

# Manipulador de Gramáticas e Transformações GLC

Êndril Castilho da Silveira, Leonardo Pellegrini Silva

Disciplina de Linguagens Formais

Departamento de Informática – UNISC

Campus de Santa Cruz do Sul

96.810-206 - Santa Cruz do Sul - RS - Brasil

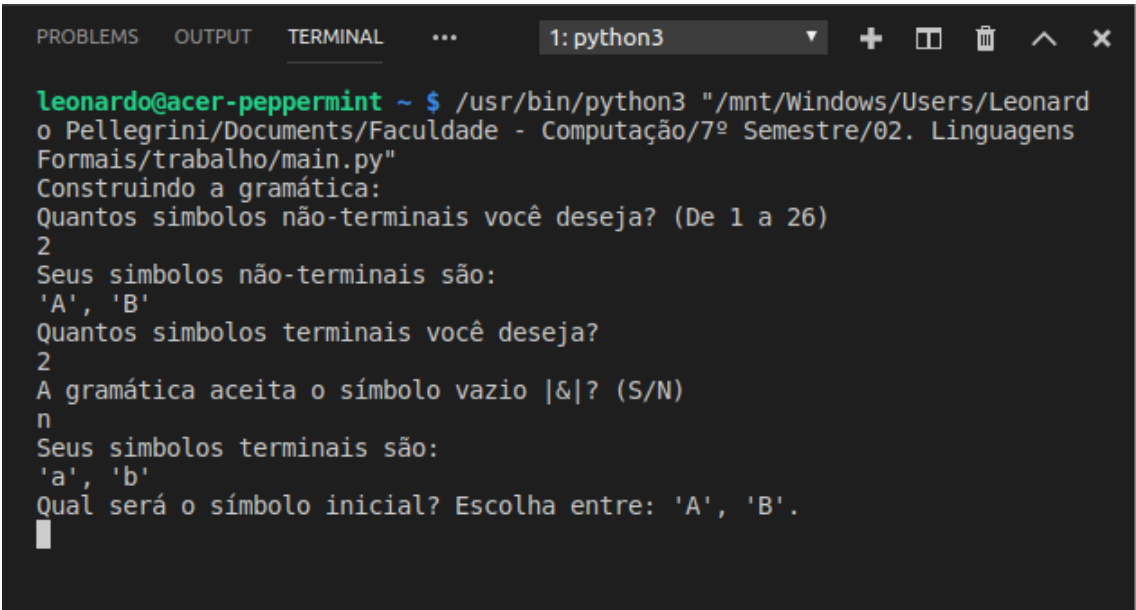
[endrilcs@mx2.unisc.br](mailto:endrilcs@mx2.unisc.br), [leonardopellegrini@mx2.unisc.br](mailto:leonardopellegrini@mx2.unisc.br)

## 1. Sobre o projeto

Desenvolvido em Python 3, o aplicativo é capaz de reconhecer uma gramática, visualiza-la, verificar sua consistência, identificar seu tipo de acordo com a hierarquia de Chomsky, gerar sentenças e transformá-la em um autômato finito (caso seja regular). Além disso, a aplicação é capaz de aplicar as transformações GLC em entradas compatíveis. Dentre as transformações que o programa está apto a realizar estão: símbolos inúteis, produção unitária, &-livre, fatoração e recursão à esquerda.

