

## Relatório técnico - Atividade de Participação 03.

Alunos: Isaac Silva Santos Ramos e Pedro Gonçalves Neto.

### 1. Objetivo

Implementar um analisador léxico que recebe como entrada um arquivo texto (programa Python) e gera como saída um arquivo texto, contendo a lista de tokens e o tipo de cada token, de acordo com a ordem de definição no arquivo de entrada. Exemplo:

Para a entrada abaixo:

```
for i in range (1,100):
    if i%3 == 0 or i%5==0:
        print (i)

penultimo , ultimo = 0, 1
soma = 0
while ultimo <= 4000000:
    if ultimo%2 == 0:
        soma = soma + ultimo
    penultimo, ultimo = ultimo, penultimo+ultimo
print (soma)
```

a saída deve ser:

Token	Tipo
for	P.R. (Palavra Reservada)
i	ID (Identificador)
in	P.R.
range	P.R.
(	Símbolo
1	Número
,	Símbolo
100	Número
)	Símbolo
:	Símbolo
if	P.R.
i	ID
%	Símbolo
3	Número
==	Símbolo
0	Número
or	P.R.
i	ID
%	Símbolo
5	Número
==	Símbolo
0	Número
:	Símbolo
print	P.R.
(	Símbolo
i	ID
)	Símbolo
...	.....

### 2. Implementação

Para o programa em questão, foi utilizada a seguinte gramática:

```

lexer grammar expr2022;

PR: 'True'|'False'|'for'|'in'|'range'|'None'|'and'|'or'|'not'|'if'|'else'|'elif'|
    '|while'|'do'|'continue'|'break'|'input'|'main'|'def'|'float'|'int'|
    '|bool'|'string'|'class'|'as'|'try'|'catch'|'exception'|'import'|'from'|'return'|
    '|list'|'dict'|'tuple'|'print';

ID: [a-zA-Z][a-zA-Z0-9]*;
NUM: [0-9]+[.]?[0-9]*;

SB: |'|'?'|'/'|'!'|'#'|'%'|'+'|'-'|'*'|'='|'<'|'>'|'(''|')'|'[''|']'|'{''|'}'|
    '|<='|'>='|'!='|'*'|':'|';'|'|'.'|';

WS: [ \t\r\n] -> skip;

