

# Inteligência Artificial

## Comparação entre

### Busca sem Informação e

### Busca com Informação

Ícaro Gonçalves Siqueira

Engenharia de Computação - 2021/1



# Introdução

- Implementar dois algoritmos: Um de **busca sem informação** e um de **busca com informação**;
- **Três árvores de busca** com diferentes quantidades de nós;
- Medir tempo de execução;
- Analisar execução com a **criação de gráficos**.

# Metodología

- O algoritmo de busca sem informação escolhido foi o **busca em profundidade**;
- Na busca com informação, o algoritmo escolhido foi o **busca gulosa**;
- A linguagem de programação escolhida foi o **python 3**;
- As árvores de busca implementadas foram **árvores binárias, de 6, 14 e 30 nodos** e preenchidas com números em ordem crescente.

# Metodología

- Nas árvores de busca com informação foram implementadas também as distâncias de cada nó em relação com o nó de destino;
- Os pontos de origem e destino escolhidos foram 0 e 11, respectivamente, nas árvores de tamanhos 14 e 30 e, nas árvores de tamanho 6, os pontos foram 0 e 3.

# Metodología - Busca em Profundidade

```
12 import time
13
14 origem = '0'
15 destino = '12'
16 #destino = '3'
17
18 dist = 0
19 testado = set()
20
21 def buscaEmProfundidade(testado, graph, node, search):
22     global dist
23     if node not in testado:
24         print (node)
25         testado.add(node)
26         dist+=1
27
28         if node == search:
29             print('Encontrado!')
30             fim = time.time()
31             print('Distancia: %d' %dist)
32             print("Tempo: %fs" % (fim - inicio))
33         for vizinho in graph[node]:
34             buscaEmProfundidade(testado, graph, vizinho, search)
```

# Metodología - Busca em Profundidade

```
87  arvore3 = {
88      '0' : ['1', '2'],
89      '1' : ['3', '4'],
90      '2' : ['5', '6'],
91      '3' : [],
92      '4' : [],
93      '5' : [],
94      '6' : []
95  }
96
97  inicio = time.time()
98
99  buscaEmProfundidade(testado, arvore1, origem, destino)
```

# Metodología - Busca Gulosa

```
12 import time
13
14 origem = 0
15 destino = 12
16 #destino = 3
17
70
71 arvore3 = {
72     0 : [(1,1),(2,3)],
73     1 : [(3,0),(4,2)],
74     2 : [(5,4),(6,4)],
75     3 : [],
76     4 : [],
77     5 : [],
78     6 : []
79 }
80
```

