

# ReadMe

Para o desenvolvimento do analisador Sintático e Léxico, utilizamos a biblioteca `PLY` ( Python Lex-Yacc), framework desenvolvido por estudantes do curso de introdução aos compiladores da Universidade de Chicago.

## Requerimentos

o projeto não exige muitas dependências, além do python 3 instalado na máquina é preciso instalar a dependência do `PLY` que já está definida no arquivo de requerimentos.

```
pip install -r requirements.txt
```

## Rodando

- primeiro crie um arquivo `.pyuai` contendo o código a ser compilado, e adicione no mesmo diretório que o projeto.

```
--ex1
-- ex1.pyuai
-- analisador.py
--ex2
-- ex2.pyuai
-- analisador.py
--ex2
-- ex2.pyuai
-- analisador.py
```

exemplo de um código de teste:

```
caso (35 == 35) intao
  @ Exemplo de código mais básico do PyUai

  intero x = 5
  intero y = 7
  intero z = x + y
  bool varBool = ((x + 2 == 7) e (5 * (2 + 3) > 4))

  caso ((z % 2) == 0) intao
    print Numero_Par
    z = 3
```

```
naodeu
    print Numero_Impar
;

caso z % 2 != 0 intao e varBool
    print Mudou_para_Impar
;
;
```

Antes de executar verifique se o diretório do seu arquivo pyuai está coerente e atualize aqui:

```
with open('ex1/ex1.pyuai', 'r') as file:
    data = file.read()
```

- após isso basta executar o código:

```
python3 analisador.py
```

- com a execução teremos a geração de alguns arquivos:
  - `parse.out`: Contém informações de depuração sobre o processo de compilação. Pode incluir mensagens informativas, avisos ou erros encontrados durante a geração do analisador.
  - `parse.tab`: Este arquivo é o arquivo de tabela gerado pelo PLY durante a compilação. Ele contém informações sobre as regras gramaticais, tokens e outras estruturas necessárias para realizar o parsing do código.
  - `token.txt`: apresenta o reconhecimento de tokens da parte léxica de forma separada.