

UNIVERSIDADE FEDERAL FLUMINENSE

GABRIEL FERREIRA ALVES  
ÍCARO GOULART FARIA MOTTA FRANÇA

**RELATÓRIO DO SEGUNDO TRABALHO DE  
IMPLEMENTAÇÃO:**

NITERÓI

2017

---

# Sumário

<b>1</b>	<b>Introdução . . . . .</b>	<b>2</b>
<b>2</b>	<b>Desenvolvimento e Requisitos . . . . .</b>	<b>3</b>
2.1	Exemplo de Entrada . . . . .	3
2.2	Saída . . . . .	3
<b>3</b>	<b>Algoritmo . . . . .</b>	<b>4</b>
3.1	main.py . . . . .	4

# 1 Introdução

Desenvolva um programa que lide com três operações básicas que devem ser executadas  $N$  vezes. O valor de  $N$  é dado como entrada do programa. As operações básicas consistem em imprimir as cores vermelho, azul e verde, exatamente nesta ordem,  $N$  vezes. A impressão de cada cor deve ser feita por uma Thread diferente. Cada uma das  $N$  vezes que for impressa a sequência de cores, ela deve ser completa e ininterrupta. Ou seja, você deve utilizar um semáforo para garantir a ordem correta da impressão (vermelho, azul e verde) para cada sequência e outro semáforo para garantir que a sequência  $n$  só se inicie após a conclusão da sequência  $n-1$  (onde  $n=1\dots N$ ). Por fim, cada Thread deve sortear um número inteiro aleatório  $i$  (que varia entre 0-9) e que será utilizado para adormecer essa Thread por  $i$  segundos, após ganhar o direito de execução do semáforo, mas antes de imprimir a cor, cada uma das  $N$  vezes.

## 2 Desenvolvimento e Requisitos

Para a implementação do semáforo de Threads foi utilizada a linguagem Python na versão 3 em um computador Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz, 4.00 GB de Memória RAM com Sistema Operacional Windows 10 x64. A IDE utilizada para codificação foi a *PyCharm Community Edition* 2017.2.3.

Todo o código está presente em uma classe "main.py", responsável por executar o algoritmo. Para performar a execução do código, é necessário informar o parâmetro de entrada: o número de vezes que a sequência de cores ("vermelho, azul, verde", nessa ordem) deverá ser exibida. Por meio da IDE *PyCharm*, esta ação pode ser executada clicando no menu *Run* e, em seguida, *Edit Configurations...* Na janela *Run/Debug Configurations*, o parâmetro deve ser informado no campo *Script parameters*. Após preencher o campo, deve-se executar a classe "main.py", que irá exibir cor por cor. É importante ressaltar que a thread que executa a impressão da cor dorme por tempo randomicamente sorteado, de 0 a 9 segundos.

Caso a execução seja realizada por meio do terminal do computador, é necessário informar "python main.py valor\_de\_n\_execuções".

### 2.1 Exemplo de Entrada

No exemplo a seguir, foi selecionado o número 3 como o valor de n. Isso significa que a sequência "vermelho, azul, verde" deverá ser exibida três vezes.

### 2.2 Saída

vermelho

azul

verde

vermelho

azul

verde

vermelho

azul

verde

## 3 Algoritmo

### 3.1 main.py

```
1 import sys
2 import threading
3 import time
4 import logging
5 from random import randint
6
7
8 count = 0
9
10
11 params = sys.argv[1:]
12
13
14 if len(params) == 1:
15     n = int(params[0])
16 else:
17     print("erro. Eh necessario informar o numero de vezes que
18           as cores deverao ser impressas.")
19     exit()
20
21 def run(cor, id):
22     global count
23     for i in range(n):
24         while(count != id):
25             if count == id :
26                 break
27
28         seconds = randint(0, 9)
29         time.sleep(seconds)
30         print(cor)
31
32         count = count + 1
33
34         if (id == 3):
35             count = 1
```

```
36
37
38 thread_vm = threading.Thread(group=None, target=run, args=("
    vermelho", 1))
39 thread_a = threading.Thread(group=None, target=run, args=("azul",
    2))
40 thread_vd = threading.Thread(group=None, target=run, args=("verde
    ", 3))
41
42
43 count = 1
44
45
46 thread_vm.start()
47 thread_a.start()
48 thread_vd.start()
49
50 thread_vm.join()
51 thread_a.join()
52 thread_vd.join()
```