



Projeto 03

Coisas Monitoradas – Prática

Jan K. S. – janks@puc-rio.br

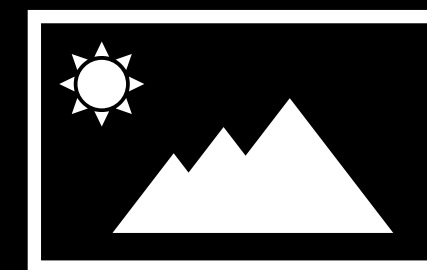
ENG4051 – Projeto Internet das Coisas

Testes Iniciais

1

No ESP32, crie um servidor com uma rota que carregue e retorne uma imagem "foto.jpeg" dentro do SD Card.

192.168.X.X/foto



2

Inscreva-se no tópico "foto/SEU_NOME".

Ao receber uma mensagem nesse tópico, tire uma foto e salve o arquivo no SD Card como foto.jpeg.

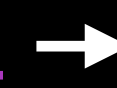


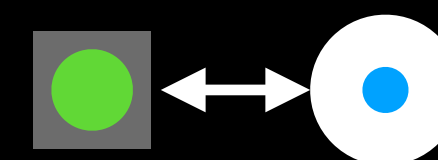
foto.jpeg

3

Configure o scanner Bluetooth para varrer iBeacons.

Ao encontrar o iBeacon da aula, calcule sua distância.

Se for menor que 2 metros, mostre a cor verde no LED RGB.
Se for maior, mostre a cor vermelha.

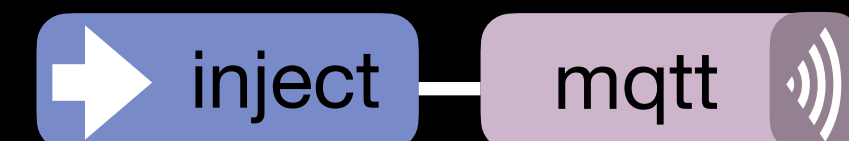


4

No Node-RED, use um bloco Inject para publicar um tópico "telegram" com o seu nome na mensagem.

Ainda no Node-RED: ao receber esse tópico do MQTT, envie o conteúdo para o seu Telegram.

Clique em Implementar e teste clicando no Inject.



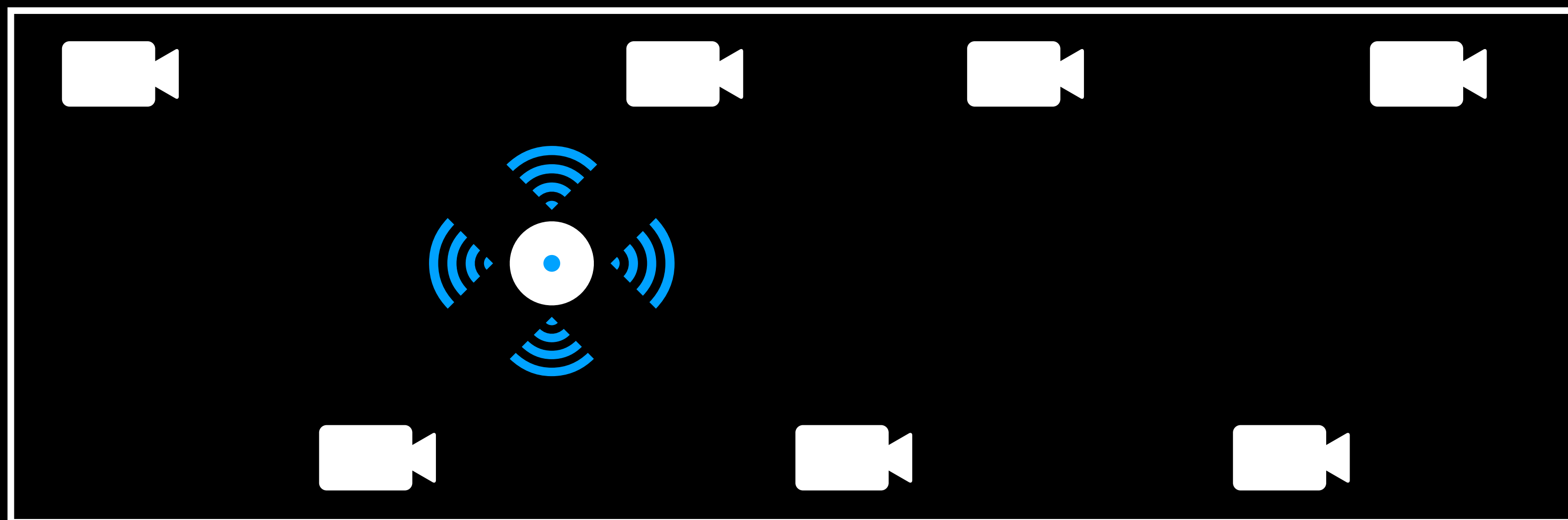
5

Use um bloco Delay configurado com a Ação "Limite de Taxa" para limitar o envio de mensagens para 1 a cada 10 segundos.



