

proyecto 1

Manual técnico

Marvin obidio perez Larios

201903712

Introducción

En el siguiente manual se explica cómo funciona el programa de forma técnica, el programa 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.

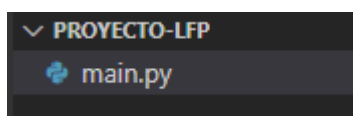
Objetivos

- implementar un software en Python que permitan plasmar los conocimientos sobre el analizador léxico
- Implementar la programación orientada a objetos en Python.

Requisitos

- Visual Studio
- python
- Graphviz
- Tkinter

Estructura del programa: todas las clases creadas



Esta parte del código es para abrir archivos

```
def abrir_archivo(self):  
    archivo = filedialog.askopenfilename(filetypes=[("Archivos JSON", "*.json")])  
    if archivo:  
        with open(archivo, "r") as f:  
            contenido = f.read()  
            self.cuadro_texto1.delete(1.0, "end")  
            self.cuadro_texto1.insert("end", contenido)
```

Esta parte es para traducir

```

def traducir(self):
    contenido = self.cuadro_texto1.get(1.0, "end").strip()
    if contenido:
        try:
            datos = json.loads(contenido)
            if "Encabezado" in datos and "TituloPagina" in datos["Encabezado"]:
                html_generado = self.generar_html(datos)
                self.cuadro_texto2.delete(1.0, "end")
                self.cuadro_texto2.insert("end", html_generado)

                tokens = self.generar_tokens(datos)
                self.mostrar_tokens(tokens)
            else:
                print("Error: Estructura JSON incorrecta.")
        except json.JSONDecodeError as e:
            print("Error de formato JSON:", e)

```

En esta parte del código genera el html

```

def generar_html(self, datos):
    html = "<!DOCTYPE html>\n<html>\n<head>\n<meta charset='UTF-8'>\n<title>{}\n</title>\n</head>\n<body>\n".format(datos["TituloPagina"])
    for elemento in datos["Cuerpo"]:
        if "Titulo" in elemento:
            html += "<h1>{}\n".format(elemento["Titulo"]["texto"])
        elif "Fondo" in elemento:
            html += "<div style='background-color:{}'>\n".format(elemento["Fondo"]["color"])
        elif "Parrafo" in elemento:
            html += "<p>{}\n".format(elemento["Parrafo"]["texto"])
    html += "</body>\n</html>"
    return html

```

Esta parte genera los tokens

```

def generar_tokens(self, datos):
    tokens = []
    for elemento in datos["Cuerpo"]:
        for clave, valor in elemento.items():
            tokens.append((clave, valor))
            if isinstance(valor, dict):
                for subclave, subvalor in valor.items():
                    tokens.append((subclave, subvalor))
    return tokens

```