```

Já no arquivo Python, existem alguns passos para a execução do analisador. A princípio, tem-se a configuração da interface gráfica que servirá para receber a entrada do usuário, essa configuração foi feita a partir da biblioteca *Tkinter*, e está definida da seguinte forma:

```

if __name__ == '__main__':
    window = Tk()
    window.iconbitmap('pusheen.ico')
    window.title('Lexer Analyzer')          # objeto Interface
    window.geometry('700x700')
    window.resizable(True, True)

    editorText = Text(height=20, font=("Consolas", 18))
    editorText.config(bg='#333333', fg='#b08102', insertbackground='white')
    editorText.pack(fill=X)

    out = Text(height=window.winfo_reqheight())
    out.config(bg='#333333', fg='#249903')
    out.pack(fill=X)

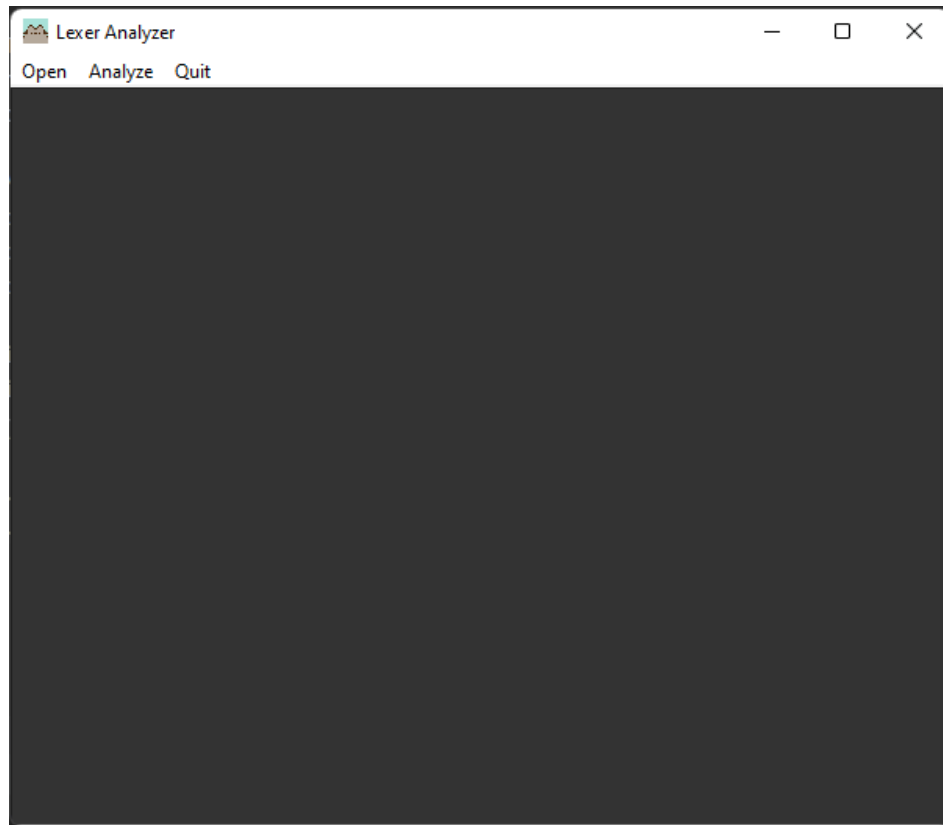
    bar = Menu(Menu(window), tearoff=0)
    bar.add_command(label='Open', command=abrirArquivo)
    bar.add_command(label='Analyze', command=analise)
    bar.add_command(label='Quit', command=exit)

    sizegrip = ttk.Sizegrip(window)
    sizegrip.pack(side="right", anchor=SE)
    #sizegrip.grid(row=1, sticky=SE)

    window.config(menu=bar)
    window.mainloop()

```

Dessa forma, uma janela se abrirá para digitar a entrada:



Nessa janela, é possível perceber três botões, são eles: “*Open*” que servirá para buscar um arquivo de entrada, “**Analyze**” que analisará o texto digitado como entrada abaixo e “*Quit*” que fechará o programa. A análise está definida de forma a receber a entrada, passar pelo analisador da gramática e recuperar a lista de *tokens*, após isso, será exibida a lista de tokens e seus respectivos tipos, além disso, essa listagem será salva em um arquivo nomeado como “lexica.txt”. *Observação*: evitar o uso de aspas simples, pois foi definido como obrigatório o uso de aspas duplas para comandos como `print(“ texto ”)`.