Testes Iniciais

Protótipo



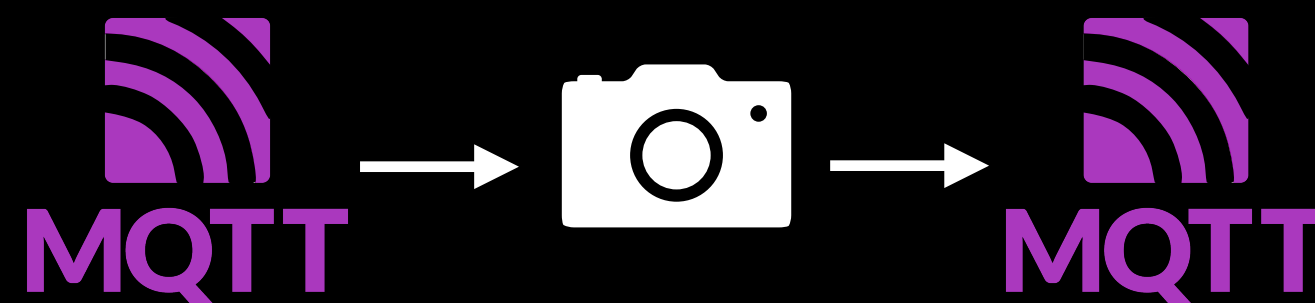
Monitoramento do Beacon e das Câmeras

1

No ESP32, inscreva-se no tópico "camera/NUMERO_DA_SUA_BANCADA".

Ao receber uma mensagem, tire uma foto e envie-a para o tópico "foto/NUMERO_DA_SUA_BANCADA".

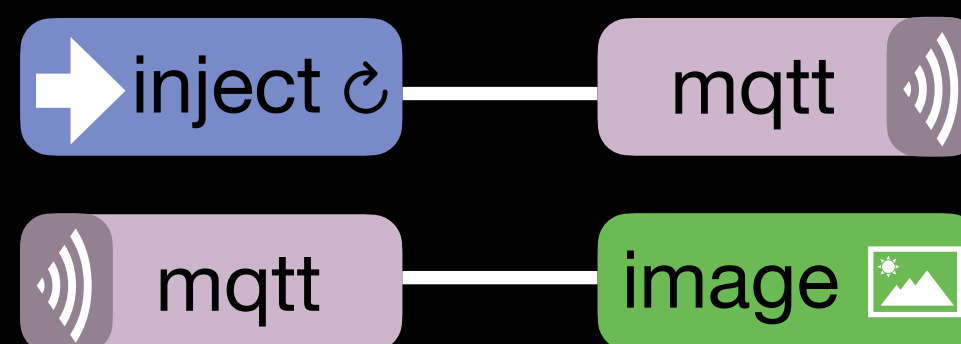
Teste no MQTTX.



2

No Node-RED, use um bloco Inject configurado para repetir a cada 10 segundos que envia uma mensagem para o tópico da câmera.

Ao receber uma mensagem do tópico da foto, exiba-a num nó Imagem.



Protótipo: Monitoramento da Câmera

3

Calcule a média de cada 5 distâncias do iBeacon.

↳ DICA: use uma variável global para contar e outra para a soma acumulada.

Em seguida, envie um dicionário com essa distância média e as coordenadas x e y e o nº da sua bancada no tópico "distancia/NUMERO_DA_BANCADA".

