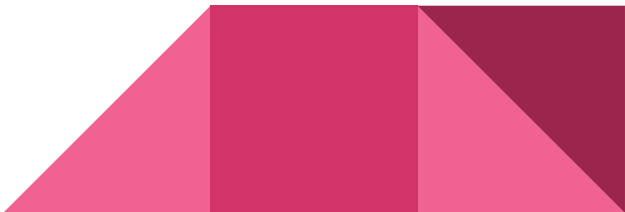


APS - Lógica da Computação

2019.1

Elisa Malzoni

Objetivos

- Criar linguagem de programação;
 - Melhorar a legibilidade;
 - loops, condições, definições em português, assim quem não sabe inglês pode aprender a programar de forma mais didática e rápida.
- 

Inspiração

- Essa linguagem criada é uma adaptação para o português da linguagem de VBA;
- Também para compilação foi usado o compilador feito nas aulas, com algumas modificações nos TOKENS (arquivo main.py).

BNF

```
Program = {SubDec | FuncDec}
```

```
SubDec = subrotina, identifier, '(', [identifier,  
como, Type,{',', identifier, como, Type}], ')',  
'\n', {statement, '\n'}, fim, subrotina;
```

```
FuncDec = funcao, identifier, '(', [identifier,  
como, Type], [',', identifier, como, Type], ')',  
como, Type, '\n', {statement, '\n'}, fim, funcao;
```

```
Statement =  $\lambda$  | ( identifier, '=', Rel. Exp. ) |  
(imprima, Rel. Exp.) | (inicialize, identifier,  
como, type) | (enquanto, Rel. Exp, '\n',  
{statement, '\n'}, enquantofim) | (se, Rel. Exp.,  
entao, '\n', {statement, '\n'}, [senao, '\n',  
{statement, '\n'}] , fim, se) | (chame, identifier,  
'(', [Rel. Exp. {'', Rel. Exp.}], ')');
```

```
Type = inteiro | booleano;
```

```
Rel. Exp. = Expression, [ ('=' | '>' | '<') ,  
expression ];
```

```
Expression = Term, { ('+' | '-' | 'ou') , Term };
```

```
Term = Factor, { ('*' | '/' | 'e') , Factor };
```

```
Fator = número | (verdadeiro | falso) | identifier  
['(', [Rel. Exp. {'', Rel. Exp.}], ')'] | (( '+'  
| '-' | 'nao') , Factor) | ( '(' , Rel. exp. , ')' )  
| input;
```

Demonstração da linguagem

```
Funcao dobra(n como inteiro) como inteiro
    dobra = n*2
fim Funcao
```

```
Subrotina menorquinze(n como inteiro)
    se n < 15 entao
        imprima 1
    senao
        imprima 2
    fim se
fim Subrotina
```

```
Subrotina Main()
    imprima dobra(5)
    Chame menorquinze(4)
fim Subrotina
```

A linguagem possui:

- variáveis;
- condicional;
- loops;
- funções (com recursão);
- permite entrada;
- imprime.



Utilizando o compilador da Linguagem

Para compilar e rodar um programa:

```
python3 main.py programa
```



Flex e Bison

Para utilizar flex e bison utilize os seguintes comandos:

```
flex tokens.l
```

```
bison linguagem.y
```

A criação do arquivo `tokens.l` foi feita a partir dos tokens da linguagem. Já a do `linguagem.y` foi feita pela BNF da linguagem.