

IoT aplicado à ambiente aviário: coleta, comunicação, armazenamento e disponibilização

Elaborador:	Vinicius Fernandes Moraes
Orientador:	Prof. Dr. José Luís Zem

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

M824i MORAES, Vinícius Fernandes

IoT aplicado à ambiente aviário: coleta, comunicação, armazenamento e disponibilização. / Vinícius Fernandes Moraes. – Americana, 2018.

32f.

Monografia (Curso de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Dr. José Luís Zem

1 Internet das coisas 2. Arduino I. ZEM, José Luís II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518

681.31

Vinícius Fernandes Moraes

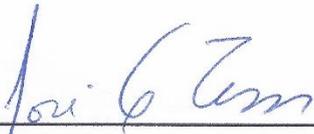
**IoT aplicado à ambiente aviário:
coleta, comunicação, armazenamento e disponibilização.**

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.

Área de concentração: Segurança da Informação

Americana, 14 de Fevereiro de 2019.

Banca Examinadora:



Prof. Dr. José Luís Zem

Professor Doutor

Fatec Americana



Prof. M.e Diogenes de Oliveira

Professor Mestre

Fatec Americana



Prof. Esp. Márcio Roberto Baldo Taglietta

Professor Especialista

Fatec Americana

SUMÁRIO

1. INTRODUÇÃO	5
2 REFERENCIAL TEÓRICO	6
2.1 Sensores.....	6
2.1.1 LDR.....	8
2.1.2 DHT	9
2.1.3 Termopar	11
2.2 Atuadores.....	12
2.2.1 Atuadores Eletromagnéticos	13
2.2.2 Atuadores Hidráulicos e Pneumáticos.....	13
2.2.3 Relê	13
2.3 MQTT.....	14
2. O AMBIENTE	17
2.1. O Laboratório Aviário	18
3. ATIVIDADES DESENVOLVIDAS	23
3.1. Possibilidades de Melhoria na Segurança.....	27
4. CONCLUSÕES E CONSIDERAÇÕES FINAIS	30

LISTA DE FIGURAS

Figura 1 – Tipo de Estímulo	6
Figura 2 - Estrutura de um dispositivo de sensoriamento	7
Figura 3 - Vista superior e corte lateral do LDR	9
Figura 4 - DHT22.....	10
Figura 5 - Circuito fechado com ligas metálicas distintas.....	12
Figura 6 - Diagrama esquemático do Relê	14

LISTA DE TABELAS

Tabela 1 - Sensores Naturais	7
Tabela 2 - Características Elétricas	10
Tabela 3 - Limites de erro para Termopares.....	11
Tabela 4 - Termopar com suas Ligas Metálicas	12
Tabela 5 - Cabeçalho do Protocolo MQTT	14
Tabela 6 - Tipos de Mensagem	15
Tabela 7 - Níveis de QoS	16

1. INTRODUÇÃO

Internet das Coisas, mais conhecida como IoT, que vem do inglês *Internet of Things*, virou um marco na sociedade pelo dinamismo na comunicação, no uso da tecnologia móvel e sensores instalados em dispositivos capazes de captar informações de diversos aspectos. A evolução da tecnologia trouxe novos aparatos tecnológicos, a partir de nanotecnologia, *wireless* e sensores, permitindo comunicação externa de um dispositivo, deixando tudo conectado entre si, como: ventiladores, alarmes, portas de garagem, semáforos, etc...(Carvalho; Souza, 2015).

Aqui será apresentado um relatório técnico de um projeto onde auxiliei com a infraestrutura de rede dum projeto de doutorado para o controle de ambiente aviário autônomo, com aviário do tipo Convencional, onde sua importância visa em mensurar e analisar o ambiente e manter o conforto térmico das aves e garantindo uma maior produtividade de frangos de corte. O autor e responsável deste projeto de doutorado mencionado foi o Doutorando Marcelo Oliveira, onde seu projeto e tese ainda está em desenvolvimento.

Assim, o processo mais utilizado nesses tipos de aviário, é o controle manual, onde o responsável pelo aviário se desloque até o aviário para medir a temperatura e umidade em seu interior, em horas pré-determinadas, como também saber a incidência de sol interna no aviário, que irão influenciar na temperatura, dado determinada hora do dia deve se ligar os ventiladores para a sua refrigeração.

Tendo em vista o processo manual utilizado, foi desenvolvido o projeto para automatizá-lo, sendo feitas medições periodicamente via sensores conectados à rede (com conceito de IoT), armazenando os dados em um banco de dados que fica em um servidor, para se monitorar os dados do aviário mais precisamente. Sendo então feito esse controle automático, monitorando a incidência de luz do sol, umidade e temperatura interna, ao decorrer de todo o dia e apenas acessando um website com uma interface demonstrativa para o usuário administrador do aviário, também acionando automaticamente os ventiladores assim que necessários, aumentando seu controle e produtividade. (Oliveira *et al.*, 2018)

Tendo em vista esse projeto, abordaremos um exemplo de infraestrutura de uma rede de sensores sem fio, onde há a coleta de dados e seu armazenamento, e a disponibilização desses dados para análise no qual o fim o sensor foi inserido, podendo o sistema tomar algumas atitudes automaticamente em relação a essa análise.

1 REFERENCIAL TEÓRICO

Para poder entender melhor sobre o projeto que será explicado, será necessário entender alguns assuntos, que não foi abordado pelo curso, mas são muito interessantes. Informando que é possível integrar tecnologias diferentes. Então esses assuntos foram abordados para um entendimento melhor de suas tecnologias.

1.1 Sensores

Os Sensores usam um sinal elétrico que foi estimulado por um determinado tipo de sinal, de acordo com sua propriedade, condição ou quantidade. Ele traduz um sinal não elétrico para elétrico, podendo amplificar ou canalizar por algum dispositivo eletrônico usado. A Saída pode se apresentar em corrente e carga ou elétrica, descritas em termos de amplitude, fase e frequência, sendo designado em formato de sinal de saída. Dessa forma os sensores podem receber sinais de entrada de qualquer tipo, tendo propriedades elétricas de saída, como na Figura 1. (Souza; Carvalho, 2011)

Falando tecnicamente, eles sofrem uma variação entre as características internas em função da variação da grandeza externa, como um Sensor Resistivo de Temperatura que em função da temperatura ele varia sua resistência elétrica, ou um Sensor Capacitivo de Umidade do Ar que em função da umidade do ar varia sua capacitância. (Martins; Vianna, 2010)

Figura 1 – Tipo de Estímulo



Fonte: Souza; Carvalho, 2011

Souza; Carvalho (2011) ainda dizem que sensor associa a um dispositivo construído pelo homem, no entanto definem também como uns sensores naturais, como os 5 sentidos do corpo humano. Conforme na Tabela 1 demonstra esses *sensores naturais* sintetizados correspondentes nos dispositivos análogos de funcionamento.