2.8 4.5 1.9 5.6 3.6 → média = 3.7

Coordenadas das Bancadas do Laboratório

01	02	03	04	05	06	07	08
(3.0, 2.5)	(4.2, 2.5)	(5.4, 2.5)	(6.6, 2.5)	(7.8, 2.5)	(9.0, 2.5)	(10.2, 2.5)	(11.4, 2.5)
(3.0, 0.5)	(4.2, 0.5)	(5.4, 0.5)	(6.6, 0.5)	(7.8, 0.5)	(9.0, 0.5)	(10.2, 0.5)	(11.4, 0.5)
16	15	14	13	12	11	10	09

(0, 0)

exemplo de mensagem
da bancada 02

```
{
  "x": 4.2,
  "y": 2.5,
  "distancia": 3.7,
  "bancada": 2
}
```

4

No Node-RED, ao receber esse tópico,

use o bloco Change para definir o payload para ser igual à distância recebida.

Em seguida, use o bloco Gauge para exibir essa distância no dashboard.



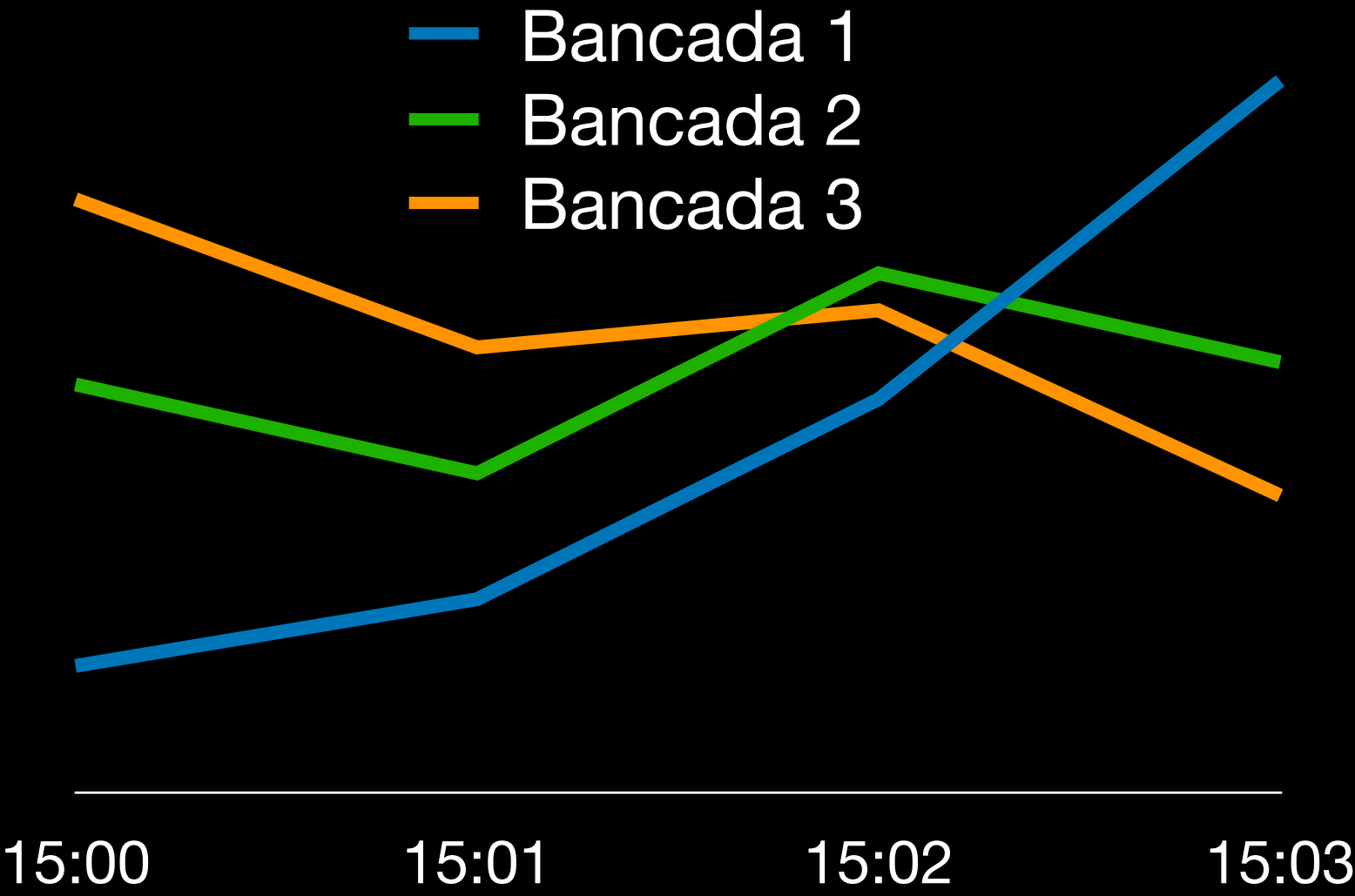
Protótipo: Monitoramento do Beacon

Integração

Distâncias entre Bancadas e Beacon

x da bancada	y da bancada	distancia até Beacon
3.0	2.5	5.4
4.2	0.5	1.2
9.0	2.5	7.3
...		

