

# Manual de Usuario, Proyecto 1, Lenguajes formales y de Programación, Sección B+

Por: Abner Misael Paredes de la Cruz, carné 201222632

## Índice:

<i>Elemento</i>	<i>Página</i>
1. Portada.....	1
2. Introducción .....	3
3. Formato de Entrada .....	4
4. Formato de salida.....	5
5. Guardado de Entrada y Salida.....	6
6. Consulta y guardado de tokens.....	

# Introducción

El propósito del proyecto ha sido mediante una entrada similar a los archivos .JSON generar un código html básico, utilizando una máquina de estados finita determinista derivada de la expresión regular que es facilitado por la entrada de formato .JSON.

La salida de este programa, es en medida de lo posible el código que al guardarse, se lee como un documento html con las etiquetas básicas.

# Formato De Entrada

El formato de entrada es similar a un archivo JSON, ejemplo:

```
{
  "Inicio": {
    "Encabezado": { "TituloPagina": "Ejemplo Titulo" },
    "Cuerpo": {
      "Array": [
        {"Titulo": {"Tamaño": "t1", "Posicion": "centro", "Color": "red", "Text": "texto"} },
        {"Parrafo": "Esto es un texto"}
      ]
    }
  }
}
```

Las entradas serán escritas en la parte izquierda del programa, y en la única que está habilitada para la modificación directa del usuario.

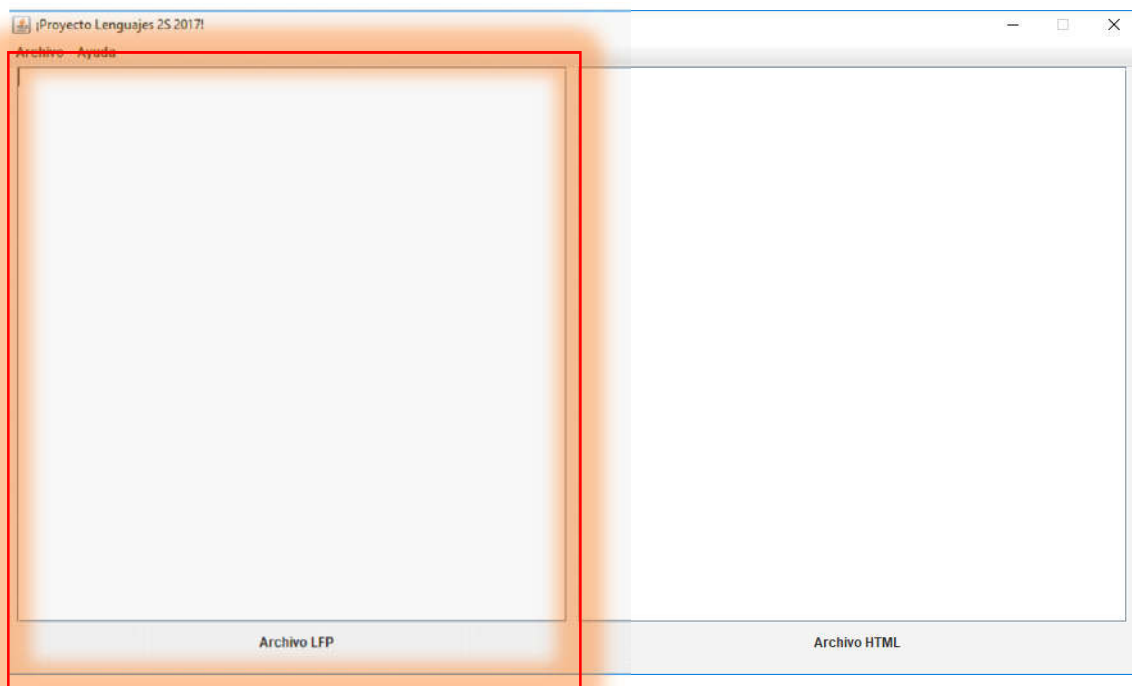


Figura 1

Fuente: Aplicación de escritorio correspondiente al proyecto de LFP, S2 2017

Como se puede apreciar en la figura 1, el área anteriormente mencionado, corresponde a la encerrada en el recuadro rojo.

# Formato De Salida

El formato de salida de la aplicación de escritorio es de formato HTML, para acceder a este, es necesario utilizar el botón de “Traducir” ubicado en el menú Archivo de la aplicación. Se muestra la posición de el botón en la Figura2.



Figura 2

Fuente: Aplicación de escritorio correspondiente al proyecto de LFP, S2 2017

Como se mencionó anteriormente, la entrada es similar a los archivos JSON pero, no son archivos JSON, son archivos “LFP” utilizado únicamente para esta aplicación.

# Guardado de Entrada y Salida

Para la continuación con el trabajo creado anteriormente, existen distintos tipos de almacenamiento de los datos modificados a través de la aplicación, estos son:

- Guardar LFP
  - Guarda todos los caracteres del área asignada para la entrada de datos en la aplicación, si se ha abierto desde un archivo previamente existente, los datos se guardarán en este, en caso contrario, se abrirá la interfaz para elegir el nombre con el cual se guardarán los datos. Este se guardará en un archivo con terminación .LFP.
- Guardar HTML
  - Guarda todos los caracteres del área asignada para la salida de datos en la aplicación, estos se guardarán en formato HTML, listo para ser abierto por un navegador.
- Guardar Como LFP
  - Guarda todos los caracteres del área asignada para la entrada de datos en la aplicación, esto lo guardara en un archivo completamente distinto al abierto, y definirá este archivo nuevo como el “abierto” en el cual se guardaran por defecto los datos con el método “Guardar LFP”
- Guardar como HTML
  - Guarda todos los caracteres del área asignada para la salida de datos de la aplicación, estos se guardaran en formato HTML, sin importar si ha habido un archivo creado o modificado anteriormente.
- Guardar LFP y HTML
  - Ejecuta el método “Guardar LFP” seguido inmediatamente por el método “Guardar HTML”
- Guardar como LFP y HTML
  - Ejecuta el método “Guardar como LFP” seguido inmediatamente por el método “Guardar como HTML”



# Consulta y Guardado de Tokens

Al ser la aplicación únicamente de traducción de un formato a otro, los tokens son sensibles únicamente al código en sí, para que el usuario tenga acceso a estos, se ha resuelto guardar todos los tokens y su información en un archivo html, cuya dirección será establecida por el usuario que lo solicite.

