

Programa de amigo oculto

(Versão 1)

Luigi Spreafico

Sumário

1	Linguagem de programação e seus módulos	3
2	Explicação do programa	4
2.1	Fluxograma	4
2.2	Explicação mais detalhada	5
3	Algumas observações	7
4	References	8
5	O código	9

1 Linguagem de programação e seus módulos

A linguagem utilizada é o Python[1]. Os módulos utilizados são os seguintes:

```
1 import numpy as np
```

O módulo do `numpy` [2] (normalmente utilizado como `np`) será utilizado para podermos embraralhar os nomes dos participantes.

2 Explicação do programa

Aqui irei explicar o funcionamento do programa. Para tal, primeiro irei apresentar um fluxograma. Depois será feito uma explicação detalhada. De qualquer forma, no próprio código tem vários comentários explicando a lógica por trás do programa.

2.1 Fluxograma

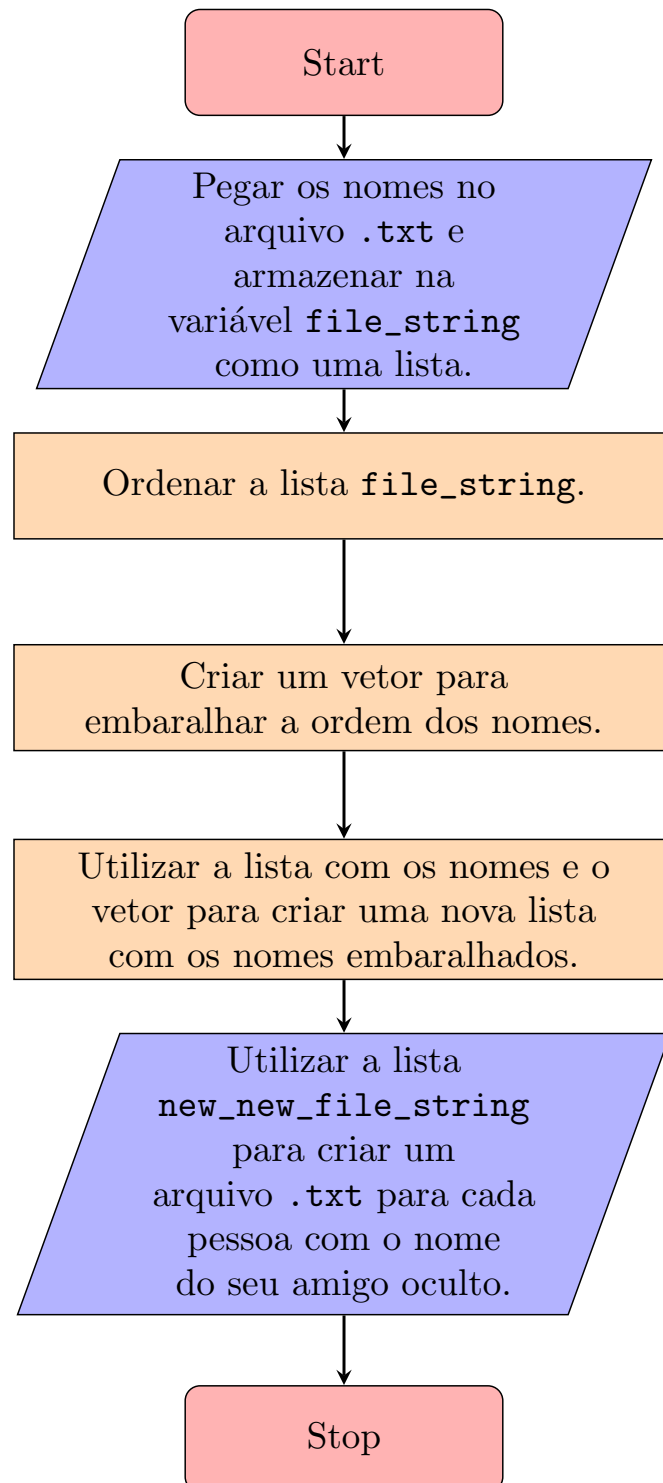


Figura 1: Fluxograma do programa.

2.2 Explicação mais detalhada

Aqui estarei explicando o programa. Antes, quero dizer que existe mais de uma forma de se fazer as coisas que eu fiz, e que essa é a versão 1. Além disso, que terá outras versões em que as coisas funcionarão de forma diferente.