Histórico de Distâncias



Dispositivo #4 desconectado!

1

No Node-RED, inscreva-se no tópico "distancia/+".

Use o bloco Join configurado em modo Manual para criar uma Matriz (lista) que agrupe as mensagens a cada 10 segundos.

Gere uma tabela no Dashboard com os resultados (apagando dados antigos).



2

Use o bloco Change para inserir a data/hora atual no payload.

Gere um gráfico de linha com os históricos das distâncias de cada bancada.



3

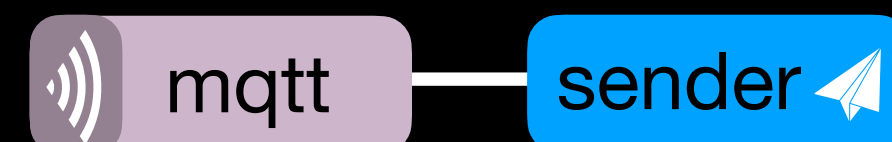
No ESP-32, use a função `mqtt.setWill` para enviar um tópico "desconectado" com o número da sua bancada quando o ESP32 perder a conexão.

Desconecte o cabo USB e veja o resultado no MQTTX.



4

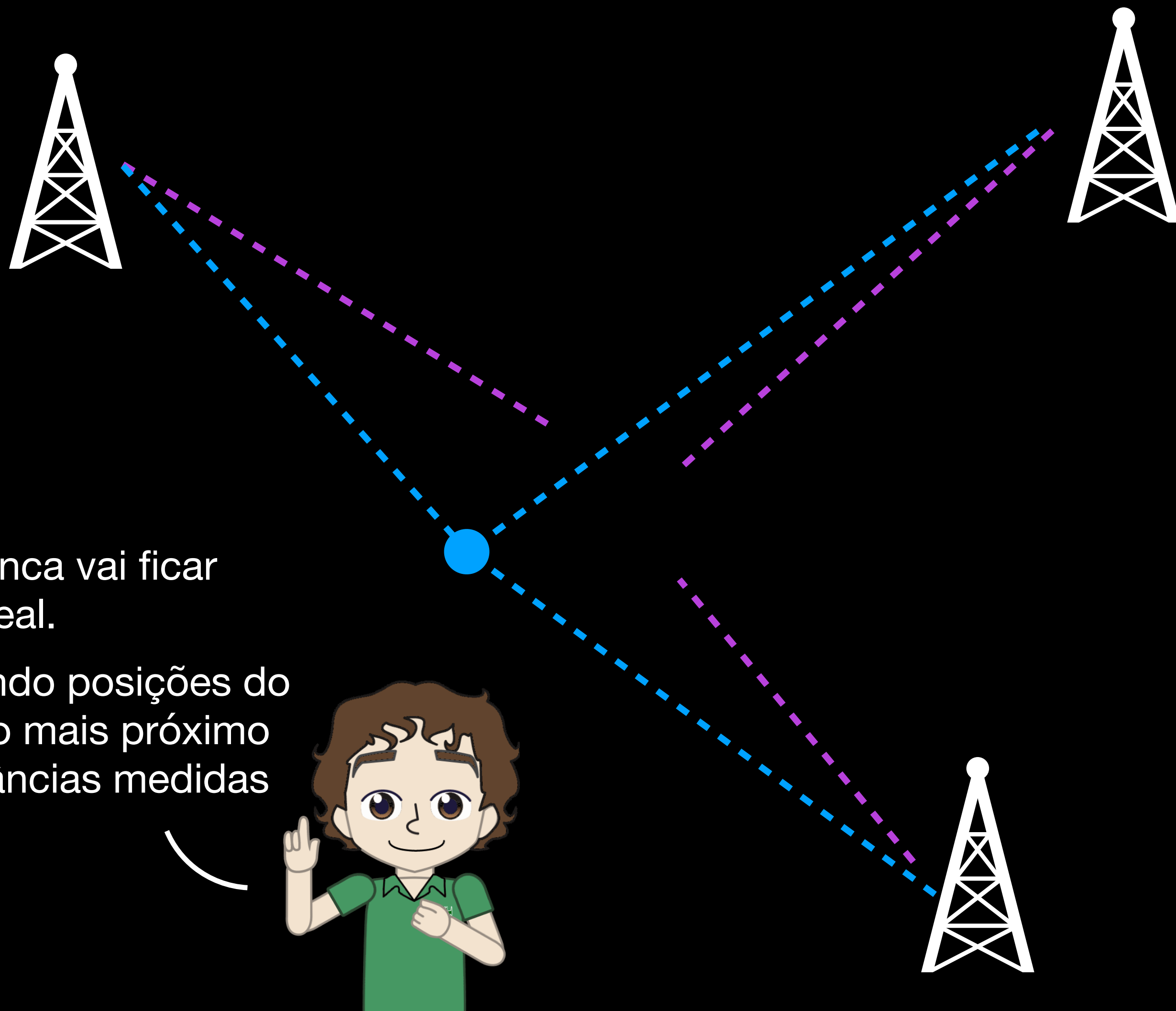
No Node-RED, envie uma mensagem para o Telegram quando qualquer dispositivo da sala for desconectado, incluindo seu número na mensagem.



