

07

Publicando temperatura

Transcrição

Continuando com nossas edições no código, agora vamos alterar a mensagem que está sendo publicada no tópico para de fato publicar a temperatura da CPU do Raspberry PI. Na sessão `# Publish to the same topic in a loop forever` temos o seguinte código:

```
loopCount = 0
while True:
    myAWSIoTMQTTClient.publish(topic, "New Message" + str(loopCount), 1)
    loopCount += 1
    time.sleep(1)
```

Nós não precisamos mais desse contador, por que estaremos alterando a mensagem e ele atualmente é usado apenas para compor a mensagem. Ao invés disso, criaremos uma variável que fará a captura do valor da temperatura e dentro do método `publish` adicionaremos esta variável, acessando o valor da temperatura capturada da seguinte forma:

```
while True:
    cpu=CPUtemperature()
    myAWSIoTMQTTClient.publish(topic, cpu.temperature, 1)
    time.sleep(2)
```

Note que mudamos o valor de `time.sleep` de 1 para 2 para aumentar o intervalo entre as mensagens. Dessa forma, a cada 2 segundos, capturamos o valor da temperatura da CPU e enviamos para o tópico. Salvamos e executamos novamente o script `start.sh` e tudo procede como esperado.

Para testes mais interessantes de visualizar do que a tela do terminal imprimindo textos a todo momento, usamos a opção `Test` do menu `AWS IoT` na plataforma assinando mais uma vez ao tópico como vimos anteriormente, porém, agora o tópico é o `telemetria/temperatura`. Ao assinar, teremos as temperaturas sendo impressas na página.

telemetria/temperatura	Oct 16, 2017 4:01:22 PM -0200	Export	Hide
43.312			
telemetria/temperatura	Oct 16, 2017 4:01:20 PM -0200	Export	Hide
44.388			

Apesar de funcionar normalmente, faremos mais um ajuste para enviarmos a mensagem no formato **JSON**. Temos dois bons motivos para isso. O primeiro deles é que se trata de uma recomendação da própria Amazon. O segundo é que ficará mais fácil para construir regras de análise e até mesmo para armazenamento no banco de dados.

Outro dado que vamos inserir nas mensagens é o *timestamp*. Esse é um dado de data e hora, ou seja, anexaremos junto ao dado da temperatura do dispositivo, o dado de data/hora e deixaremos tudo formatado em JSON.