No pedaço de código abaixo abrimos o arquivo `.txt` com os nomes das pessoas e armazenamos cada nome como um elemento de uma lista. Utilizamos dessa forma porque se tivermos algum problema enquanto o arquivo está aberto ele vai fechar. Tem outra forma de abrir um arquivo, mas que não fecha automaticamente, e com isso, dessa outra forma, acabamos tendo de lembrar de fechar o arquivo, além disso se tivermos algum problema enquanto o arquivo está aberto ele não vai fechar e podemos ter problemas quanto a isso.

```
with open('01_nome_das_pessoas_V1.txt') as f:
    file_string = f.readlines()
```

Listing 1: Pegando os nomes.

No pedaço de código abaixo pegamos a lista com os nomes e organizamos ela em ordem alfabética. Isso não é necessário. Só coloquei pra testar a função.

```
file_string.sort()
```

Listing 2: Ordenando os nomes.

No pedaço de código abaixo iremos criar um vetor que será utilizado para embaralhar a nossa lista de nomes.

```
idx = np.argsort(np.random.random(len(file_string)))
```

Listing 3: Criando um vetor para embaralhar.

O pedaço de código acima funciona da seguinte forma:

1. `len(file_string)`: pegamos o número de elementos da nossa lista
2. `np.random.random(len(file_string))`: criamos um vetor com o mesmo número de elementos que o número de participantes. Onde este vetor é feito com números aleatórios de 0 até 1 (não incluindo o 1). Dessa forma, cada vez que rodarmos o programa iremos obter um vetor com elementos diferentes.
3. `np.argsort(np.random.random(len(file_string)))`: Depois vamos e obtemos um vetor que nos diz a ordem dos índices de forma que esse nosso vetor fique em ordem correta.

Abaixo temos um exemplo onde mostramos o que acontece passo a passo.

```
>>> import numpy as np
>>> x = ['A', 'B', 'C', 'D', 'E']
>>> the_length = len(x)
>>> the_length
5
>>> the_random = np.random.random(the_length)
>>> the_random
array([0.39967192, 0.87469726, 0.14839865, 0.95127142, 0.44237115])
>>> idx = np.argsort(the_random)
>>> idx
array([2, 0, 4, 1, 3])
```

Listing 4: Exemplo criando um vetor para embaralhar no terminal do Python.

No pedaço de código abaixo criamos uma lista com o mesmo número de elementos que o número de participantes.

```
new_new_file_string = [0] * len(file_string)
```

Listing 5: Criando uma lista para os nomes embaralhados.

No pedaço de código abaixo estamos utilizando o vetor com índices embaralhados `idx` para colocar de forma embaralhada os nomes na lista nova.

```
for i1 in range(len(new_new_file_string)):
    new_new_file_string[i1] = file_string[idx[i1]]
```

Listing 6: Colocando os nomes na lista nova de forma embaralhada.

No pedaço de código abaixo pegamos e tiramos do final de cada nome o caracter \n. Isso é importante para podermos gerar o nome dos arquivos.

```
for i1 in range(len(new_new_file_string)):
    new_new_file_string[i1] = new_new_file_string[i1].replace('\n', '')
```

Listing 7: Retirando o caracter \n.

No pedaço de código abaixo vamos criar um arquivo .txt para cada pessoa, e depois colocar dentro dele quem essa pessoa tirou. A forma que vai ser criado o nome do .txt será para ser um amigo oculto de natal, para ser por exemplo de ano novo so mudar _natal_AAAA para _AnoNovo_AAAA. E substituir AAAA pelo ano do amigo oculto (ou data completa: DD_MM_AAAA).

```
for i1 in range(len(new_new_file_string)):
    file_name = 'amigo_oculto_' + new_new_file_string[i1] + '_natal_2022' + '.txt'
    if ( i1 == ( len(new_new_file_string) - 1 ) ):
        the_length = len(new_new_file_string[0])
        with open(file_name, 'w') as f:
            f.write('0 seu amigo oculto é: / A sua amiga oculta é:')
            f.write('\n')
            f.write('---> ')
            f.write(new_new_file_string[0])
            f.write('!'*(20-the_length))
            f.write('\n')
    else:
        with open(file_name, 'w') as f:
            the_length = len(new_new_file_string[i1+1])
            f.write('0 seu amigo oculto é: / A sua amiga oculta é:')
            f.write('\n')
            f.write('---> ')
            f.write(new_new_file_string[i1+1])
            f.write('!'*(20-the_length))
            f.write('\n')
```

Listing 8: Retirando o caracter \n.