Dispositivo #4 desconectado!

Integração

Desafio Extra



A triângulação nunca vai ficar certinha na vida real.

Eu posso ir testando posições do Beacon até ficar o mais próximo possível das distâncias medidas

Minimização do Erro Médio Quadrático das Distâncias Até o Beacon

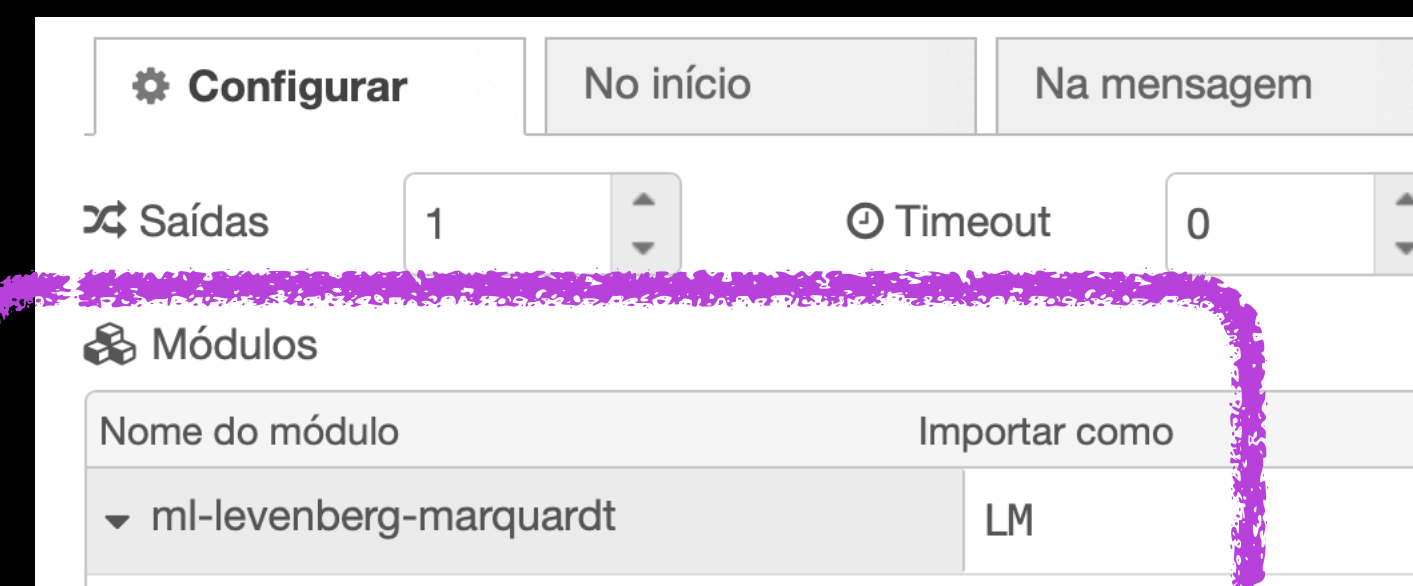
```
var bancadas = [
  { x: 3.0, y: 2.5 }, { x: 4.2, y: 2.5 }, { x: 3.0, y: 0.5 }, // ...
];
```

```
var distanciasMedidas = [
  1.9, 1.3, 2.1, // ...
];
```

```
var chuteInicialBeacon = [7, 1.2]; // x e y do meio da sala
```

```
function funcaoDistanciaParaOtimizar(chuteCoordenadasBeacon) {
  return (bancada) => Math.sqrt(
    Math.pow(chuteCoordenadasBeacon[0] - bancada.x, 2) +
    Math.pow(chuteCoordenadasBeacon[1] - bancada.y, 2)
  );
}
```

```
var resultado = LM.levenbergMarquardt(
  { x: bancadas, y: distanciasMedidas },
  funcaoDistanciaParaOtimizar,
  { initialValues: chuteInicialBeacon },
);
var xEstimadoBeacon = resultado.parameterValues[0];
var yEstimadoBeacon = resultado.parameterValues[1];
```