# Metodologia - Busca Gulosa

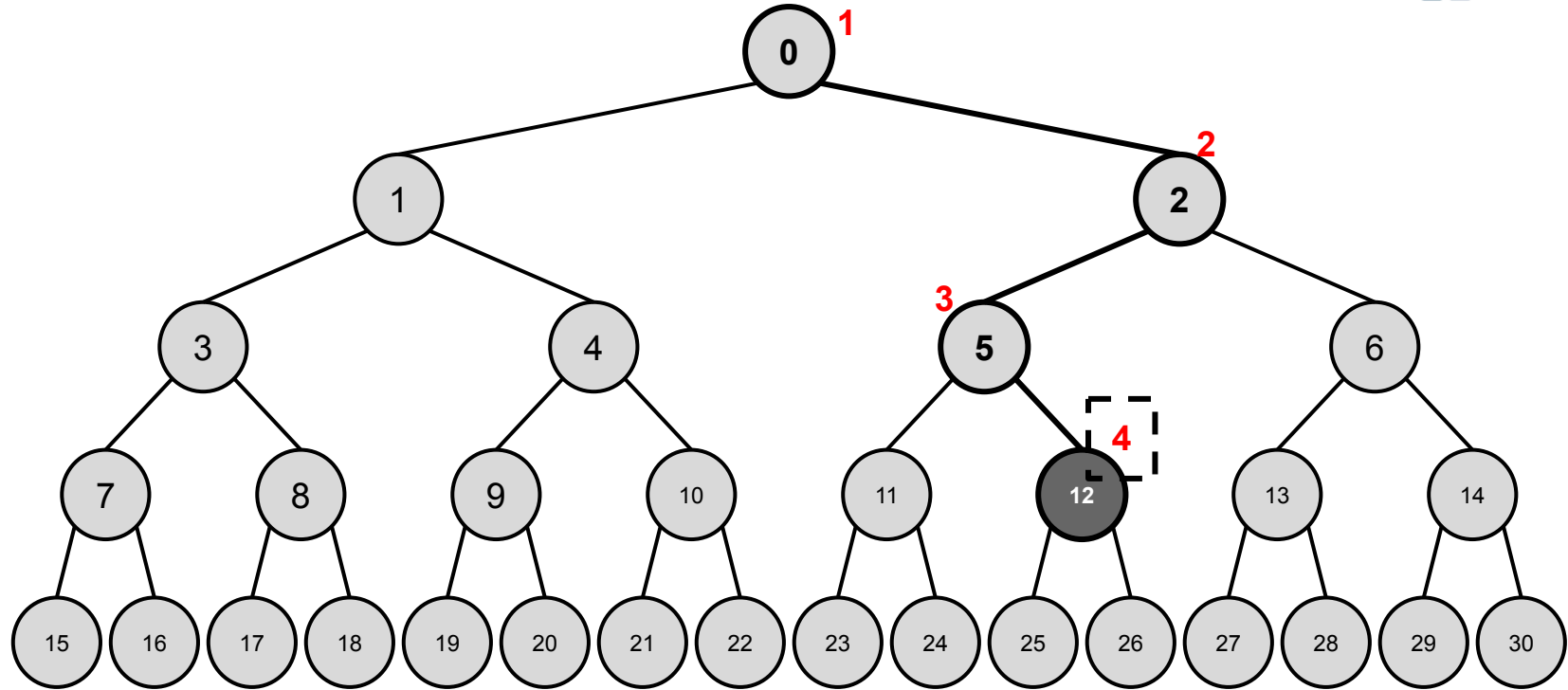
```
81 def buscaGulosa(graph, node, search):
82     nodoAtual = node
83     caminho = []
84     caminho.append(nodoAtual)
85     distancia = 0
86     while len(set([nodoVizinho for (nodoVizinho, distance) in graph.get(nodoAtual, [])]).
87               difference(set(caminho))) > 0:
88         vizinhoProximo = None
89         menorDistancia = None
90         for vizinho, vizinhoDistancia in graph[nodoAtual]:
91             if vizinho != nodoAtual and vizinho not in caminho:
92                 if menorDistancia is not None:
93                     if menorDistancia > vizinhoDistancia:
94                         menorDistancia = vizinhoDistancia
95                         vizinhoProximo = vizinho
```