Agora mostrando graficamente como é feita a escolha. Veja a figura abaixo (onde cada quadrado representa um elemento da lista final):

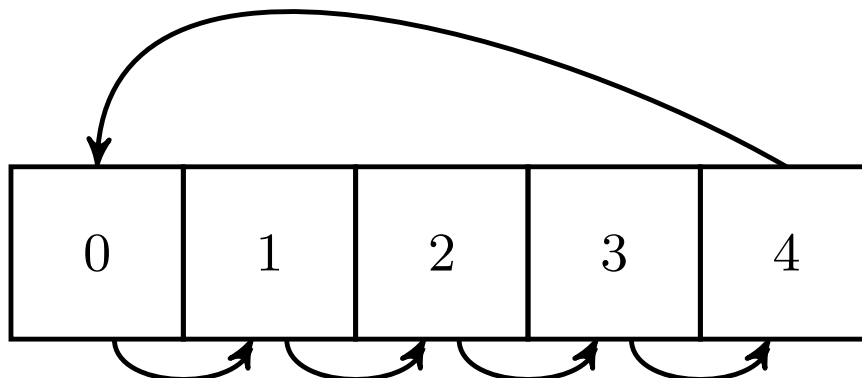


Figura 2: Fluxograma do programa.

Na figura acima, podemos ver o quadrados representando um elemento da lista final (da lista embaralhada), onde os números representam os índices da lista e começamos pelo 0 por estarmos utilizando Python. As setas mostram quem vai ter quem como amigo oculto, por exemplo, o nome que estiver no índice 1 terá a pessoa com nome no índice 2 como seu amigo oculto.

3 Algumas observações

Nesta versão é gerado um arquivo .txt para cada pessoa onde dentro tem o nome do seu amigo oculto ou da sua amiga oculta. Teria de mandar para cada um o seu arquivo (por e-mail, whatsapp, ...).

Exemplo de mensagem.
Oi! Tudo tranquilo? No arquivo tem o nome do seu amigo oculto ou da sua amiga oculta, só abrir para ver.

Nas próximas versões o objetivo é de por exemplo:

- Colocar algo para podermos inserir os nomes das pessoas, seja pelo terminal, ou por uma interface gráfica.
- Colocar para enviar automaticamente para o e-mail das pessoas os resultados.
- Quando for rodar o programa (no começo, antes de tudo) apagar todos os arquivos que não sejam os necessários (por exemplo, se você for rodar uma segunda vez com menos pessoas, nessa versão ele já vai ter gerado algum arquivo .txt referente a uma pessoa que não vai participar e você teria de apagar manualmente).

Além disso, com esse código temos que só terá um círculo, por exemplo, tendo 5 pessoas teríamos

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1$$

e nunca

$$1 \rightarrow 2 \rightarrow 1$$

e

$$3 \rightarrow 5 \rightarrow 4 \rightarrow 3$$

, e, ninguém tira a si mesmo.

Outra observação a ser feita é que as exclamações no final de cada nome é para que todos os arquivos .txt tenham o mesmo número de caracteres, de forma que todos os arquivos tenham o mesmo tamanho. Isso porque como cada nome tem um número diferente de letras, acabariam tendo um arquivo com tamanho diferente. Dessa forma, eu poderia acabar olhando sem querer que um arquivo é maior ou menor que o outro (mesmo sem abrir o arquivo) e com isso ter uma dica de quem a pessoa tirou. Dessa forma, eu não terei nenhuma dica sobre quem alguém tirou sem querer, e tudo pode ficar bem protegido. A única forma de eu saber algo seria abrindo os arquivos gerados.

E por fim, no arquivo 01_nome_das_pessoas_V1.txt colocar cada nome em um linha (como no exemplo).

4 References

- [1] Python programming language description. <https://www.python.org>. Accessed: 2022-12-01.
- [2] NumPy python module description. <https://numpy.org/>. Accessed: 2022-12-01.

