Interfaz compilador

```
# /src/gui/compiler_gui.py
# import sys
# sys.path.append('/home/hugo/Documents/Tec/compiler/src/main_compiler.py')

import tkinter as tk
from tkinter import filedialog
from main_compiler import first_pass, second_pass
import csv

class CompilerGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Compileradar GUI")

        self.select_file_button = tk.Button(root, text="Seleccionar Archivo", command=self.select_file)
        self.select_file_button.pack(pady=10)

        self.compile_button = tk.Button(root, text="Compilar", command=self.compile)
        self.compile_button.pack(pady=10)

        self.file_path = None

    def select_file(self):
        file_path = filedialog.askopenfilename(title="Seleccionar Archivo", filetypes=[("Archivos de
Texto", "*.txt")])
        if file_path:
            self.file_path = file_path
            print(f"Archivo seleccionado: {self.file_path}")

    def save_symbol_table_to_csv(self, symbol_table, file_name):
        csv_data = []
        for lexema, tipo in symbol_table.items():
            csv_data.append({
                'Lexema': lexema,
                'Tipo': tipo
            })

        csv_file_path = f"{file_name}_symbol_table.csv"
        with open(csv_file_path, 'w', newline='') as csv_file:
            fieldnames = ['Lexema', 'Tipo']
            writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

            writer.writeheader()
            writer.writerows(csv_data)

        print(f"Tabla de Símbolos guardada en {csv_file_path}")
```

```
    def compile(self):
        if self.file_path:
            with open(self.file_path, "r") as file:
                # line = file.readline()
                # print(line.split())
                source_code = [line.split() for line in file.readlines()]

            symbol_table = first_pass(source_code)
            self.save_symbol_table_to_csv(symbol_table, file_name=self.file_path)

            error_table = second_pass(source_code)

            # Aquí puedes hacer algo con las tablas generadas
            # print("Tabla de Símbolos:", symbol_table.symbol_table)
            # print("Tabla de Errores:", error_table.error_table)
        else:
            print("Por favor, selecciona un archivo antes de compilar")

def main():
    root = tk.Tk()
    app = CompilerGUI(root)
    root.mainloop()

if __name__ == "__main__":
    main()
```