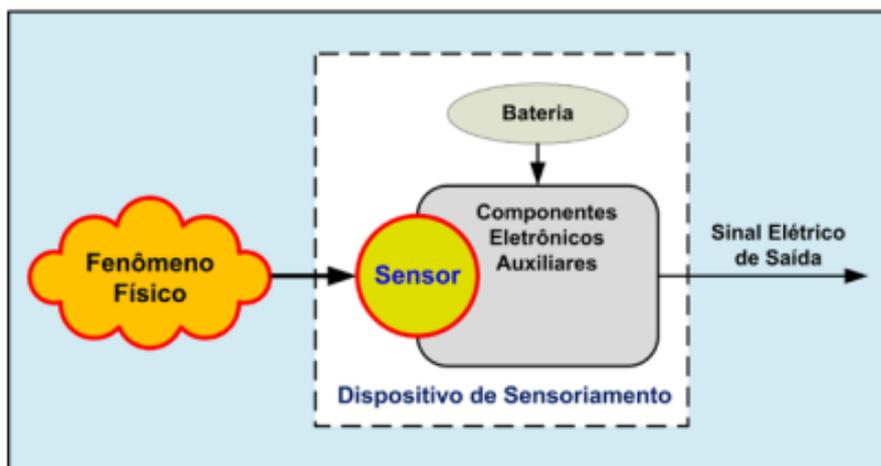
Tabela 1 - Sensores Naturais

Sentido	Tipo de Sinal	Propriedade	Sensor	Dispositivo Análogo
Visão	Radiante	Intensidade luminosa e comprimento de onda da luz	Bastonetes e cones da retina	Filme fotográfico, fotodiodo, fototransistor
Audição	Mecânico	Intensidade e frequência do som	Caracol (cóclea) no canal auditivo interno	Microfone
Tato	Mecânico	Pressão, força	Terminais dos nervos	Potenciômetro e LVDT. Detectores ópticos e sensores matriciais tácteis
Olfato	Químico	Odores	Papilas olfativas no nariz	Nariz eletrônico
Paladar	Bioquímico	Proteínas	Papilas gustativas na língua	-

Fonte: Souza; Carvalho, 2011

Muitas vezes os sensores precisam de uma fonte de alimentação externa, e talvez um conjunto de componentes eletrônicos a mais, tornando um circuito para fornecer a saída de sinal elétrico proporcionalmente a grandeza medida, como demonstrado na Figura 2 (Marthins; Vianna, 2010)

Figura 2 - Estrutura de um dispositivo de sensoriamento



Fonte: Marthins; Vianna, 2010

Há também sensores que não precisam de uma fonte externa de alimentação, onde eles por si só já geram a tensão elétrica de acordo com sua excitação, como os termopares que por sua variação de temperatura gera uma tensão equivalente. (Marthins; Vianna, 2010)

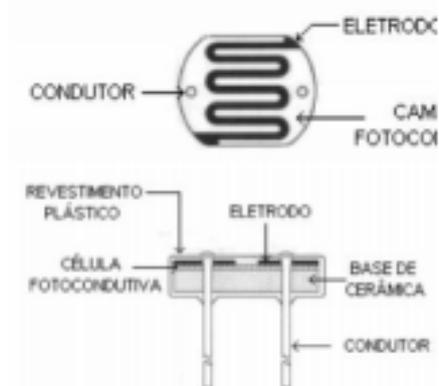
A classificação dos sensores pode ser genericamente de dois tipos, os Passivos e os Ativos. Os Ativos precisam de uma fonte de energia que produz um sinal de excitação, onde essa excitação que passa pelo sensor, produz o sinal de saída, como o sensor de passagem, de intensidade de corrente ou de movimento. Os Passivos já não precisam de alguma fonte de energia, eles geram diretamente um sinal elétrico apenas pelo estímulo externo, como o Termopar, o Sensor de PH ou o Sensor Piezoelétrico. (Souza; Carvalho, 2011).

1.1.1 LDR

Light Dependent Resistor (Resistor Dependente da Luz), é um dispositivo semicondutor, com dois terminais que possuem resistência conforme a incidência de luz sobre ele, é um elemento não polarizado, sendo assim a corrente pode circular por ambos os lados. (Mendes; Stevan, 2013). Essa incidência de luz tecnicamente mede o *iluminamento*, que é uma grandeza expressada em Lux, e representa uniformemente distribuído sobre uma área de 1m² a incidência de 1 *lúmen*. (Marthins; Vianna, 2010).

Sensores com elementos condutores eletrostáticos tem a diferença de energia entre a banda de valência e a banda de condução nula, sem obstáculos para a passagem de corrente, e com materiais isolantes se comportam de maneira inversa, diferentemente dos semicondutores, como o LDR, possui diferença de energia entre os condutores e isolantes necessitando ainda de alguma força externa, como a luminosidade, pra conduzir eletricidade. Podemos definir que a luz entrega energia com um comprimento de onda determinado, através de fótons, onde o feixe luminoso incide na superfície do LDR os elétrons do material começam a interagir os fótons ficarem com a mesma frequência, assim o material dispõe os elétrons livres fazendo com que diminua sua resistência, passando mais energia elétrica, quanto menos luminosidade falta elétrons livres, no entanto aumenta sua resistência, deixando passar menos energia. Podemos ver na Figura 3 o desenho do LDR em vista de cima e de corte lateral, demonstrando o material fotocondutivo (o semicondutor), em cima de uma base de cerâmica com espaços para incidência de luz, ligados por um eletrodo à camada condutiva. Existem LDRs de 5, 7, 10 e 25 mm de diâmetro, sendo que quanto maior seu tamanho, maior sensibilidade e capacidade de suportar maiores correntes e temperaturas. (Marthins; Vianna, 2010)

Figura 3 - Vista superior e corte lateral do LDR



Fonte: Marthins; Vianna, 2010

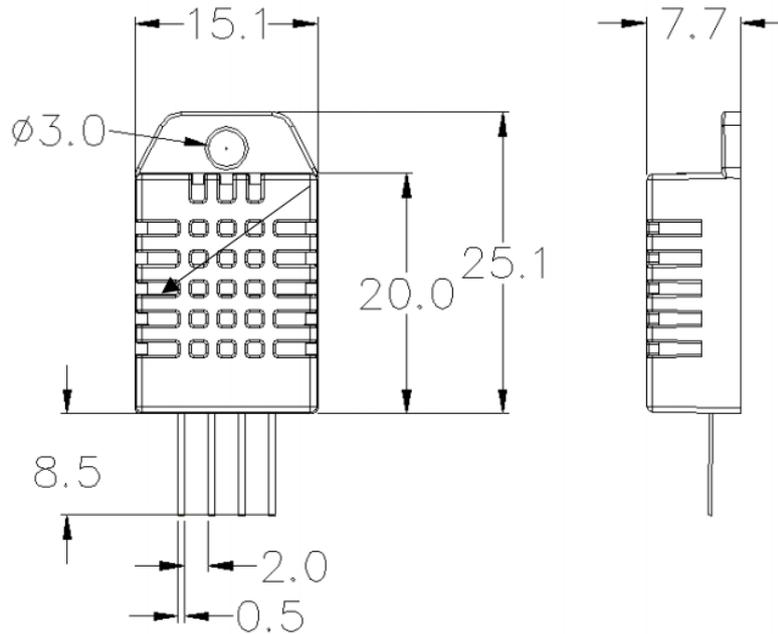
1.1.2 DHT

É um sensor de alta precisão, para medir a umidade, garantindo a estabilidade e confiabilidade. Há um microcontrolador de 8 bits internamente com uma interface que trata o sinal para digital. Possui um tipo de memória OTP, com um cociente de calibração para compensar em temperatura, quando detectar uma temperatura ele está acessando esse cociente de memória, fazendo assim também leituras de temperatura. (Aosong Electronics, 2013).

