

## Informações sobre a roleta

*como montar - como instalar - troubleshooting*

### 1) Estrutura e circuito

#### a) Verificar que os imãs e circuito estão no lugar.

A bicicleta está com o circuito e imãs presos por fita isolante (que pode soltar) portanto verifique antes de usar se não soltou (apertando os imãs e fita - são 16 imãs) e verifique também que o circuito está bem preso. A bicicleta deve ser usada de cabeça para baixo (mantenha sempre o guidão alinhado para que ela não caia - se achar mais seguro prenda no chão com alguma fita).

#### b) Reconectar reed (apenas se necessário)

O circuito tem um protoboard preso na bicicleta de onde saem dois fios para uma chave magnética reed, posicionada de forma a ser acionada quando os imãs passarem perto na hora que a roda girar. Se os fios se soltarem, é só recolocar. A chave não tem polaridade. Deve ter um fio preso no GND (negativo) e outro na saída D3 do ESP8266.

#### c) Fixar componentes (apenas se necessário - se cair ou soltar)

Verifique se os componentes do circuito não se soltaram (às vezes, se mexer muito, a chave fica um pouco frouxa). Eles devem estar bem encaixados no protoboard. Existe um desenho do circuito que pode ser usado como guia (anexo).

#### d) Alimentação (3,3V e/ou USB) - use bateria e também o USB (por segurança)

O ESP8266 puxa muita corrente, e a bateria pode estar fraca. Então o ideal é plugar o cabo USB no ESP e ligar em um computador (o cabo está na bicicleta, perto do selim). Mesmo com o cabo, aperte o botão de ligar (o ESP pisca um LED na hora que liga) pois é bom ter fornecimento de energia também da pilha (o ESP puxa muita corrente quando está transmitindo muitos dados).

### 2) Ambiente necessário no computador

#### a) Ambiente

O computador usado deve ter o seguinte:

- Servidor **mosquitto** instalado e rodando na porta padrão (1883)
- Ambiente **Java 8** com **Maven** e servidor Java EE 7 (eu uso Java 8 + **Eclipse Neon** com Maven 3 + **WildFly 10**).
- Acesso a **internet** (pois a página Web usa um CDN do Query)

#### b) IP (importante)

O IP deste computador precisa ser conhecido. Ele está hardwired no programa que é gravado no ESP8266 (podemos melhorar isto depois). **O ESP (cliente MQTT) precisa saber o IP do computador** onde está instalado o servidor Mosquitto

(servidor MQTT). Para alterar, é preciso usar um computador que tenha a IDE do Arduino + driver para ESP8266 genérico, e configurá-la para a comunicação com o ESP-8266. **Esta alteração é feita apenas uma vez** - se o mesmo computador for usado nos outros sorteios e não mudar o IP, não precisa fazer mais). Detalhes estão na seção seguinte.

(O programa do ESP tem **hardwired** o IP, o nome da rede e a senha – se qualquer um desses itens mudar, é preciso subir novo programa)

### c) Código-fonte

O código-fonte está em <https://github.com/helderdarocha/IoTProjects>

## 3) Programação / inicialização do ESP8266 (se necessário - se o IP ou rede ou senha for diferente)

Para reprogramar é necessário ter ou instalar um ambiente contendo a **IDE do Arduino + biblioteca ESP8266 genérica + driver**, e selecionar a placa e driver na IDE. Para transferir (upload) o programa é preciso apertar o botão RESET no ESP (à esquerda da entrada USB), depois **segurar o botão FLASH** (à direita da entrada USB) até iniciar o upload (aparecem uns pontinhos indicando que o upload está sendo feito). Às vezes falha e é necessário repetir algumas vezes.

Depois que o upload terminar com sucesso, aperte o RESET. Se o ESP estiver conectado ao computador, abra o **Serial Monitor** para ver se ele está conectando na rede. Quando conectar desligue o serial monitor para não prejudicar a performance, e pode fechar o ide do arduino.

O ESP funciona sem cabo. Depois de instalar é só ligar o botão. Mas como a bateria está fraca, o ideal é operar durante o sorteio com o cabo ligado.

## 4) Aplicação Web

Não será necessário mexer no código. Apenas **trocar o arquivo** a cada sorteio (podemos melhorar isto depois), e depois redeploy.

### a) Inicialização (uma vez):

O componente lê \*um\* arquivo XLSX chamado "**dados.xlsx**" localizado no mesmo pacote que a classe (**iot/web/roulette**) durante a inicialização, extrai os registros de uma coluna selecionada, embaralha os registros e grava em um ArrayList.

### b) Operação (a cada pulso):

Componente Camel **from** é cliente MQTT e conecta-se ao **Mosquitto** como subscriber - a cada pulso, lê o próximo elemento do ArrayList (criado e embaralhado na inicialização) no componente Camel **process**, e manda para a conexão

WebSocket (que é aberta pela aplicação Web com um Refresh na página) – no componente Camel **to**.

Para testar a aplicação Web sem servidor MQTT ou quando não houver pulsos, comente a linha **from** do Camel que define o endpoint **mqtt**, e descomente a linha **from** que usa um **timer**. Nesse modo, um timer irá gerar os pulsos. Assim é possível testar as conexões do WildFly e WebSocket sem precisar da bicicleta estar rodando.

## 5) Arquivo de dados

- a) Se arquivo for CSV, importe numa planilha nova Excel **formato XSLX** (verifique que as colunas estão corretamente separadas). Grave como **dados.xlsx** em uma pasta (talvez com o nome do csv original). Crie uma pasta para cada sorteio, com um arquivo dados.xlsx dentro.
- b) Se o arquivo for XLS converta antes para XLSX
- c) Para sortear, **copie o arquivo dados.xlsx** correspondente para a pasta **src/main/resources/iot/web/roulette** do projeto maven, depois rode comando **maven clean install**, faça **deploy** e rode no servidor.
- d) Se não funcionar verifique as exceptions no log do WildFly. Elas podem acontecer se o arquivo não estiver no lugar correto, se o nome estiver errado, se não houver permissões para abri-lo, se o formato não for XLSX, se a coluna de dados não for a coluna 0 (pode-se alterar isto no código)

## 6) Passo a passo para executar a roleta

- a) No computador, rode o **mosquitto** (ex: /usr/local/sbin/mosquitto - se tiver problemas, use /usr/local/sbin/mosquitto -v - modo verbose - mas isto impacta um pouco na performance)
- b) Ligue o **ESP** (aperte o botão da bateria + conecte USB). Se o Mosquitto estiver no modo **-v** deve aparecer a informação que um cliente conectou. Se nada estiver acontecendo, use o Monitor Serial do IDE do Arduino para ver se o ESP está conectando em WiFi)
- c) Rode a **aplicação Web** no servidor. Este é um segundo cliente que irá registrar no mosquitto (pode aparecer essa informação se o mosquitto estiver com -v)
- d) Acesse **http://localhost:8080/iot-web-roulette-1.0.0/** Abra a página (refresh se necessário). Agora é só girar a roleta e a página deve responder.

## 7) Múltiplos sorteios

Para cada novo sorteio, é preciso:

- a) **interromper** o servidor web
- b) trocar o arquivo de dados apagando o arquivo existente em **/src/main/resources/iot/web/roulette/dados.xlsx** e copiando o arquivo do novo sorteio para **/src/main/resources/iot/web/roulette/dados.xlsx**
- c) **maven clean install + deploy** no servidor + **reload** na página web