



MANUAL TÉCNICO

Lenguajes Formales y de Programación B+

Andrea Alejandra Pérez Sandoval
202201136



Código Proyecto 1 B+

Clase “inicio.py”

Clase principal, en la que se encuentra el main para inicializar/correr el programa. Cuanta con distintas importaciones como:

- Tkinter
- Filedialog
- re
- Analizador (clase)

Estas importaciones son importantes para la visualización de una interfaz, abrir el explorador de archivos, utilizar los atributos de las clases donde se almacenan los datos y para llamar utilizar expresiones regulares dentro del código.

Funciones:

- (): Esta función almacena los datos del estudiante, junto a una opción para presionar cualquier tecla, y luego de esta siga con el funcionamiento normal.
- (): Función para abrir la ventana emergente para buscar el archivo a utilizar (.lfp), leer el archivo, quitar espacios innecesarios entre datos, e indicar como se separan y manejan los datos dentro del documento.
- (): Muestra las opciones del menú, la función input para ingresar el número de la opción deseada, se manda a llamar a la clase lista_peliculas, a la cual se le nombra ListaPeliculas. Se crea un ciclo while verdadero, dentro de el encontramos condicionales if, los cuales, dependiendo de la opción ingresada, muestran submenús, mandan a llamar métodos de la clase lista_peliculas, y muestran los datos solicitados, crean gráficas; realizarán distintas acciones, dependiendo de lo solicitado.

Por último, mandamos a llamar a las funciones datosEstudiante() y menu() para que el programa corra correctamente.

```

inicio.py > ...
1  import tkinter as tk
2  from tkinter import filedialog, messagebox
3  from Analizador import instruccion_inicio
4  import re
5
6  ventana = tk.Tk()
7  ventana.title("Proyecto 1 Lenguajes Formales y de Programación")
8
9  contenido_archivo = ""
10
11 def abrir_archivo():
12     global contenido_archivo
13
14     file_name = filedialog.askopenfilename(filetypes=[("Archivos LFP", "*.lfp")])
15     if file_name:
16         with open(file_name, 'r', encoding="utf-8") as file:
17             contenido = file.read()
18             print(contenido)
19             txt_entrada.delete(1.0, tk.END)
20             txt_entrada.insert(tk.END, contenido)
21
22     lexemas, errores = instruccion_inicio(contenido)
23     mostrar_lexemas(lexemas)
24     mostrar_errores(errores)
25     print("LEXEMAS DESPUÉS DE ABRIR ARCHIVO")
26     print(lexemas)
27     print("ERRORES DESPUÉS DE ABRIR ARCHIVO")
28     print(errores)
29
30     txt_entrada.delete(1.0, tk.END)
31     txt_entrada.insert(tk.END, contenido)
32     txt_salida.delete(1.0, tk.END)
33
34 #Función para extraer los datos del contenido del archivo .lfp
35 def extraer_datos_contenido(archivo_lfp):
36     with open(archivo_lfp, 'r', encoding='utf-8') as f:
37         contenido = f.read()
38
39         datos_contenido = {}
40
41         etiquetas = ["Titulo", "Fondo", "Parrafo", "Texto", "Codigo", "Negrita", "Subrayado", "Tachado", "Cursiva", "Salto", "Tabla"]
42
43         for etiqueta in etiquetas:
44             patron_etiqueta = re.compile('{:}\s*{{{(.*)}}}'.format(etiqueta), re.DOTALL)
45             coincidencias = patron_etiqueta.findall(contenido)
46
47             if coincidencias:
48                 datos_contenido[etiqueta] = []
49                 for match in coincidencias:
50                     datos = re.findall(r'(\w+)(?:=|:)(?:"([^\"]*)"|(\w+))', match)
51                     datos_dict = {clave: valor.strip("'") for clave, valor, _ in datos}
52                     datos_contenido[etiqueta].append(datos_dict)
53
54     return datos_contenido

```