É um tipo de sensor capacitivo, que trabalha de 3 à 5.5 V, com saída digital de um fio, mede a umidade entre 0% à 99,9% e temperatura de -40 à 80°, precisão de leitura de umidade varia +-2% e temperatura +-0,5°C, conseguindo medir até de 2 em 2s. (Aosong Electronics, 2013).

Aosong Electronics (2013) demonstrada com a Figura 4, as dimensões de 25.1x15.1x7.7 mm, possui 4 terminais de ligação, vendo-o de frente o primeiro é o *VDD*, que seria a entrada de energia; o segundo o *Data* de sinal de dados para comunicação com o microcontrolador; o terceiro é nulo, sem uso; e o quarto é o *GND*, que seria o terra.

Figura 4 - DHT22



Fonte: Aosong Electronics, 2013.

O dispositivo ainda conta com um sistema de barramento único para comunicação, esse barramento passa apenas um dado por linha para o sistema de troca de informação, controlando a passagem de dados até que se complete, para poder passar mais dados. O microprocessador interno através do *open-drain* ou a porta *tri-state*, fica conectado com a linha de dados, não deixando que o dispositivo envie dados enquanto o anterior não for completado. Usando um único barramento geralmente precisa de um resistor externo de 5.1 K Ω , por quando o barramento estiver ocioso o status é alto. Isso porque são de uma estrutura mestre-escravo, que fica aguardando uma chamada do microcontrolador externo para poder responder, ainda sim o microcontrolador deverá respeitar a sequência do barramento único, porque se houver uma sequência de confusão ao solicitar dados antes que complete a linha de troca, que seria os 2s já ditos anteriormente, o sensor não responderá. (Aosong Electronics, 2013).

Podemos ver também através da Tabela 2 as características elétricas deste dispositivo.

Tabela 2 - Características Elétricas

Itens	Condição	Mínima	Típica	Máxima	Unidade
Tensão de Alimentação	DC	3.3	5	6	V
Intensidade de Corrente	Medindo	1		1.5	mA
	Ocioso	40	nula		μ A
Período de Coleta	Segundos		2		Segundos

Fonte: Aosong Electronics, 2013

1.1.3 Termopar

Os Termopares são divididos em 3 categorias de tipos, os Padronizados de Metal Base (K, J, N, E, T), os Padronizados de Metal Nobre (R, S, B), e os que não são definidos por letras nos seus tipos. A diferença entre os Padronizados de Metal Nobre e Base, é que os de Metal Nobre são compostos por Platina e os de Metais Base são compostos por Níquel. São usados para medir temperaturas de superfície muito altas ou muito baixas, tendo cada tipo de acordo com sua composição suporta um range diferente de temperatura, assim como demonstrado na Tabela 3. (Moreira, 2002)

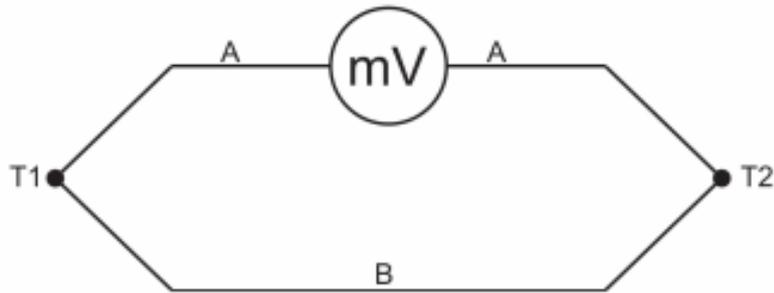
Tabela 3 - Limites de erro para Termopares

Sensor (Tipo)	Faixa de Utilização (°C)	Limite de Erro (escolher o maior)	
		Padrão	Especial
E	0 – 870 -200 a 30	± 1,7 °C ou 0,5% ± 1,7 °C ou 1%	± 1 °C ou 0,4% -
J	0 - 760	± 2,2 °C ou 0,75%	± 1,1 °C ou 0,4%
K, N	0 – 1260 -200 a 30	± 2,2 °C ou 0,75% ± 2,2 °C ou 2%	± 1,1 °C ou 0,4%
T	0 – 370 -200 a 30	± 1 °C ou 0,75% ± 1 °C ou 1,5%	± 0,5 °C ou 0,4% -
R, S	0 - 1480	± 1,5 °C ou 0,25%	± 0,6 °C ou 0,1%
B	870 - 1700	± 0,5%	-

Fonte: Moreira, 2002

As medições de temperatura são feitas através do “Efeito Seebeck” (nome do físico que descobriu a teoria). Em 1981, o Físico Seebeck descobriu que se dois metais distintos A e B, compondo um circuito fechado, com suas temperaturas diferentes nesses metais T1 e T2, apresentam uma Força Eletro-Motriz, em ordem de alguns milivolts (mV). Seebeck ainda constatou que quanto maior a diferença de temperatura entre os metais, maior é a Força Eletro-Motriz (f.e.m), que também varia de acordo com a natureza dos metais A e B, como demonstrado na Figura 5, que também pode ser chamado de circuito termoelétrico. (Sense Sensors & Instruments, 2015)

Figura 5 - Circuito fechado com ligas metálicas distintas



Fonte: Sense Sensors & Instruments, 2015

Pares de metais podem ser utilizados em Termopares, mas alguns tiveram maior aceitação na prática industrial, quanto a sua precisão de calibração, resistência a corrosão e oxidação, f.e.m termal grande, relação entre f.e.m com temperatura aproximadamente linear. Os do Tipo K são feitos de Chromel (liga de 90% níquel e 10% cromo) no polo Positivo e, Alumel (liga de 94,5% níquel, associado a manganês, alumínio e silício, ou chamado de Ni-Cr/Ni) no polo Negativo. (Sense Sensors & Instruments, 2015).

Vemos na Tabela 4 os materiais de cada tipo de Termopar fabricado.

Tabela 4 - Termopar com suas Ligas Metálicas

TIPO	POSITIVO	NEGATIVO
T	Cobre	Constantan
J	Ferro	Constantan
E	Chromel	Constantan
K	Chromel	Alumel
R	Platina-Ródio 13%	Platina
S	Platina-Ródio 10%	Platina
B	Platina-Ródio 30%	Platina-Ródio 6%

Fonte: Sense Sensors & Instruments, 2015

1.2 Atuadores

Através de um sinal de energia elétrica, hidráulica ou pneumática, é convertido em potência mecânica, que é levada aos elos através dos sistemas de transmissão para que se movimentem.

