

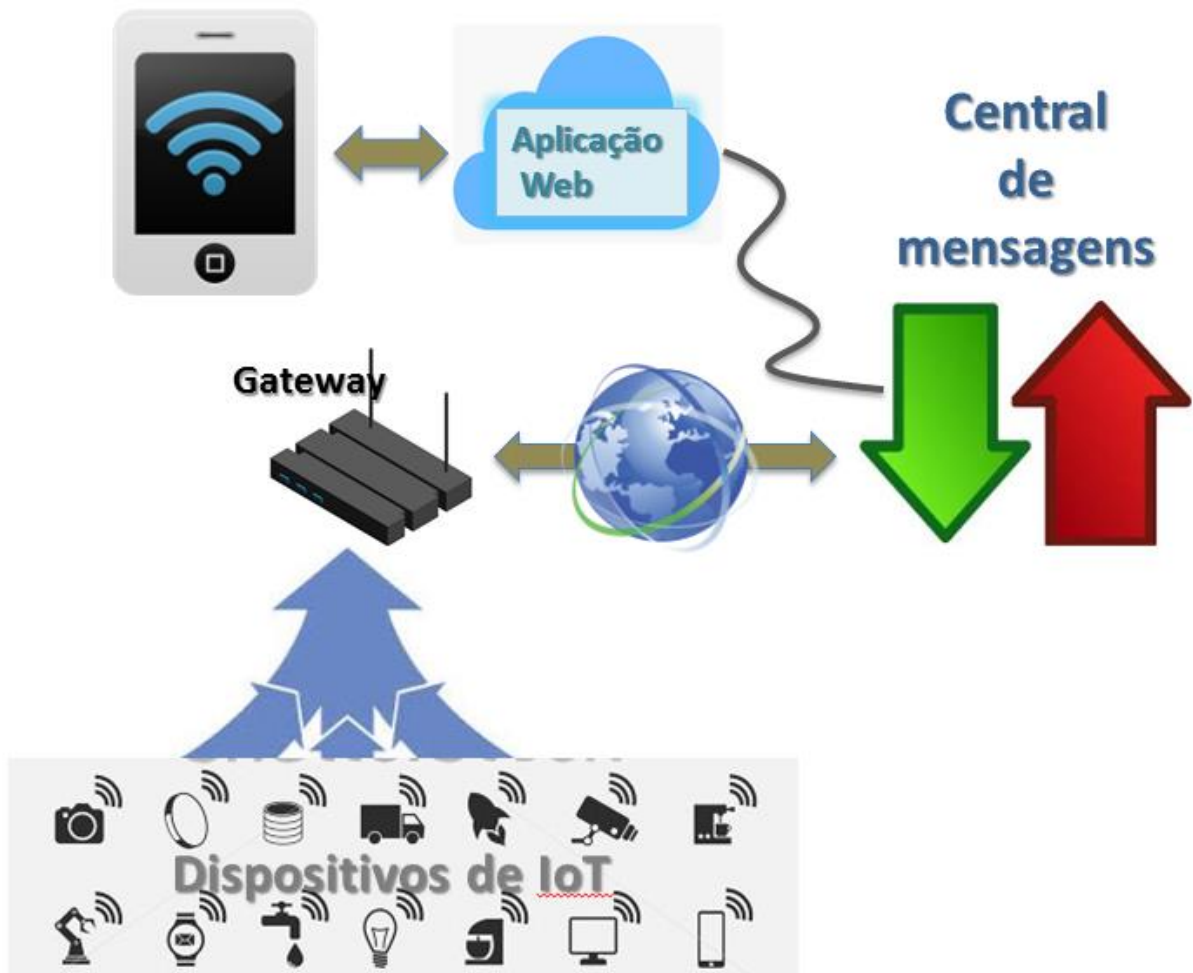
**Disciplina:** Arquiteturas Disruptivas, IoT, IA e Big Data

**Curso:** Tecnologia em Análise e Desenvolvimento de Sistemas

**Professor Antonio Selvatici**

## NAC 10

### Trabalho sobre arquitetura de Internet das Coisas



**Objetivo:** Construção de uma solução de IoT que abrange todas as pontas da comunicação entre usuário e dispositivos

**Forma de execução:** grupo do AM (até 5 alunos)

**Entregável:** 1 (um) arquivo .ZIP contendo:

- O nome dos integrantes em arquivo .TXT
- A programação do Arduino (arquivo .INO)
- A programação do Node-RED (arquivo .JSON)
- A aplicação web desenvolvida (pasta do projeto Java)
- O aplicativo mobile desenvolvido (pasta do projeto)

A NAC consiste no desenvolvimento de uma solução simples mas completa que ilustra as partes mais importantes de uma arquitetura de IoT: os dispositivos e os usuários. A solução também não contempla as ferramentas de segurança que seriam necessárias ao implantar um sistema real.

