

Para simular o funcionamento, eu transformei os valores absolutos dos minutos para segundo seguindo a regra:

60 minutos → 6 segundos

Dessa forma, surgiu um cliente a cada 3 segundos e o barbeiro trabalha entre 2 e 3 segundos.

O algoritmo foi escrito em Python3 usando a biblioteca *threading* para executar em múltiplas threads e para controle da fila de clientes, foi usada a biblioteca *queue* que pode ser acessada por vários threads simultaneamente e já possui um lock interno.

O barbeiro fica de olho na fila e, no momento que um cliente entra na fila, se o barbeiro estiver dormindo, ele acorda e já atende o cliente.

O algoritmo usado para solucionar o problema do barbeiro dorminhoco pode ser encontrado no link:

<https://github.com/huine/sistemas-distribuidos/blob/master/aula6/barbeiro.py>

```
gabriel@DESKTOP-L4097KH:/mnt/c/Users/gabri/Desktop/sistemas-distribuidos/aula6$ python barbeiro.py
0 indo ao barbeiro
Atendendo o cliente: 0
1 indo ao barbeiro
Atendendo o cliente: 1
2 indo ao barbeiro
Atendendo o cliente: 2
3 indo ao barbeiro
Atendendo o cliente: 3
4 indo ao barbeiro
Atendendo o cliente: 4
5 indo ao barbeiro
Atendendo o cliente: 5
6 indo ao barbeiro
Atendendo o cliente: 6
7 indo ao barbeiro
Atendendo o cliente: 7
8 indo ao barbeiro
Atendendo o cliente: 8
9 indo ao barbeiro
Atendendo o cliente: 9
clientes_atendidos: 10
clientes_recusados: 0
barbeiro dormindo: 5.13239 s
Terminated
```

Figura 1 Output do programa