1.2.1 Atuadores Eletromagnéticos

Usados em sua maior parte na robótica, em motores de corrente contínua e de passo. Como eles podem ser controlados por sinais elétricos utilizando controladores de movimento, são empregados para geração de força ou posição do robô, além da facilidade de programação de movimentos, ter também uma gama variedade de fabricantes disponíveis no mercado.

Nos Motores de Corrente Contínua (cc) seu torque se mantém numa faixa constante para grandes variações de velocidade, mas precisam fazer um controle de posição de malha fechada com sensores de posição angular e velocidade. Para seu melhor funcionamento é comum se ter redutores de velocidade, já que eles trabalham normalmente com velocidades elevadas, para com que a diminuição da velocidade haja um aumento no torque, que são precisos para gerar a potência mecânica.

Nos Motores de Passo, funcionam no controle de malha aberta em posição e velocidade, com seu torque reduzido e maior velocidade, com isso gerando vibrações mecânicas, bastante usados em garras.

1.2.2 Atuadores Hidráulicos e Pneumáticos

São cilíndricos e lineares, para movimentos lineares ou com motores para gerar deslocamento angular. Possuem uma unidade de comando com válvulas direcionais, que direcionam o deslocamento dos fluídos.

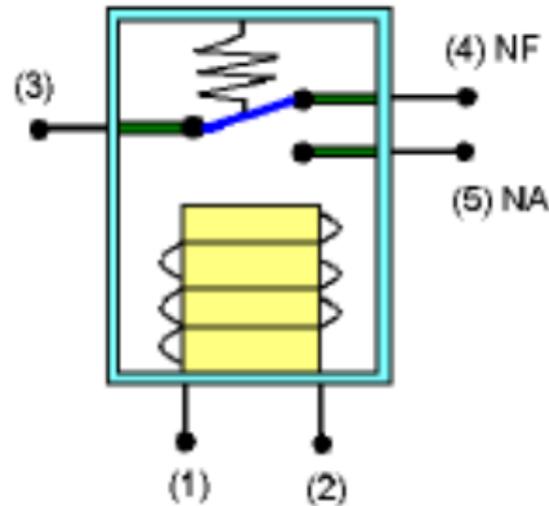
Os hidráulicos possuem uma elevada rigidez, pela incomplexidade de fluído (óleo), portanto é permitido o controle de posicionamento e velocidade, mas isso atrapalha no controle de força. Com uma bomba para fornecer o óleo às válvulas do atuador.

Os Pneumáticos são mais usados em robôs de carga onde é mais controlado as posições caracterizando o movimento ponto-a-ponto. Por sua baixa rigidez devido a compressão de ar comprimido em seus cilindros, suas operações são mais suaves, porém isso deixa menos preciso em seu devido posicionamento. (Romano, 2002)

1.2.3 Relê

Os relês permitem combinação de lógica de comando, separando os circuitos de comando do de potência. No Circuito de Comando a interface se trabalha com corrente (até 10A) e baixas tensões, que são utilizadas para o acionamento através de uma bobina, que aciona um terminal do Circuito de Potência, onde se encontra as cargas com maiores correntes e tensões para energizar e acionar algo, como motores, resistência de calor, etc. Como demonstrado na figura 5, os relês possuem terminais que energizam sua bobina nos pontos 1 e 2, com de 5Vcc à 24 Vcc, onde apenas pela força magnética aciona a chave com os terminais 3, 4 e 5, sem contato físico, onde trabalham com correntes e tensões maiores (110 Vca ou 220 Vca), fazendo o acionamento ou desligamento do outro dispositivo maior que se está querendo acionar. (Souza, 2018)

Figura 6 - Diagrama esquemático do Relê



Fonte: Souza, 2018

1.3 MQTT

É um Protocolo de Telemetria, inventado pela IBM em 1999 extremamente simples, para redes de baixa largura de banda, alta latência ou não confiáveis. Ideal para estruturas M2M (máquina a máquina) ou Internet das Coisas, por ser desenhado para garantir a confiabilidade e recursos de banda reduzidos. (International Business Machines Corporation (IBM) Eurotech)

Por ser simples, seu cabeçalho é pequeno, fixo, como exemplificado na Tabela 5. No byte 1, possui o tipo da mensagem em 4 bits enumerado com a cada tipo, conforme demonstrado na Tabela 6, juntamente com seus sinalizadores. No sinalizador DUP Flag, na posição de bit 3, é feito uma identificação da mensagem, definido quando é tentado entregar novamente as mensagens de PUBLISH, PUBREL, SUBSCRIBE ou UNSUBSCRIBE e o valor de QoS for maior que 0. Nos bits de posição 2 e 1, indicam os níveis de garantia de entrega QoS, demonstrados conforme a Tabela 7 esses níveis. E no último bit (de posição 0) o RETAIN, usado apenas quando o cliente envia uma mensagem de PUBLISH ao servidor, quando é estabelecido uma assinatura em determinado tópico, deverá ser enviado ao assinante com esse Sinalizador Retido definido; as mensagens retidas caso haja reinicialização do servidor. (International Business Machines Corporation (IBM) Eurotech)

Tabela 5 - Cabeçalho do Protocolo MQTT

bit	7	6	5	4	3	2	1	0
byte 1	Tipo da Mensagem				DUP Flag	Nível de QoS		RETAIN
byte 2	Restante/ mensagem							

Fonte: International Business Machines Corporation (IBM) Eurotech

Tabela 6 - Tipos de Mensagem

Mnemonic	Enumeração	Descrição
Reserved	0	Reservado
CONNECT	1	Solicitação para o cliente conectar ao servidor
CONNACK	2	Confirmação de conexão
PUBLISH	3	Publicação da mensagem
PUBACK	4	Confirmação da Publicação
PUBREC	5	Publicação recebida (parte 1 da confirm. de entrega)
PUBREL	6	Relançar a publicação (parte 2 da confirm. de entrega)
PUBCOMP	7	Publicação completa (parte 3 da confirm. De entrega)
SUBSCRIBE	8	Pedido de ouvir/subscrição ao cliente
SUBACK	9	Confirmação de subscrição
UNSUBSCRIBE	10	Pedido de cancelamento a subscrição do cliente
UNSUBACK	11	Confirmação de cancelamento da subscrição ao cliente
PINGREQ	12	Pedido de PING
PINGRESP	13	Resposta de Ping
DISCONNECT	14	Desconectar do client
Reserverd	15	Reservado

Fonte: International Business Machines Corporation (IBM) Eurotech

Tabela 7 - Níveis de QoS

Valor de QoS	bit 2	bit 1			
0	0	0	No máximo uma vez	Fogo e esqueça	≤ 1
1	0	1	Pelo menos uma vez	Confirmação de entrega	≥ 1
2	1	0	Exatamente uma vez	Entrega assegurada	$= 1$
3	1	1	Reservado		