A solução pedida requer a implementação de:

1 – Programa em linguagem do Arduino que realize (2 pontos):

- A) a leitura de pelo menos dois sensores a cada 200ms, disponibilizando as leituras na porta serial, no formato JSON { "<SENS1>":<VALOR1>, "<SENS2>":<VALOR2> }, onde <SENS1> e <SENS2> são os nomes dos sensores, e <VALOR1> e <VALOR2> são os valores lidos, respectivamente.
- B) a execução de comandos em pelo menos dois atuadores, lendo esses comandos pela porta serial. Os comandos devem ser na forma { "<CMD>":<VALOR> }, onde CMD é o nome do comando e VALOR é o parâmetro enviado. Por exemplo, se <CMD> for led, então <VALOR> pode ser "On" ou 255, por exemplo.

O programa do Arduino pode ser executado no próprio Arduino UNO (ou outro modelo) ou mesmo numa placa ESP-8266

2 – Comunicação com o broker MQTT de sua escolha: de ser desenvolvido um programa Node-RED ou Arduino que conecte o dispositivo de IoT à central de mensagens (MQTT Broker) através dos seguintes tópicos (3 pontos):

- A) Realize PUB sempre que for realizada a leitura dos sensores, convertendo e enviando a informação de cada sensor no formato JSON { "value": <VALOR> } ao tópico:

fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/sensor/<SENS>

onde:

- i. <TT> corresponde à turma (4sia, 4sir ou 4sis)
- ii. <NNN> corresponde ao nome do grupo
- iii. <TYPE> corresponde ao tipo de dispositivo (arduino ou esp8266)

- iv. <ID> corresponde à identificação do dispositivo (pode ser o MAC Address do ESP8266, ou o número de série do Arduino, ou qualquer identificação que quiser)
- v. <SENS> corresponde ao nome do sensor

B) Para cada sensor, faça a subscrição (SUB) ao seguinte tópico, que irá receber um JSON da forma {"value": <VALOR>} :

fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/cmd/<CMD>

onde <CMD> corresponde ao nome do comando a ser executado. Ao receber o comando, ele deve ser convertido e enviado à placa para ser executado.

Este item pode ser realizado de duas formas:

- A) No caso da placa de IoT ser um Arduino, o item 2 deve ser executado na forma de um Gateway programado em Node-RED, que se conecta ao Arduino através da porta serial.
- B) No caso da placa de IoT ser um ESP8266 Node-MCU, este item deve constar da programação da placa (em linguagem Arduino), e não é necessário um gateway adicional.

3 – Aplicação Web (desenvolvida em Java ou Node-RED) que se conecte ao Broker MQTT e contenha uma API ReSTful com os seguintes endpoints (4 pontos):

A) Recupera uma lista de IDs de dispositivos do tipo especificado, no formato JSON

[GET]http://<HOST>:<PORTA>/<PROJNAME>/fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/all

B) Recupera a lista dos sensores disponíveis para o dispositivo, no formato JSON

[GET]http://<HOST>:<PORTA>/<PROJNAME>/fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/sensor/all

C) Recupera o último valor lido no sensor, no formato JSON {"value": <VALOR>}

[GET]http://<HOST>:<PORTA>/<PROJNAME>/fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/sensor/<SENS>

D) Recupera a lista dos comandos disponíveis para o dispositivo, no formato JSON

[GET]http://<HOST>:<PORTA>/<PROJNAME>/fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/cmd/all

E) Executa um comando, passando um argumento da forma {"value": <VALOR>}

[POST]http://<HOST>:<PORTA>/<PROJNAME>/fiap/iot/turma/<TT>/grupo/<NNN>/devtype/<TYPE>/devid/<ID>/cmd/<CMD>

4 – Aplicativo móvel, desenvolvido em qualquer linguagem de programação, cuja interface gráfica demonstre o funcionamento de todas os endpoints do exercício anterior, ou seja (1 ponto):

- A) Apresente um menu que permita ver todos os dispositivos disponíveis e, para cada dispositivo, quais são os sensores e comandos disponíveis
- B) Permita enviar comandos que possam ser executados pelo dispositivo de IoT

A execução e entrega da NAC será ocupará duas aulas presenciais:

- Uma aula para a programação do Arduino ou do ESP8266 e saneamento de dúvidas do trabalho
- Outra aula para a apresentação do trabalho funcionando

Os itens 2 e 3 do trabalho correspondem ao tópicos que serão cobrados na parte prática da Prova Semestral, que será individual.

Datas:

- Plantão tira dúvidas e programação do dispositivo: aula da semana com início em 6/5/2019
- Entrega e avaliação: aula da semana com início em 20/5/2019