se abra el archivo se deberá cargar su contenido en el primer cuadro de texto y al presionar el botón “traducir” deberá enviar el texto al servidor que devolverá el código traducido, adicionalmente se deberá abrir el archivo HTML generado. En caso de que se encuentre un elemento no perteneciente al lenguaje se deberá mostrar un reporte con los caracteres no pertenecientes encontrados.

Diseño de la interfaz

El diagrama muestra la interfaz de usuario de una aplicación de escritorio titulada "Proyecto 1 Lenguajes". La interfaz tiene un diseño limpio con un fondo blanco y un borde azul. En la parte superior, hay una barra de título azul con tres botones de control (rojo, amarillo, verde) a la izquierda y el título "Proyecto 1 Lenguajes" a la derecha. El contenido principal está organizado en una cuadrícula. En la esquina superior izquierda, hay un botón rectangular con el texto "Abrir archivo". Debajo de este botón, a la izquierda, hay un cuadro de texto grande con el label "Texto de entrada:" a su izquierda. A la derecha de este cuadro, hay otro cuadro de texto grande con el label "Traducción:" a su izquierda. En la parte inferior central, hay un botón rectangular con el texto "Traducir".

Este solamente es un diseño sugerido, queda a discreción del estudiante la presentación que considere más adecuada. Para la implementación de la interfaz se deberá utilizar solamente la librería Tkinter.

ESTRUCTURA DEL ARCHIVO DE ENTRADA

Las estructuras de los archivos de entrada tendrán un formato similar a un archivo JSON. A continuación, se describen sus distintos apartados.

Bloque Inicio

Este bloque es la raíz principal del archivo de entrada. Dentro de este bloque se encontrará obligatoriamente el bloque “Encabezado” y el bloque “Cuerpo”.

```
Inicio:{
  Encabezado:{
    ... Atributos encabezado ...
  }
  Cuerpo:[
    ... Elementos cuerpo ...
  ]
}
```

```
Inicio:{
  Cuerpo:[
    ... Elementos cuerpo ...
  ]
  Encabezado:{
    ... Atributos encabezado ...
  }
}
```

Consideraciones:

Ambos bloques pueden venir en distinto orden.
Cada bloque (Encabezado y cuerpo) solo aparecerán una vez.

Bloque Encabezado

El bloque encabezado contiene el texto que se mostrará en la etiqueta title del head de los archivos html a generarse. Su estructura se muestra a continuación:

```
Encabezado:{
  TituloPagina:"Ejemplo titulo";
}
```

Bloque Cuerpo

Este bloque contendrá todos los elementos que deberán mostrarse en la sección Body de los archivos hmtl a generarse. La estructura principal se muestra a continuación:

```
Cuerpo:[
    ... Elementos cuerpo ...
]
```

Elementos Bloque Cuerpo

Estos elementos conforman el cuerpo del archivo HTML a generarse. Cada elemento a su vez tiene una definición de los atributos con los que deberá mostrarse.

A continuación, se definen cada uno de los elementos que pueden encontrarse en el cuerpo del archivo, luego se expone una vista general del bloque cuerpo y por último las consideraciones generales sobre el bloque cuerpo y sus elementos.

Titulo

Este elemento permite agregar títulos a los archivos html.

```
Titulo:{
    texto:"Este es un titulo";
    posicion:"izquierda";
    tamaño:"t1";
    color:"rojo";
}
```

Consideraciones:

Las posibles opciones para el atributo "posicion" son : izquierda, derecha, centro.

Los posibles tamaños estarán dados como "t1,t2,t3,t4,t5,t6".

Los posibles colores serán: rojo, amarillo, azul, o un color en hexadecimal.

Ejemplo: Color:" #FF0099";

Los atributos solo se especificarán una vez y pueden venir en desorden.

Fondo

Este elemento determina el color del fondo del archivo HTML.

```
Fondo:{  
    color:"#FF0099";  
}
```

Consideraciones:

Los posibles colores serán: rojo, amarillo, azul, o un color en hexadecimal.

Ejemplo: `Color:"amarillo";`

Este elemento tiene un único atributo.

Párrafo

Este elemento permite agregar párrafos a los archivos HTML.

```
Parrafo:{  
    texto:"Este es un parrafo de ejemplo.";  
    posicion:"izquierda";  
}
```

Consideraciones:

Las posibles opciones para el atributo "posicion" son : izquierda, derecha, centro.

Los atributos solo se especificarán una vez y pueden venir en desorden.

Texto

Este elemento permite agregar texto con formato.

```
Texto:{  
    fuente="Arial";  
    color="azul";  
    tamaño="11";  
}
```

Consideraciones:

Los posibles colores serán: rojo, amarillo, azul, o un color en hexadecimal.

Ejemplo: `Color:" #FF0099";`

Los atributos solo se especificarán una vez y pueden venir en desorden.

Código

Este elemento permite agregar texto con un tipo de fuente similar al de los editores de código fuente.

```
Codigo:{
  texto:"Muestra el texto con fuente de codigo de ordenador.";
  posicion:"centro";
}
```

Consideraciones:

Las posibles opciones para el atributo “posicion” son: izquierda, derecha, centro.
Los atributos solo se especificarán una vez y pueden venir en desorden.