- **Implementação da Análise**

```
def analise():
    out.delete('1.0', END)
    code = editorText.get('1.0', END)
    data = InputStream(code)
    lexer = expr2022(data)
    tokens = CommonTokenStream(lexer)
    list_tokens = lexer.getAllTokens()

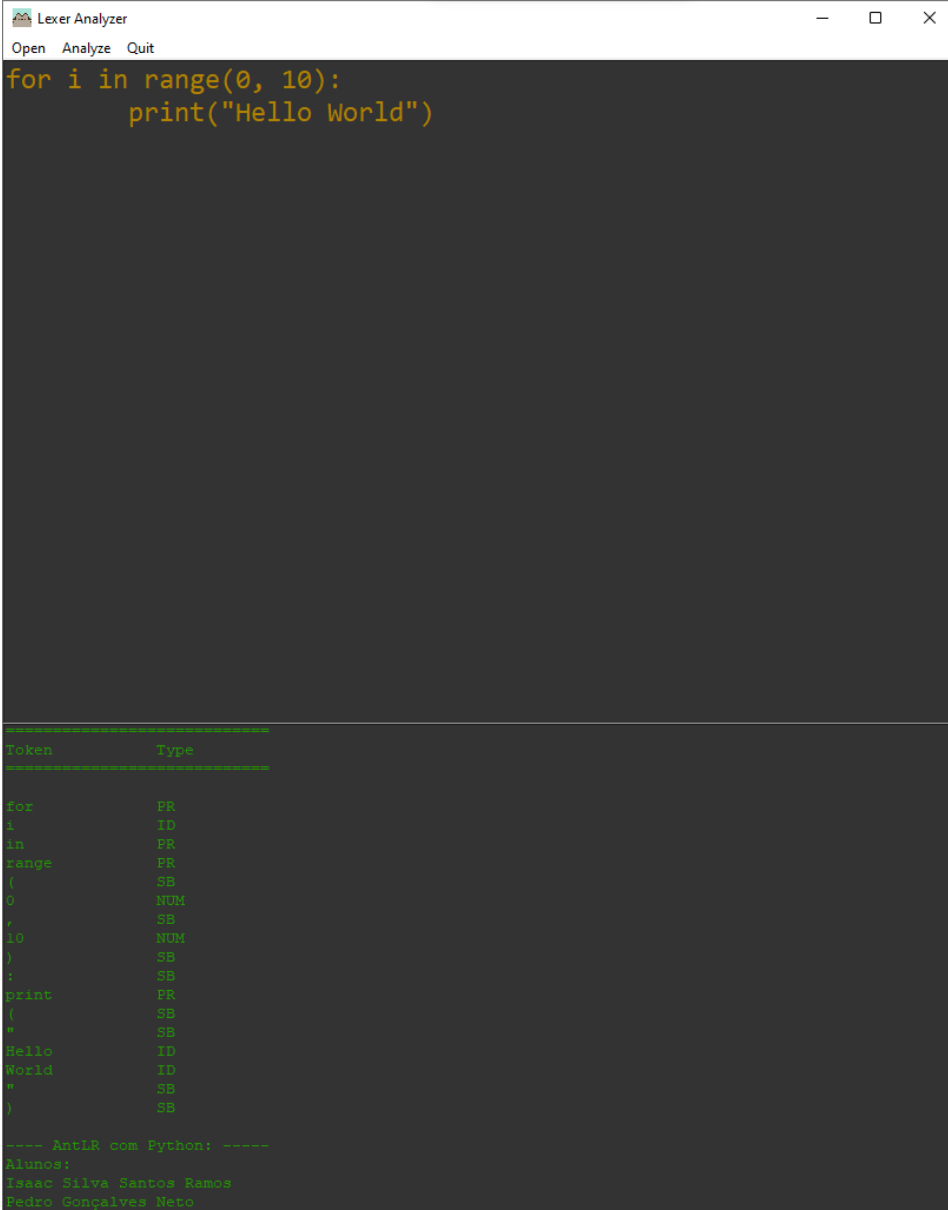
    file = open('lexica.txt', 'w+')
    file.write("=====\n")
    file.write("{:<15} {:<8}\n".format('Token', 'Type'))
    file.write("=====\n")

    for token in list_tokens:
        nomeToken = expr2022.symbolicNames[token.type]
        file.write("\n{:15} {:<8}".format(token.text, nomeToken))

    out.insert('1.0', '\n\n--- AntLR com Python: ----\nAlunos:\nIsaac Silva Santos Ramos\nPedro Gonçalves Neto')

    file.seek(0)
    out.insert('1.0', 'Process finished with exit code 0\n' + file.read())
    lexer.reset()
    file.close()
```

- Exemplo de entrada e saída



The image shows a window titled "Lexer Analyzer" with a menu bar containing "Open", "Analyze", and "Quit". The main text area contains the following Python code:

```
for i in range(0, 10):  
    print("Hello World")
```

Below the code, the output of the lexer is displayed as a table with two columns: "Token" and "Type".

Token	Type
for	PR
i	ID
in	PR
range	PR
(	SB
0	NUM
,	SB
10	NUM
)	SB
:	SB
print	PR
(	SB
"	SB
Hello	ID
World	ID
"	SB
)	SB

At the bottom of the window, there is a footer section with the text:

```
---- AntLR com Python: ----  
Alunos:  
Isaac Silva Santos Ramos  
Pedro Gonçalves Neto
```