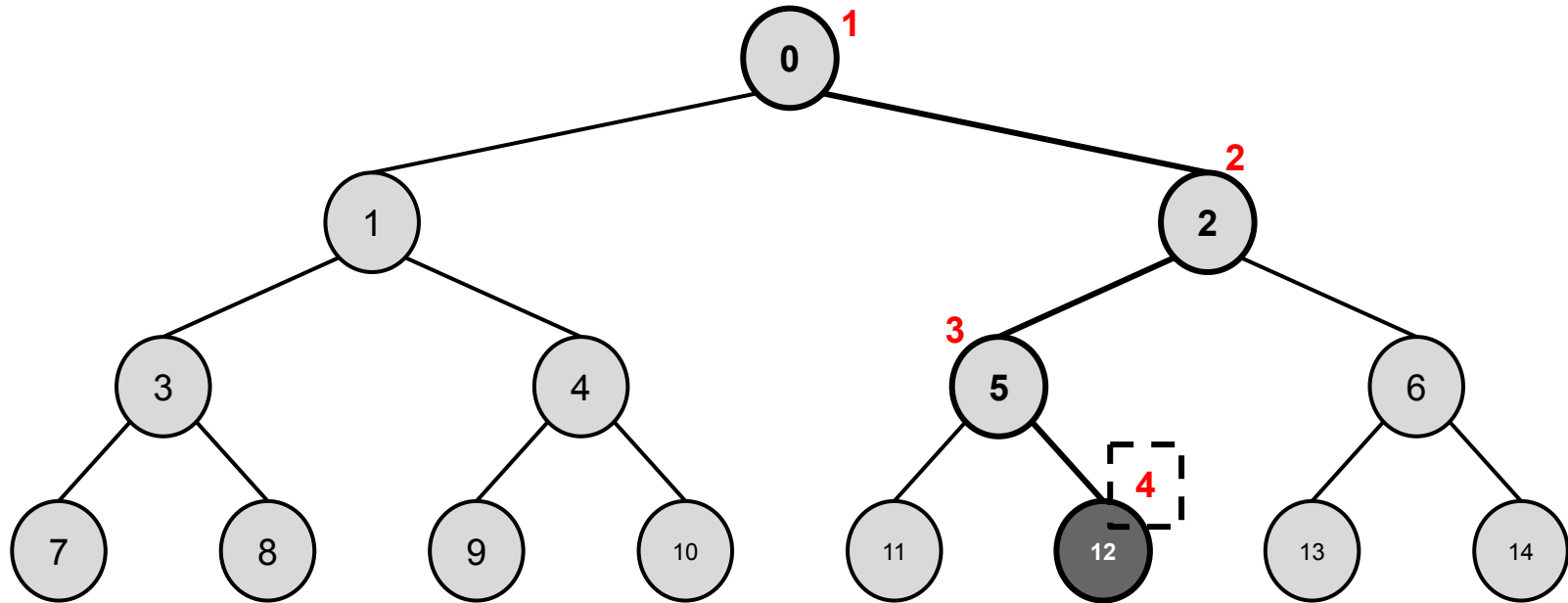
# Metodologia - Busca Gulosa

```
95         else:
96             menorDistancia = vizinhoDistancia
97             vizinhoProximo = vizinho
98         vizinhoMaisProximo = (vizinhoProximo, menorDistancia)
99         nodoAtual = vizinhoMaisProximo[0]
100        caminho.append(nodoAtual)
101        distancia += vizinhoMaisProximo[1]
102        if nodoAtual == search:
103            fim = time.time()
104            print("caminho: %s" %caminho)
105            print("distancia: %d" %distancia)
106            print("Tempo: %fs" %(fim - inicio))
107            return 0
108
109        inicio = time.time()
110
111        buscaGulosa(arvorel, origem, destino)
```

