

## Exercício Avaliativo 1 – Simula Sensores

Na programação existe um conceito muito importante que permite um processo dividir a si mesmo em duas ou mais tarefas para poderem ser executadas concorrentemente. Este conceito é chamado de **THREADS**.

Abaixo segue um exemplo de como abrir uma Thread em Python:

```
import threading #Lib para se trabalhar com Threads
import time #Permite executar uma Thread mais devagar

# Funcao que sera chamada pela Thread
def sayhello(nome, intervalo):
    while True:
        print(nome, 'say Hello :)')
        time.sleep(intervalo)

# Abrindo uma Thread
x = threading.Thread(target=sayhello, args=('Thread 1', 2))
x.start()
```

Utilizando deste conceito e dos seus conhecimentos em Bancos de Dados (BDs) Orientados a Documentos, pede-se:

- Crie três Threads, **cada uma responsável por simular a temperatura de um sensor** em uma rede de Internet das Coisas (IoT);
- Cada sensor deverá, de tempos em tempos (tempo pode ser configurado por você, programador), gerar uma temperatura aleatória, que deve variar entre 30 e 40 C°;
- Estes valores de sensores, ao serem gerados, além de serem apresentados no terminal do Python, também deverão ser guardados no BD MongoDB (mais particularmente dentro de um **database** chamado **bancoiot**, que possui uma **collection** chamada **sensores**);
- Cada sensor terá seu próprio **documento** na collection sensores (que podem ser criados manualmente de forma antecipada), com o objetivo de não só guardar a última leitura de cada um deles, mas também informar se alguns deles estão ou não alarmados. Cada documento deve possuir os campos: **\_id** (gerado automaticamente), **nomeSensor** (Ex: Temp1, Temp2, etc..), **valorSensor** (que é gerado aleatoriamente no Python), **unidadeMedida** (que é C° para todos) e **sensorAlarmado** (setado inicialmente com o valor **false**);
- O campo **sensorAlarmado** deverá ter seu valor alterado no documento de um sensor para **true** quando a temperatura gerada pelo respectivo sensor for > 38 C°;
- A partir do momento que um sensor tenha seu campo alarmado setado em **true**, dali em diante ele não deverá mais gerar valores aleatórios de temperatura (e consequentemente não precisará mais atualizar seu documento no BD), mas sim mostrar a seguinte mensagem no terminal do Python: **“Atenção! Temperatura muito alta! Verificar Sensor X!”**, em que X é o nome do sensor alarmado.

Bom trabalho a todos!