

# Linguagem para Manipulação de Grafos

Marcelo Otaviano - 573124

Murilo Guedes Toloni - 489433

Iago Bernardes - 407569

Cristiano Ornelas - 488925

# Apresentação Geral

- Linguagem simples e intuitiva
- Escrita em português
- Grafos e Multigrafos, direcionais ou não direcionais
- Exibição do grafo em tela
- Guardar grafo em arquivo .png

# Aplicação pretendida

- Facilitar a manipulação de grafos
- Abstrair funções da biblioteca em python
- Representações gráficas dos grafos criados

# Esboço da gramática

```
1 grammar LinguagemLMG;
2
3
4
5 Criar Grafo grafo
6 Criar Grafo Direcional grafo
7 Exibir Grafo grafo
8 Exibir Grafo Direcional grafo
9
10 Novo Vertice vertice Em grafo
11
12
13 IDENT = ('a'..'z'|'A'..'Z') ('a'..'z'|'A'..'Z'|'_'|'0'..'9')*;
14 CADEIA = '"' ~('\r'|'\n'|'"')* '"';
15 NUM = ('0'..'9')+ ( '|' ('0'..'9')+);
16
17 arquivo:
18     CADEIA
19 ;
20
21 var:
22     IDENT
23 ;
24
25 key:
26     IDENT
27 ;
28
29 value :
30     NUM
31     | CADEIA
32 ;
33
```

```
74 comparacao_key_vertice :
75     key_vertice '==' value
76     | key_vertice '!=' value
77     | key_vertice '>=' value
78     | key_vertice '<=' value
79     | key_vertice '<' value
80     | key_vertice '>' value
81 ;
82
83 key_aresta :
84     'Valor De' key 'Na Aresta' aresta 'Em' grafo
85 ;
86
87 comparacao_key_aresta :
88     key_aresta '==' value
89     | key_aresta '!=' value
90     | key_aresta '>=' value
91     | key_aresta '<=' value
92     | key_aresta '<' value
93     | key_aresta '>' value
94 ;
95
96 Dados dados
97 Dado dado
98 Vertices vertice
99
100
101 Adicionar Vertices vertices Em grafo
102 Adicionar dado Ao vertice Em grafo
103 Adicionar dados Ao vertice Em grafo
104
105
106 Exibir Vertice vertice Em grafo
107 Exibir Vertices Em grafo
108 Exibir Vertices Em grafo Onde (expressao)
```

# Exemplos

```
1  inicio-GR
2
3  criar grafo G
4
5  vertices v : [a,b,c,d,e,f]
6
7  adicionar vertices v em G
8
9      dados dmurilo : nome=murilo, peso=100, idade=24
10     dados djoao : nome=Joao, peso=62, idade=12
11     dados dcarlos : nome=Carlos, peso=77, idade=32
12     dados dnickson : nome=Nickson, peso=54, idade=34
13     dados dstephanie : nome=Stephanie, peso=45, idade=55
14     dados ddiego : nome=Diego, peso=70, idade=33
15
16     adicionar dmurilo ao vertice a em G
17     adicionar djoao ao vertice b em G
18     adicionar dcarlos ao vertice c em G
19     adicionar dnickson ao vertice d em G
20     adicionar dstephanie ao vertice e em G
21     adicionar ddiego ao vertice f em G
22
23     adicionar aresta a1: a--b em G:
24     adicionar dados relacao="irmão" a aresta a1: a--b em G
```