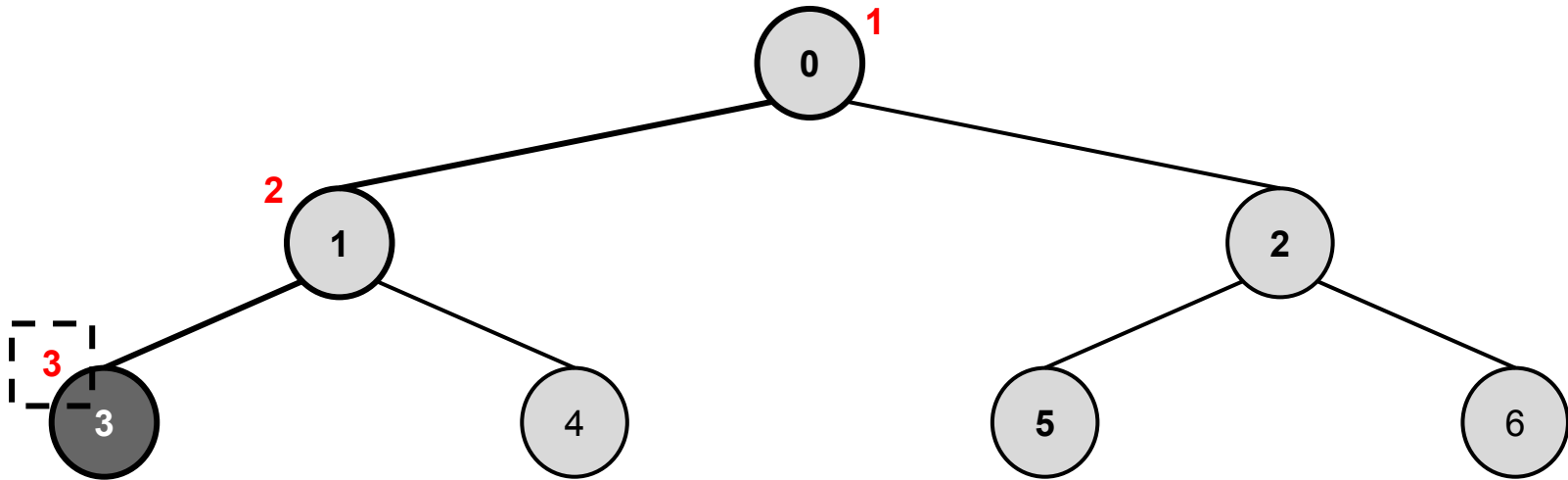
# Resultados - Busca Gulosa



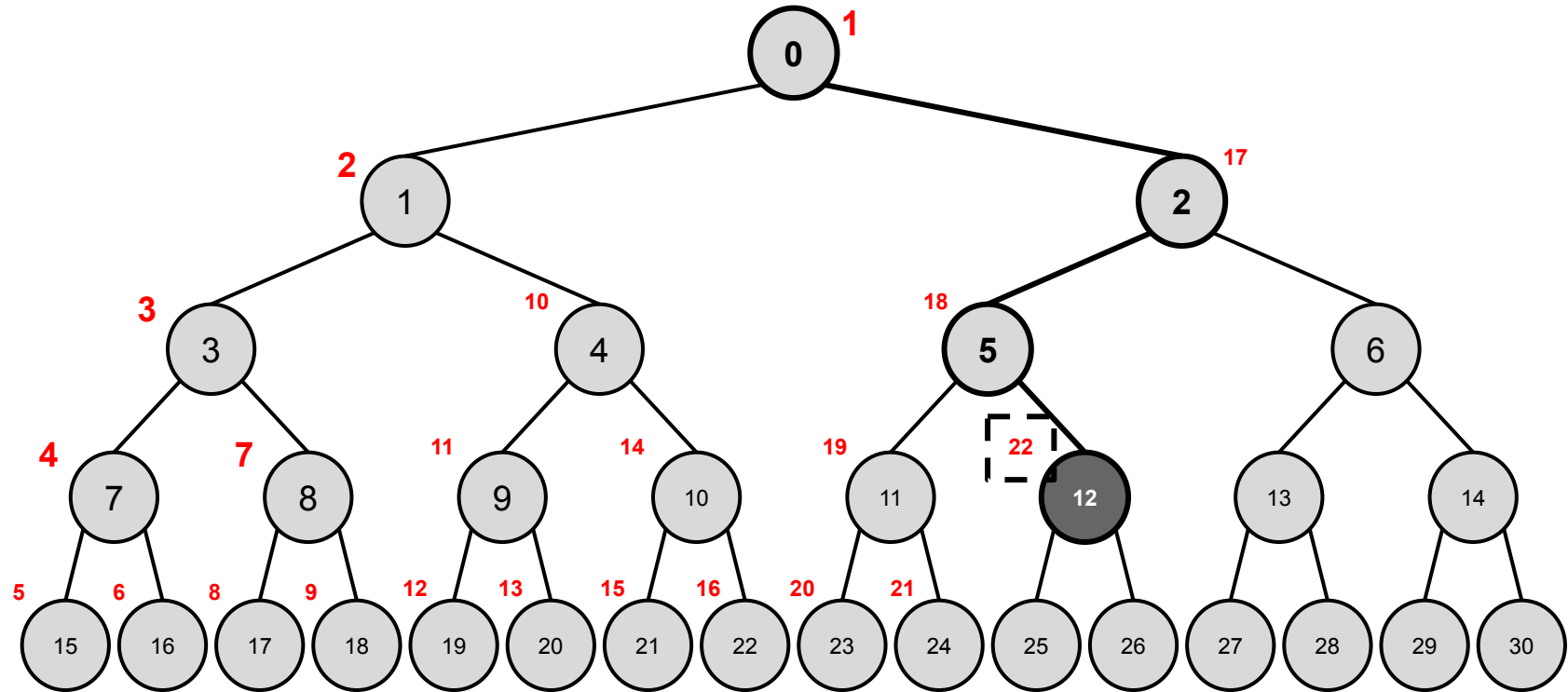
# Resultados - Busca Gulosa



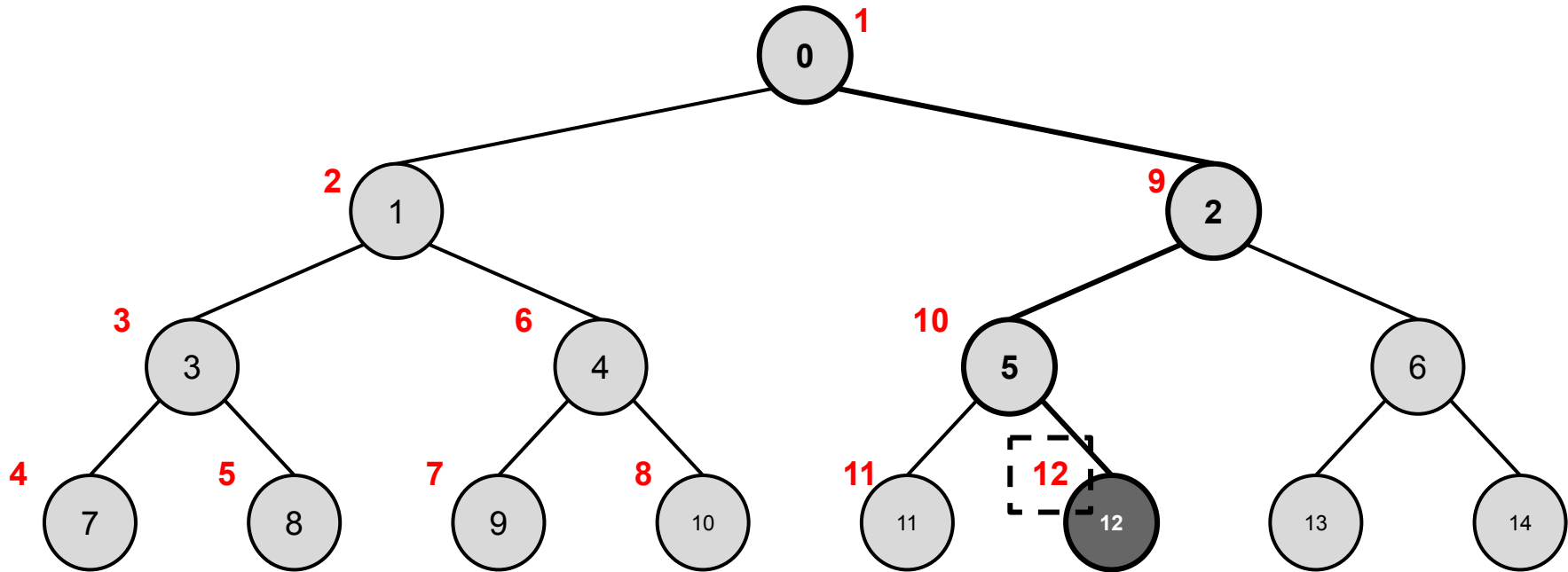
# Resultados - Busca Gulosa



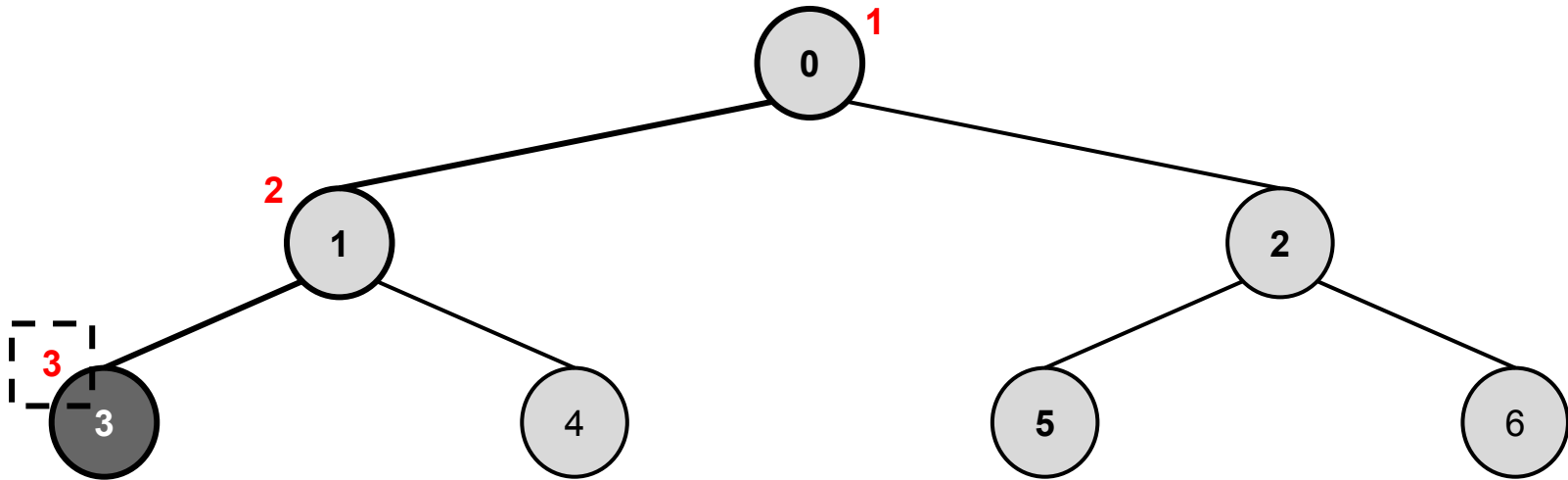
# Resultados - Busca em Profundidade



# Resultados - Busca em Profundidade



# Resultados - Busca em Profundidade



# Resultados - Busca Gulosa

Busca Gulosa			
Número de Nós	6	14	30
1	0.000006	0.000007	0.000008
2	0.000006	0.000007	0.000007
3	0.000006	0.000007	0.000008
Média(s)	0.000006	0.000007	0.000008



# Resultados - Busca em Profundidade

Busca em Profundidade			
Número de Nós	6	14	30
1	0.000015	0.000028	0.000046
2	0.000025	0.000046	0.000047
3	0.000034	0.000030	0.000055
Média(s)	0.000025	0.000035	0.000049

# Conclusão

- Independente do número de nós a busca com informação tem um melhor desempenho comparada à busca sem informação;
- Em ambas posições do nó objetivo, a busca com informação tem um melhor desempenho comparada à busca sem informação;
- O número de nós influencia muito mais na busca em profundidade.



Universidade Católica de Pelotas

Engenharia de Computação - 7º Semestre

Inteligencia Artificial

Professora: Alexandra Lackmann Zimpeck

Aluno: Ícaro Gonçalves Siqueira

05/05/2021 - São Lourenço do Sul