Fonte: International Business Machines Corporation (IBM) Eurotech

Sua segurança pode ser implementada colocando-se usuário e senha no protocolo; a criptografia pode ser incrementada com SSL separadamente, porém o SSL exige uma maior taxa de processamento significativo à rede; seguranças adicionais podem ser implementadas com aplicativos que criptografa os dados trafegados, mas isso não são incluídos dentro do MQTT, para mantê-lo simples e leve. (International Business Machines Corporation (IBM) Eurotech)

2. O AMBIENTE

A Faculdade de Zootecnia e Engenharia de Alimentos (FZEA/USP) tem a missão de oferecer ensino, pesquisa, extensão e serviços de alta qualidade para a sociedade, na área de Produção Animal e de Alimentos, ligando fortemente com o agronegócio do Brasil, principalmente no Estado de São Paulo. Localizada no Campus Fernando Costa (USP/Pirassununga), Avenida Duque de Caxias Norte, 255 – Campus Universitário.

Neste campus estão localizados laboratórios de pesquisa e aprendizado, salas de aula, administração e salas de docentes, distribuídos em construções históricas, em cerca de 19000 m². Contendo modernos laboratórios, equipamentos e instalações adequados, operados por docentes e funcionários qualificados, com uma biblioteca moderna com acervo especializados em suas áreas.

Um dos laboratórios do ZAB é o Laboratório de Física Aplicada e Computacional (LAFAC/ZAB), onde são desenvolvidos projetos de eletrônica e computação para aplicar na indústria animal.

Atividades de Pesquisa e Ensino: atua principalmente no paradigma cérebro-computador realizando pesquisas em processamento digital provenientes da atividade elétrica cerebral em humanos e animais. Atua também, no desenvolvimento de instrumentação eletrônica inteligente e no estudo teórico e prático das redes neurais e artificiais.

Equipamentos: Laboratório de prototipação eletrônica; Sistema digital de aquisição de sinais biológicos; Sistema de gravação de microprocessadores e Laboratório de desenvolvimento de softwares. Analisador de espectro.

Durante o período de estágio foram desenvolvidas as seguintes atividades:

- I. Apoio na configuração e administração da infraestrutura de rede do laboratório, onde são desenvolvidos pesquisas e projetos dos alunos, de acordo com os recursos disponíveis, juntamente com a segurança dos dados no servidor local do laboratório, assim como a disponibilidade das informações e proteção de Firewall dos dados no servidor, como também na infraestrutura dos projetos a campo onde se utiliza tecnologia da informação (barracões, estufas).
- II. Participação na configuração do Servidor local e de um servidor remoto baseado na tecnologia Raspberry Pi.
- III. Auxílio em projetos desenvolvidos no laboratório que envolveram a Tecnologia da Informação, como utilização da rede, programação, servidores que recebem os dados, projetos com embarcados como Arduino e NodeMCU-ESP.
- IV. Acompanhamento de projetos, como também utilizando meus conhecimentos de redes e servidores para auxiliar alunos que não possuíam conhecimento na área de TI.

O relatório técnico apresentado neste texto refere-se aos itens II. e III.

As atividades desenvolvidas consistiam em participar de alguns experimentos relacionados a monitoramento de ambientes produtivos, ligados à área de Tecnologia da Informação neste meio, como por exemplo, monitoramento de ambiente aviário, onde se utiliza controle térmico através de sensores *IoT*, medindo temperatura do ar, umidade e quantidade de luz solar, bem como armazenando os dados em um banco de dados, com acesso remoto via sistema Web.

Estas atividades envolviam a participação em atividades de campo, com a aplicação prática dos conhecimentos adquiridos em sala de aula, complementando com a montagem dos hardwares (sistemas micro controlados) embarcados e de sensores de ambiente em rede.

Um exemplo desta aplicação são os dados coletados pelo sistema aquapônico localizado ao lado do Laboratório, o qual envia e recebe os dados de forma online. Como a estufa fica ao lado do laboratório, os sensores se comunicam com o roteador *Wi-Fi*. O mesmo procedimento foi realizado em um aviário experimental utilizando os softwares PHP, MySQL, MQTT, dentre outros, além de sensores e atuadores e tecnologia de comunicação sem fio com NodeMCU.

2.1. O Laboratório Aviário

A pesquisa do laboratório envolve dois prédios, o laboratório propriamente dito, com computadores e servidor, e um barracão aviário sem aves, para testes do protótipo para depois de pronto o teste ser aplicado com as aves vivas.

Foi utilizado um aviário do tipo convencional, localizado no campus da USP de Pirassununga, com dimensões de 29,70 x 8,04 m, contendo 4 ventiladores e 4 aspersores, onde são ligados para deixar o ambiente confortável dentro do barracão, com as laterais abertas tendo cortinas de lona para rebater o sol e a chuva (Figura 1).

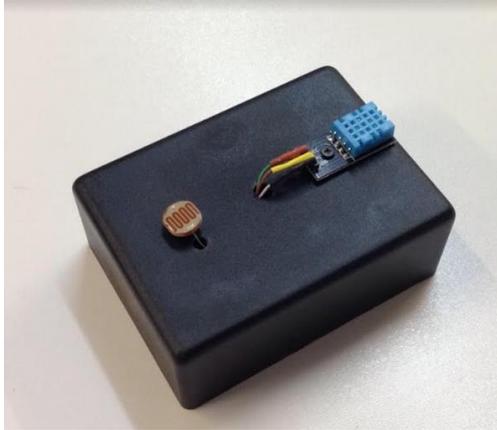
Figura 7 - Foto de dentro do aviário



Fonte: Autoria própria

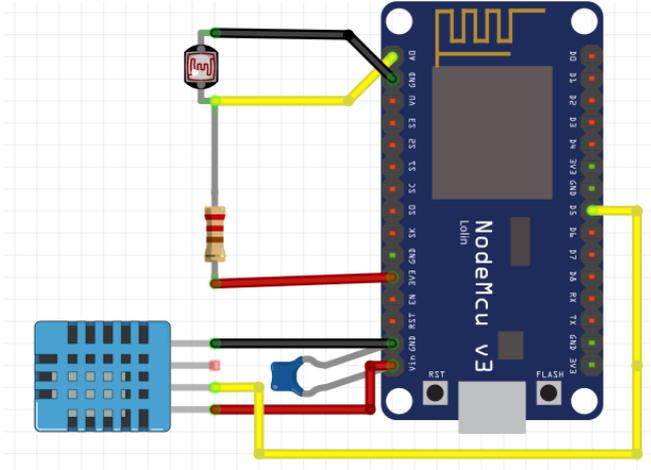
Para mensurar a temperatura e umidade de dentro do aviário, foi construído um hardware que o chamaremos de “Sensor” (Figura 2), onde é composto por um sensor *LDR* para medir a quantidade de luz e saber a incidência de sol no ambiente, um sensor *DHT 22* que mede a umidade entre 0% à 100% e a temperatura entre -40° à 125° Celsius, conectados a uma placa micro controladora open source *NodeMCU*, que utiliza um processador *Wi-Fi ESP8266*, preparada para usar *IoT*. Os sensores citados, estão interligados para seu correto funcionamento com um capacitor no *DHT* evitando a perda de corrente e um resistor pra ser feito o controle da leitura do *LDR* e funcionar com a tensão correta, conectados conforme o esquema elétrico na Figura 3. Distribuídos em cada quadrante do aviário (Figura 4), no qual foi dividido, de acordo com os ventiladores e aspersores do local, totalizando 8 desses sensores, junto com mais 3 utilizando um outro tipo de sensor de temperatura, *Termopar* (Figura 5), para medir a superfície das paredes e cumieira do aviário. Este sensor, é um outro sensor, onde utiliza também uma placa *NodeMCU*, com o diferente de utilizar o sensor *termopar* tipo-k, que são interligados por seu módulo. A cada leitura destes, são enviados esses dados ao Servidor pelo protocolo *HTTP*, feito leitura e envio num determinado período de tempo. Junto também é enviado o *Status* do Atuador que deverá estar via protocolo *MQTT*, para que ligue ou desligue.