## 2. Modo de usar

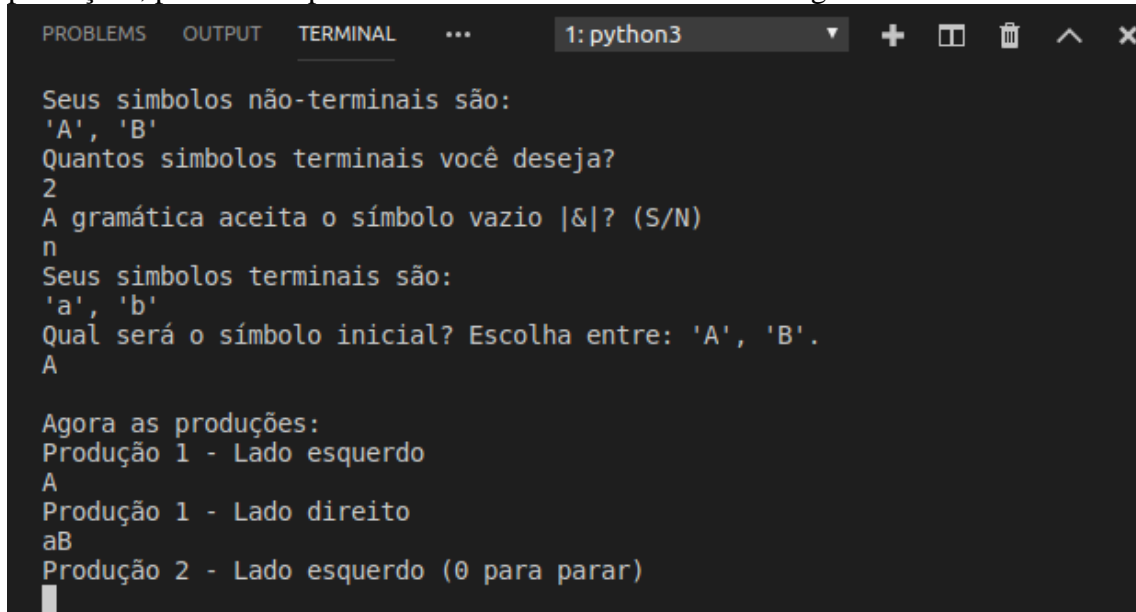
Primeiramente, deve-se inserir quantos símbolos não-terminais e terminais serão utilizados, se a gramática aceita o símbolo vazio e qual dos símbolos não-terminais será o símbolo inicial.



```
PROBLEMS OUTPUT TERMINAL ... 1: python3 + [icon] [icon] [icon] [icon] [icon]
leonardo@acer-peppermint ~ $ /usr/bin/python3 "/mnt/Windows/Users/Leonard
o Pellegrini/Documents/Faculdade - Computação/7º Semestre/02. Linguagens
Formais/trabalho/main.py"
Construindo a gramática:
Quantos simbolos não-terminais você deseja? (De 1 a 26)
2
Seus simbolos não-terminais são:
'A', 'B'
Quantos simbolos terminais você deseja?
2
A gramática aceita o símbolo vazio |&|? (S/N)
n
Seus simbolos terminais são:
'a', 'b'
Qual será o símbolo inicial? Escolha entre: 'A', 'B'.
█
```

Imagem 1. Construção da gramática.

Logo após, deverão ser informadas as produções da gramática, informando o lado esquerdo, e em seguida, o lado direito de cada produção. Quando informar todas as produções, pressione 0 para terminar e exibir os dados sobre a gramática inserida.



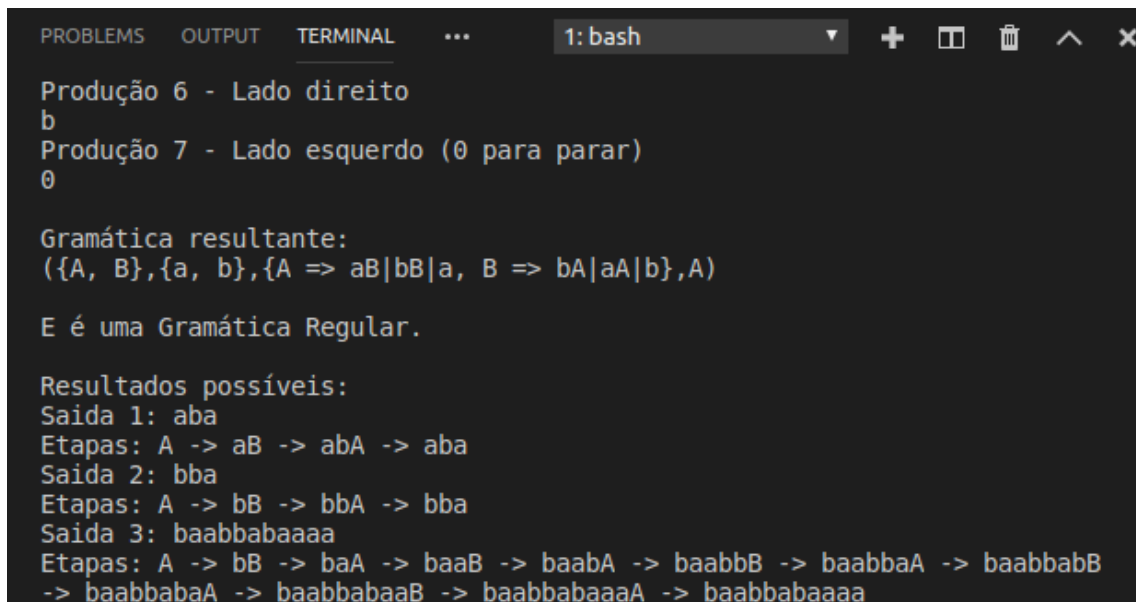
```
PROBLEMS OUTPUT TERMINAL ... 1: python3
Seus simbolos não-terminais são:
'A', 'B'
Quantos simbolos terminais você deseja?
2
A gramática aceita o símbolo vazio |&|? (S/N)
n
Seus simbolos terminais são:
'a', 'b'
Qual será o símbolo inicial? Escolha entre: 'A', 'B'.
A

Agora as produções:
Produção 1 - Lado esquerdo
A
Produção 1 - Lado direito
aB
Produção 2 - Lado esquerdo (0 para parar)