Formatos clásicos de texto

Los siguientes elementos permiten darle el formato correspondiente al texto que contienen:

Negrita

```
Negrita:{
  texto:"Este texto aparecerá en negrita.";
}
```

Subrayado

```
Subrayado:{
  texto:"Este texto aparecerá Subrayado.";
}
```

Tachado

```
Tachado:{
  texto:"Este texto aparecerá tachado.";
}
```

Cursiva

```
Cursiva:{
  texto:"Este texto aparecerá en cursiva.";
}
```

Salto

Este elemento permite agregar saltos en el archivo html.

```
Salto:{
    cantidad:"5";
}
```

Consideraciones:

El atributo cantidad especifica el número de saltos que se insertarán.

Tabla

Contiene la información necesaria para crear una tabla en el documento HTML.

```
Tabla:{
    filas:"4";
    columnas:"3";
    elemento:{"fila":"1","columna":"1","Texto mostrado en fila 1 columna 1"};
    elemento:{"fila":"1","columna":"2","Texto mostrado en fila 1 columna 2"};
    elemento:{"fila":"1","columna":"3","Texto mostrado en fila 1 columna 3"};
}
```

Consideraciones:

Los atributos “filas” y “columnas” se especificarán una vez y pueden venir en desorden.

El atributo elemento puede venir n veces, dado que especifica el contenido de cada una de las celdas de la tabla. Las celdas que no tengan definido un elemento deberán aparecer en blanco.

Cada atributo “elemento” estará definido con una ubicación correcta. No será necesario validar si la celda existe.

CONSIDERACIONES GENERALES DEL BLOQUE CUERPO

1. Todos los elementos descritos anteriormente conforman los elementos del bloque cuerpo.
2. Los elementos pueden venir en desorden y pueden aparecer más de una vez. Por ejemplo, pueden venir 10 párrafos, 5 títulos, 3 tablas, etc.
3. A continuación, se muestra la presentación del archivo de entrada, observe como cada uno de los atributos del bloque cuerpo se encuentran separados por coma y todo el bloque se encuentra encerrado por corchetes.

VISTA GENERAL DEL ARCHIVO

```
Inicio:{
  Encabezado:{
    TituloPagina:"Ejemplo titulo";
  },
  Cuerpo:[
    Titulo:{
      texto:"Este es un titulo";
      posicion:"izquierda";
      tamaño:"t1";
      color:"rojo";
    },
    Fondo:{
      color:"cyan";
    },
    Parrafo:{
      texto:"Este es un parrafo de ejemplo.";
      posicion:"izquierda";
    },
    Texto:{
      fuente="Arial";
      color="azul";
      tamaño="11";
    },
    Codigo:{
      texto:"Muestra el texto con fuente de codigo de ordenador.";
      posicion:"centro";
    },
    Negrita:{
      texto:"Este texto aparecerá en negrita.";
    },
    Subrayado:{
      texto:"Este texto aparecerá Subrayado.";
    },
    Tachado:{
      texto:"Este texto aparecerá tachado.";
    },
    Cursiva:{
      texto:"Este texto aparecerá en cursiva.";
    },
    Salto:{
      cantidad:"5";
    }
  ]
}
```


Análisis del archivo de entrada

Para analizar el archivo de entrada no será permitido el uso de librerías, funciones específicas para extracción de cadenas o herramientas de análisis, el estudiante deberá implementar una solución basada en **autómatas finitos deterministas**, que permita recorrer carácter a carácter todo el archivo de entrada buscando los patrones que le permitan organizar la información.

Salidas

Las salidas del programa serán las siguientes:

- Código HTML generado (sin errores).
- Listado de Tokens y lexemas encontrados (incluir línea, columna y no incluya errores) en un archivo HTML.
- Listado de elementos no pertenecientes al lenguaje (carácter, línea, columna) en un archivo HTML.

Reporte general de elementos encontrados

Token	Lexema	Línea	Columna
Cadena	"Ejemplo titulo"	3	23
Llave abrir	{	2	17

Reporte de caracteres no permitidos

Carácter	Línea	Columna
@	3	23
%	2	17

Entregables:

- **Manual técnico en PDF:** En este manual se debe especificar la lógica utilizada para el funcionamiento del programa, con la ayuda de la herramienta **Graphviz** se deberán generar los grafos correspondientes a los **AFD** utilizados para reconocer el lenguaje.
- **Manual de usuario en PDF**
- **Código fuente del programa**

Notas importantes:

- El proyecto se debe desarrollar de forma individual.
- Este proyecto se deberá desarrollar utilizando Python, Tkinter y Graphviz.
- **La entrega se realizará en la plataforma UEDI.**
- **Se deberá de crear un repositorio en GitHub para llevar un control de la realización del proyecto con el nombre LFP_S1_2024_PROYECTO1_#CARNET.**
- No será permitida la modificación de archivos de entrada durante la calificación.
- No se puede utilizar ninguna herramienta para generar analizadores (como ply o similares).
- La calificación deberá ser en línea y se estará grabando la reunión para tener un respaldo de la forma en que se procedió.
- La calificación de la práctica será personal y durará como máximo 15 minutos, en un horario que posteriormente será establecido.
- El estudiante es responsable del horario que elija para calificarse, en caso de no poder presentarse deberá notificar al auxiliar con suficiente anticipación 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.
- No se calificará el proyecto sobre la terminal de la computadora, únicamente sobre la interfaz gráfica.
- **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 la practica; por lo cual, se tendrá una nota de cero puntos.**

Fecha de entrega: 01 de abril de 2024 antes de las 23:59.