Figura 8 - Sensor



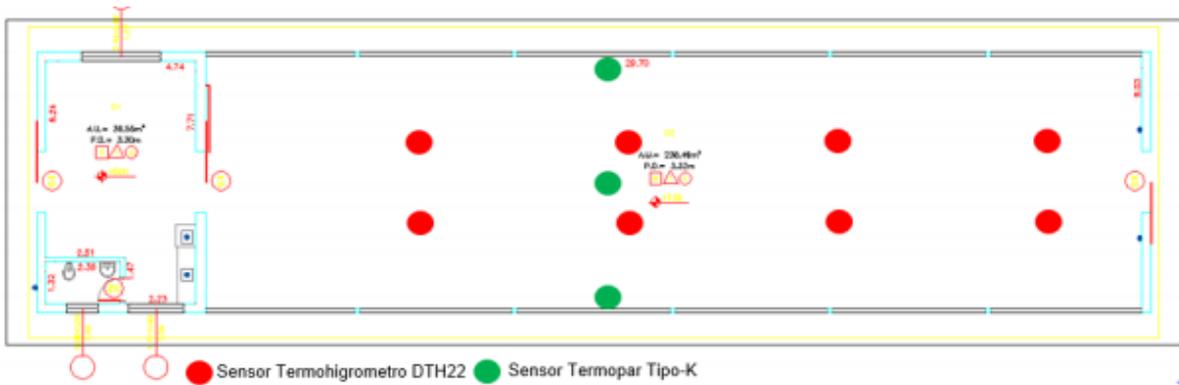
Fonte: Autoria própria

Figura 9 - hardware sensor



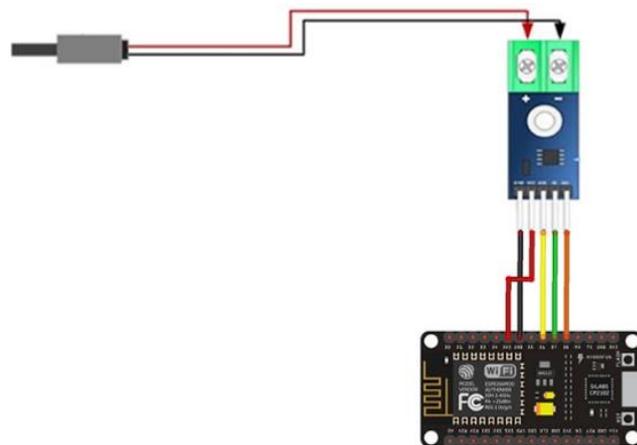
Fonte: Autoria própria

Figura 10 - Distribuição dos sensores



Fonte: Oliveira et al., 2018

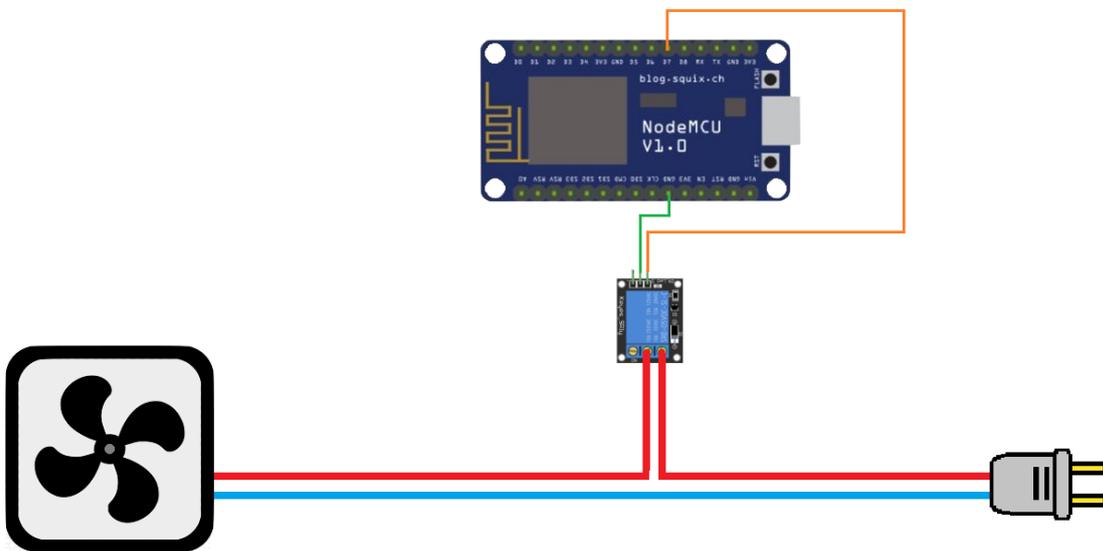
Figura 11 - Sensor Termopar



Fonte: Autoria própria

Os ventiladores e aspersores do local, estão separados nos mesmos quadrantes dos sensores, esses estão ligados cada um a um chamado de Atuador (Figura 6), onde foi embarcado também em NodeMcu um relê e plugs de tomada, para fazer a ligação com os ventiladores e aspersores, onde o relê fará com que passe ou não energia, ligando-os ou desligando-os. Dada temperatura eles são acionados ligando o ventilador daquele quadrante pertencente ao sensor medido, ou são desligados conforme a temperatura ter caído, lidos essa mensagem pelo protocolo MQTT, providos do Sensor. Ao ele ligar ou desligar, este também envia uma mensagem pelo protocolo HTTP para o Servidor, informando o status dele, se ele ligou ou desligou, a cada vez que ele executa essas ações.

Figura 12 - Atuador



Fonte: Autoria própria

Sendo assim o controle é feito a cada quadrante do aviário, trabalhando em par. Quando aquele quadrante/zona do aviário é medido e a temperatura estiver acima da ideal, o Sensor mandará uma informação ao Atuador para que ele ligue os Ventiladores/Aspersores, ou se a temperatura já estiver chego no ideal, O Sensor mandará uma informação que desligue.