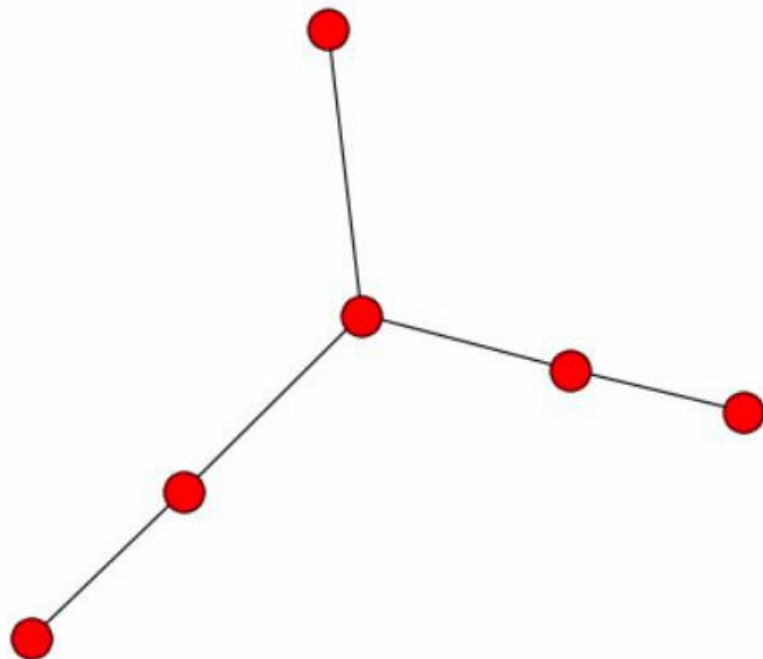
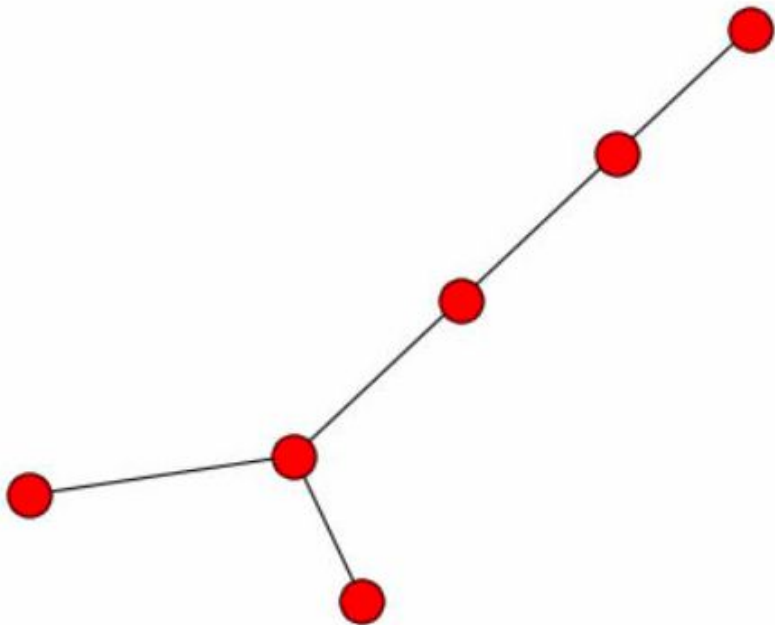
```
25
26     adicionar aresta a2: a--c em G
27     adicionar dados relacao="primo" a aresta a2: a--c em G
28
29     adicionar aresta a3: c--d em G
30     adicionar dados relacao="amigos" a aresta a3: c--d em G
31
32     adicionar aresta a4: d--e em G
33     adicionar dados relacao="amigos" a aresta a4: d--e em G
34
35     adicionar aresta a5: d--f em G
36     adicionar dados relacao="irmão" a aresta a5: d--f em G
37
38     Desenhar G em tela
39
40     Salvar tela em "relacoes.png"
41
42     Plotar tela
43
44
45  fim-GR
```

# Exemplos

```
1 import networkx as nx
2 import matplotlib.pyplot as plt
3
4 G = nx.MultiGraph()
5 v = ['a','b','c','d','e','f']
6 G.add_nodes_from(v)
7
8
9
10
11 G.add_node('a', nome='murilo', peso=100, idade=24)
12 G.add_node('b', nome='Joao', peso=62, idade=12)
13 G.add_node('a', nome='Carlos', peso=77, idade=32)
14 G.add_node('a', nome='Nickson', peso=54, idade=34)
15 G.add_node('a', nome='Stephanie', peso=45, idade=55)
16 G.add_node('a', nome='Diego', peso=70, idade=33)
17
18 G.add_edge('a','b',key='a1')
19 G.add_edge('a','b',key='a1',relacao="Irmão")
20
21 G.add_edge('a','c',key='a2')
22 G.add_edge('a','c',key='a2',relacao="primo")
23
24 G.add_edge('c','d',key='a3')
```

```
25 G.add_edge('c','d',key='a3',relacao="amigos")
26
27 G.add_edge('d','e',key='a4')
28 G.add_edge('d','e',key='a4',relacao="amigos")
29
30 G.add_edge('d','f',key='a5')
31 G.add_edge('d','f',key='a5',relacao="Irmão")
32
33
34 print (list(G.edges(data=True, keys=True)))
35
36
37 nx.draw(G)
38
39 plt.savefig("relacoes.png")
40
41 plt.show()
42
```

# Exemplos



# Dificuldades encontradas

- Familiarização com a linguagem python
- Familiarização com as bibliotecas utilizadas
- Definição das regras da gramática
- Tipos de dados disponíveis para manipulação