5 O código

```
1
2 # ===== #
3
4 # VERSAO 1 (A VERSAO MAIS SIMPLES)
5
6 # Depois colocar de forma mais completa, por exemplo:
7 # . Colocar algo para podermos inserir os nomes das pessoas, seja pelo terminal,
8 # ou por uma interface grafica.
9 # . Colocar para enviar para o e-mail das pessoas por exemplo os resultados.
10 # . Quando for rodar o programa (no comeco, antes de tudo)
11 # apagar todos os arquivos que nao sejam os necessarios.
12 # . ...
13
14 # OBS.: com esse codigo temos que so tera um circulo
15 # (por exemplo, tendo 5 pessoas teriamos 1->2->3->4->5->1 e nunca 1->2->1 e 3->5->4->3),
16 # e, ninguem tira a si mesmo.
17
18 # Luigi Spreafico
19
20 # ===== #
21
22 # Comandos que eu uso.
23
24 # cd Documents/GitHub/Amigo_oculto/Versao_1
25
26 # python3.9 01_amigo_oculto_V1.py
27
28 # ===== #
29
30 # Modulos que eu uso.
31
32 import numpy as np
33
34 # ===== #
35
36 # O programa em si.
37
38 # Abrindo o arquivo com os nomes das pessoas
39 # e armazenando cada nome como um elemento de uma lista.
40 with open('01_nome_das_pessoas_V1.txt') as f:
41     file_string = f.readlines()
42
43 # A proxima linha e para checar se o codigo esta funcionando direitinho,
44 # deixar ela comentada quando for usar para nao saber quem tirou quem.
45 # print(file_string)
46
47 # Colocando em ordem alfabetica a lista dos nomes.
48 # (OBS.: Isso nao era preciso.)
49 file_string.sort()
50
51 # A proxima linha e para checar se o codigo esta funcionando direitinho,
52 # deixar ela comentada quando for usar para nao saber quem tirou quem.
53 # print(file_string)
54
```

```

55 # Aqui criamos um vetor com o mesmo numero de elementos que o numero de participantes.
56 # Onde este vetor e feito com numeros aleatorios de 0 ate 1 (nao incluindo o 1).
57 # Dessa forma, cada vez que rodarmos o programa
58 # iremos obter um vetor com elementos diferentes.
59 # Depois obtemos um vetor que nos diz a ordem dos indices,
60 # de forma que esse nosso vetor fique em ordem correta.
61 idx = np.argsort(np.random.random(len(file_string)))
62
63 # Criamos uma lista com o mesmo numero de elementos que o numero de participantes.
64 new_new_file_string = [0] * len(file_string)
65
66 # Utilizamos a lista original com os nomes
67 # e o vetor com os indices para embaralhar a ordem dos nomes.
68 for i1 in range(len(new_new_file_string)):
69     new_new_file_string[i1] = file_string[idx[i1]]
70
71 # A proxima linha e para checar se o codigo esta funcionando direitinho,
72 # deixar ela comentada quando for usar para nao saber quem tirou quem.
73 # print(new_new_file_string)
74
75 # Agora pegamos e tiramos do final de cada nome o caracter \n.
76 # Isso e importante para podermos gerar o nome dos arquivos.
77 for i1 in range(len(new_new_file_string)):
78     new_new_file_string[i1] = new_new_file_string[i1].replace('\n', '')
79
80 # A proxima linha e para checar se o codigo esta funcionando direitinho,
81 # deixar ela comentada quando for usar para nao saber quem tirou quem.
82 # print(new_new_file_string)
83
84 # Aqui criar um arquivo .txt para cada pessoa,
85 # e depois colocar dentro dele quem essa pessoa tirou.
86 # Obs.: a forma que vai ser criado o nome do .txt sera para ser um amigo oculto de natal,
87 # para ser por exemplo de ano novo so mudar "_natal_AAAA" para "_AnoNovo_AAAA".
88 # E substituir AAAA pelo ano do amigo oculto (ou data completa: DD_MM_AAAA).
89 for i1 in range(len(new_new_file_string)):
90     file_name = 'amigo_oculto_' + new_new_file_string[i1] + '_natal_2022' + '.txt'
91     if ( i1 == ( len(new_new_file_string) - 1 ) ):
92         the_length = len(new_new_file_string[0])
93         with open(file_name, 'w') as f:
94             f.write('0 seu amigo oculto e: / A sua amiga oculta e:')
95             f.write('\n')
96             f.write('---> ')
97             f.write(new_new_file_string[0])
98             f.write('!'*(20-the_length))
99             f.write('\n')
100     else:
101         with open(file_name, 'w') as f:
102             the_length = len(new_new_file_string[i1+1])
103             f.write('0 seu amigo oculto e: / A sua amiga oculta e:')
104             f.write('\n')
105             f.write('---> ')
106             f.write(new_new_file_string[i1+1])
107             f.write('!'*(20-the_length))
108             f.write('\n')
109
110 # ===== #

```