```

*Imagem 2. Inserção das Produções.*

São exibidos na tela a gramática resultante, o tipo de gramática e três possíveis sentenças criadas pela gramática. Nessa exibição, as produções já passaram pela as etapas de transformação GLC.



```
PROBLEMS OUTPUT TERMINAL ... 1: bash
Produção 6 - Lado direito
b
Produção 7 - Lado esquerdo (0 para parar)
0

Gramática resultante:
({A, B},{a, b},{A => aB|bB|a, B => bA|aA|b},A)

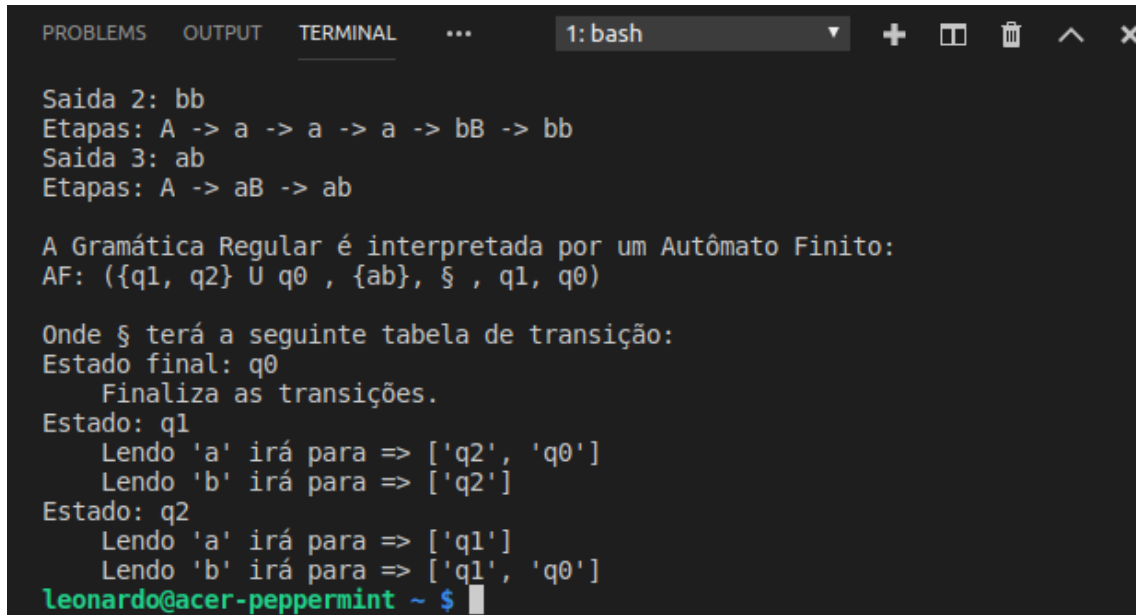
E é uma Gramática Regular.

Resultados possíveis:
Saida 1: aba
Etapas: A -> aB -> abA -> aba
Saida 2: bba
Etapas: A -> bB -> bbA -> bba
Saida 3: baabbabaaaa
Etapas: A -> bB -> baA -> baaB -> baabA -> baabbB -> baabbbaA -> baabbabB
-> baabbabaA -> baabbabaaB -> baabbabaaaA -> baabbabaaaa

```

*Imagem 3. Exibição dos dados sobre a gramática.*

Por fim, é exibido o tipo de autômato que reconhece essa gramática, e, caso seja uma gramática regular, será exibido o autômato finito que a interpreta.



```
PROBLEMS OUTPUT TERMINAL ... 1: bash
Saida 2: bb
Etapas: A -> a -> a -> a -> bB -> bb
Saida 3: ab
Etapas: A -> aB -> ab

A Gramática Regular é interpretada por um Autômato Finito:
AF: ({q1, q2} U q0 , {ab}, § , q1, q0)

Onde § terá a seguinte tabela de transição:
Estado final: q0
Finaliza as transições.
Estado: q1
Lendo 'a' irá para => ['q2', 'q0']
Lendo 'b' irá para => ['q2']
Estado: q2
Lendo 'a' irá para => ['q1']
Lendo 'b' irá para => ['q1', 'q0']
leonardo@acer-peppermint ~ $
```

Imagem 4. Exibição do autômato finito.