Exemplo de Otimização da Distância em Javascript

1

No Node-RED, use um bloco Function que receba um payload com a lista de x, y e distâncias e retorne um payload com o x e y estimados do Beacon.

↳ DICA: adapte o exemplo do slide anterior, gerando as listas de bancadas e distanciasMedidas a partir do payload recebido do bloco join da Integração.

```
[
  {"x": 3.0, "y": 2.5, "distancia": 1.9},
  {"x": 4.2, "y": 2.5, "distancia": 1.3},
  ...
]
```

→  function → {"x": 5.3, "y": 1.2}



2

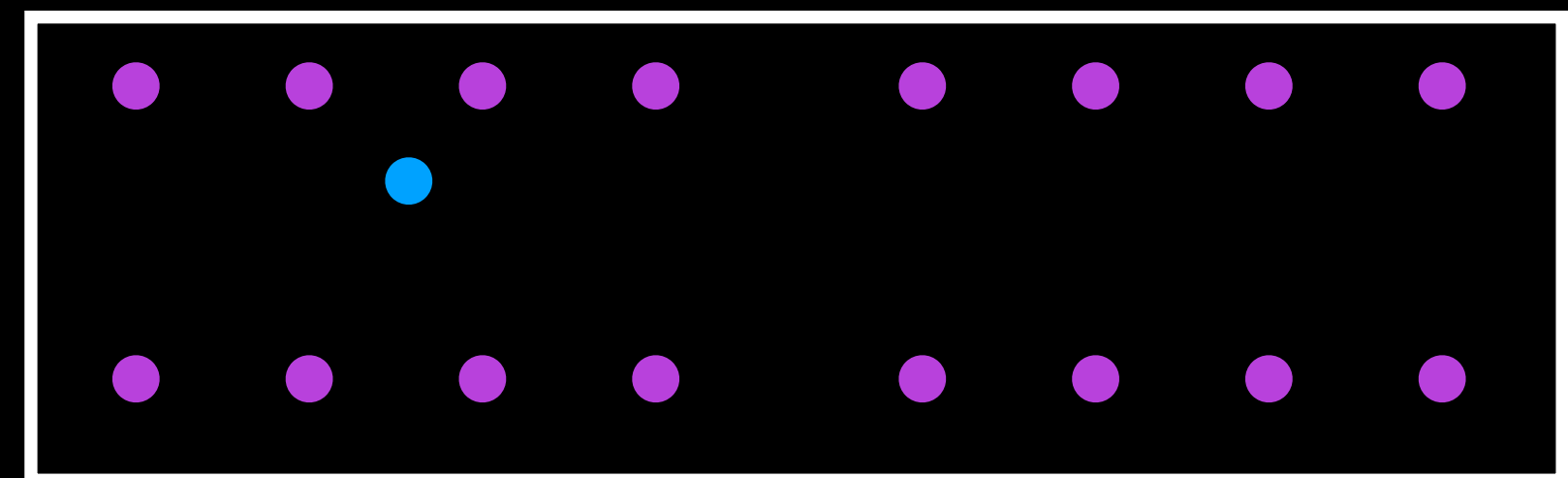
Crie um payload com um array de objetos com campos x, y e tipo, para todos ESPs e a estimativa do beacon, seguindo o modelo abaixo.

Exiba a posição do beacon e a de todos os ESP32 num gráfico tipo scatter.

↳ DICA: use a configuração de substituir os dados e de buscar os pontos por key: em vez de msg:.

```
[
  {"x": 3.0, "y": 2.5, "tipo": "bancada"},
  {"x": 4.2, "y": 2.5, "tipo": "bancada"},
  ...
  {"x": 6.4, "y": 1.3, "tipo": "beacon"}
]
```

 bancada  beacon



Desafio



janks.link/iot/projeto03.zip

Material do Projeto 03