Tendo essa rede sem fio de Sensores e Atuadores, totalizam 19 hosts, que comunicam com um roteador Wireless localizado dentro do aviário em preso a uma das paredes, com saída direta para a internet, com um IP fixo liberado na internet. Este sinal é liberado pela infraestrutura de rede da universidade por uma Vlan, que lá na frente se comunicará com o servidor do laboratório, também com IP fixo.

Os Sensores foram programados através da plataforma da *IDE Arduino*, para que o sensor faça as leituras e envie para o IP do Servidor localizado no laboratório. O sensor conecta-se ao *wi-fi*, faz a leitura, encapsula num pacote *HTTP* através de um *GET*, numa *string* contendo todas as leituras efetuadas juntamente a identificação de cada sensor, conforme

o número que ele corresponde ao seu quadrante dividido no aviário, e o envia pelo protocolo, pra uma página PHP do Servidor. No meio da programação o sensor com as leituras de temperatura e umidade analisa através de um cálculo a temperatura ideal e envia uma mensagem via tópico do protocolo MQTT para que o atuador ligue ou desligue, fazendo com que refrigere o local daquele sensor que pertence a aquele quadrante. Cada par Sensor e Atuador se comunicam por MQTT pelo mesmo tópico, sendo que cada Sensor possui o seu respectivo tópico MQTT, ele envia um Pub para o broker MQTT e o Atuador escuta por Sub o mesmo tópico, de acordo com o seu par.

Nas dependências do Laboratório possui um servidor para o gerenciamento, coleta e armazenamento desses dados. Estão instalados neste servidor, Sistema Operacional Linux Debian 8.1 (Jessie) no modo gráfico, serviços de PHP para a linguagem do sistema, Apache como servidor Web e MySQL como servidor de Banco de Dados; há permissão à pasta “www” pro usuário usado na máquina, para que possa criar arquivos e pastas do sistema Web, onde recebe as informações providas dos sensores. Este servidor também provê o serviço de *Gateway* dentro do Laboratório, por meio de uma interface ethernet secundária e conectado um roteador *wireless*, o qual envia e recebe os dados provenientes dos projetos e experimentos executados pelos grupos de pesquisa de dentro do Laboratório, como também navegação a internet para pesquisas.

Neste Servidor há também um sistema Web para coleta e apresentação de dados, onde esses são armazenados em um Banco de Dados para receber esses dados. O sistema contém a página PHP que recebe os dados dos Sensores pelo GET, armazena no banco de dados à sua determinada tabela, junto com a data e a hora. Da mesma forma essa mesma página recebe os dados do acionamento dos Atuadores pelo GET e os armazenam com sua data e hora, no banco de dados à sua determinada tabela, também feito o cálculo da hora para determinar quanto tempo ficou ligado ou desligado e armazenado também na mesma tabela. Ao acessar esse sistema Web através do IP fixo dele, são apresentados em formato de tabelas todas, com a leitura dos Sensores, suas respectivas referências de identificação, data e a hora da leitura.

3. ATIVIDADES DESENVOLVIDAS

Como ao chegar os Atuadores não estavam funcionando por completo, por estarem montando os hardwares e testando o código da programação dos mesmos, não havia um serviço de MQTT no Servidor. Os testes eram feitos em uma plataforma on-line livre chamada “MQTT Cloud”, que é uma plataforma Web onde há instalado o serviço de MQTT, é criado um cadastro gratuito para utilizar o serviço, funcionando de forma que o Sensor envia a mensagem via MQTT por Tópico apontando para o endereço dessa plataforma e o Atuador lê o mesmo Tópico apontado esse mesmo endereço.

Foi aí então que instalei um serviço de MQTT em nosso Servidor, o *Mosquitto MQTT*. Instalado o Mosquitto Server e o Mosquitto-client, assim com o serviço cliente do MQTT é possível fazer o teste se está funcionando a comunicação, utilizando o comando “mosquitto_sub” para ouvir as mensagens que passam pelos tópicos MQTT que chegam nesse servidor, assim como no teste mostrado na Figura 7, onde utiliza esse comando para ouvir as mensagens de todos os tópicos (representado por “#”), e um outro terminal para testar se foi instalado corretamente, dando um comando “mosquitto_pub” para enviar a mensagem ao serviço por um Tópico “tópico_teste”.

Figura 13 - MQTT no Terminal

The image shows two terminal windows. The top window displays the command `sudo mosquitto_sub -h localhost -t "#"` and the output `olá mundo`. The bottom window displays the command `sudo mosquitto_pub -h localhost -t topico_teste -m "olá mundo"` and the prompt `moraes@ltsi:~$`.

Fonte: Autoria própria

Foi também instalado nos celulares Android o App “MQTT Dash”, que consegue montar uma conexão com o Servidor MQTT, dar um Pub ou Sub no Tópico configurado a cada objeto adicionado, como por exemplo adicionamos um objeto nesse “*swith button*” com o desenho de um ventilador, como na Figura 8, para poder ser feito o monitoramento se os Atuadores estavam sendo acionados e poder fazer testes tentando acionar os Atuadores,

para ver se estão funcionando corretamente e ligando os ventiladores, então ao acionar é determinado a mensagem que envia no tópico e ao desacionar também é determinado a mensagem, fazendo com que ligue ou desligue-os. Para facilitar esse monitoramento de dentro do laboratório, sem a necessidade de estar com o celular de todos instalado o App, foi criado uma Máquina Virtual num dos computadores do laboratório, rodando Android com esse App e configurado de acordo com os parâmetros corretos, apontando o Servidor e Tópicos MQTT utilizados.

Figura 14 - MQTT Dash



Fonte: Autoria própria

Assim então foram feitos os testes e monitoramento para ver se todo o sistema estava funcionando corretamente. Foi aí então que alguns Sensores começaram a apresentar problemas de conexão de rede, principalmente os que se encontram no barracão do aviário do lado contrário ao Roteador, que estava instalado em uma das paredes.

Então eu sugeri que colocássemos o roteador preso no centro do barracão e no alto, para que possa ser distribuído melhor o sinal do Wi-fi com menos interferências. Feito dessa forma, prendendo o roteador em uma das hastes do teto do barracão, puxando os cabos de energia e rede para a nova posição, fazendo com que o sinal melhorasse e ficasse melhor distribuído os Sensores e Atuadores.

Porém após continuarmos monitorando, percebemos que ainda alguns Sensores perdiam a conexão com o Servidor. Suspeitado do Servidor que estava caindo o serviço do Apache, até porque ao reiniciá-lo todos os Sensores voltavam a reconectar, mas se reiniciassem a cada Sensor separadamente, também eles voltavam a reconectar normalmente. Para tentar identificar, analisei o Log do Apache instalado no Servidor, e não havia nada indicando que o serviço parou ou reiniciou ou algo do tipo, apenas os acessos ao sistema Web, e as informações de leituras que lhe eram enviadas para armazenar no BD, sendo assim o problema não era com os serviços e instalações do Servidor. Verificado o Roteador, monitorando a lista de hosts conectados a ele pela lista de DHCPs, nisso então

foi constatado que os NodeMCUs é que estavam perdendo a conexão, por algum motivo desconhecido perdiam a conexão com o Roteador em algum momento e ficavam perdidos em encontrar a rota para o Servidor. Então foi resolvido inserindo mais uma instrução na programação dos NodeMCU dos Sensores e Atuadores, para que em determinado tempo, que era o tempo que desconectavam sozinhos, eles reiniciem seu sistema, refazendo a conexão de rede, sendo assim antes de que perdessem a conexão eles reiniciavam.

Após esse acontecimento, voltamos a analisar o Servidor, visto que já havia muitas outras coisas que foram instaladas, arquivos de uso pessoais, e tudo isso junto com o modo gráfico instalado já estava comprometendo a eficiência da máquina. Portanto tiramos e separamos em outra máquina esses arquivos pessoais, para que essa máquina seja apenas servidor mesmo, e então formatada e reinstalada o Sistema Operacional, porém dessa vez em Modo Texto, sem interface gráfica nenhuma, com instalação apenas dos serviços exigidos para o funcionamento do Sistema Autônomo do Aviário e demais projetos, como: Mosquitto/MQTT, Apache com os arquivos do sistema Web já feito antes, onde tínhamos feito backup, PHP, MySQL.

Passado mais um tempo, como o hardware desse Servidor já estava velho e tinha usado bastante para outros fundamentos passados, sua placa mãe começou a apresentar superaquecimento nos Chipsets Norte e Sul, fazendo com que a máquina travasse, precisando resetá-la, mas mesmo após isso essa máquina começou a travar de 5 em 5 min, tendo que desligar e religar todo esse tempo. Mesmo os Chipsets contendo dissipadores de calor eles chegavam a ferver, tentamos ar uma manutenção geral, desmontando toda ela, limpando, adicionando coolers em cima desses dissipadores dos chipsets, porém não surgiram efeito, a máquina continuou esquentando a ponto de nem ligar, ou quando ligava ela esquentava e viria a travar novamente.

Nesse caso precisávamos de um novo Servidor, porém não tínhamos outra máquina e até conseguir um novo equipamento levaria muito tempo. Mas o laboratório tinha um Domínio com Servidor na Nuvem, para outros projetos com um site rodando, onde já tinha os serviços de PHP, Apache e MySQL rodando. Então foram transferidos todo o sistema Web do Aviário para esse Servidor na Nuvem, transferindo todos os arquivos que já haviam feito backup anteriormente, e o banco de dados salvo no tempo que o servidor antigo conseguia ficar ligado, e migrado também para esse Servidor na Nuvem, o MQTT voltado a utilizar a plataforma livre, Eclipse, que estava sendo usado anteriormente, até acharmos uma nova solução e não deixar nessa plataforma, por se tratar de uma plataforma apenas para testes. Alterado também os Sensores e Atuadores para onde deverão enviar suas leituras, mudando o IP de destino, como também o IP do Broker MQTT para essa Plataforma Eclipse.

Sendo assim nesse servidor ficariam até mais seguros, por ter uma empresa por trás se dedicando exclusivamente pela saúde do servidor e sua Segurança. Porém após um tempo recebendo dados dos Sensores, são essas diretivas de Segurança do Servidor na Nuvem, que acabou bloqueando as próprias leituras dos Sensores, bloqueando pacotes vindos deste determinado IP, que era da nossa rede de Sensores. O Servidor recebia requisições HTTP com as leituras a cada 5 min, de 8 Sensores com DHT, mais 3 Sensores com Termopar e mais 8 Atuadores respondendo seus status, recebendo então 19 requisições de rede a cada 3 min, sendo assim o Firewall desse Servidor na Nuvem identificava como se fosse um ataque de Dos, então pra evitar que chegasse mais

requisições que poderiam derrubar suas conexões, ele começou a descartar os pacotes e com o continuou envio vindo de um único IP, ele bloqueou pacotes com origem deste IP, fazendo com que não receba mais as leituras que precisávamos.

Ao discutirmos uma solução junto ao desenvolvedor do Sistema Web, foi então pensado em concentrar todas as leituras em ponto, para enviar tudo de uma vez em uma única requisição com um espaço de tempo maior para enviar a próxima requisição. Mas para isso precisaríamos utilizar algo para concentrar esses dados, algum concentrador.

Com essa necessidade de um novo hardware para concentrar pequena quantidade de dados para depois enviar ao Servidor na Nuvem, foi utilizado um *Raspberry Pi 3*. Onde instalei o Sistema Operacional mais usado e leve pra ele, o Raspbian, em modo texto, com os Serviços do Apache, MySQL, PHP e PhpMyAdmin instalados, com também o serviço de FTP, o VSFTPD, assim o desenvolvedor possa enviar sua nova aplicação para esse Raspberry Pi. Foi também instalado o serviço Mosquitto/MQTT, para que os Sensores e Atuadores possam se comunicar, deixando de usar novamente aquela aplicação free, Eclipse, utilizando recursos que temos de dentro do laboratório mesmo.

Ficando então esse Raspberry Pi ligado apenas na tomada e um cabo de rede, o mesmo que dava conexão direta com IP fixo para o outro Servidor que tinha parado de funcionar, utilizando o mesmo número de IP, que foi liberado na infraestrutura da universidade.

Sendo assim, os Sensores ficaram de fazer a leitura e enviá-las para Raspberry Pi, onde com uma nova aplicação é recebida e armazenado em uma linha de um arquivo TXT, ficando as informações distribuídas na linha como eram no banco de dados. Os Sensores enviam as leituras, ficando um arquivo TXT com várias linhas, uma com cada leitura, e armazenando mais de uma leitura as vezes do mesmo Sensor, por conter um tempo mais de envio. Dado um tempo, a aplicação envia esse TXT do Raspberry Pi para o Servidor na Nuvem, onde lá é lido as informações do TXT e armazenados no Banco de Dados normalmente, de acordo com os seus campos certos, e esse arquivo TXT enviado é apagado do Raspberry Pi para não tomar espaço na memória, durante esse tempo de envio do arquivo e confirmação são criados outros arquivos TXT, então após o tempo determinado a aplicação começa gerar outro arquivo TXT, mesmo que o primeiro não tenha sido enviado, e só exclui se o Servidor na Nuvem tenha recebido, retornando uma mensagem de ok. Sendo assim o Servidor da Nuvem tem apenas uma requisição, que seria esse arquivo TXT, e em um espaço maior de tempo, não bloqueando mais as conexões providas desse IP. Já que irá ficar o tempo todo ligado, foi providenciado um Cooler grande de gabinete, colocado bem em cima do Raspberry Pi, refrigerando-o o suficiente para se manter o tempo todo ligado, e esse sistema de refrigeração será pensado em melhorar no futuro para ser mais prático.

Tendo em vista com esses testes de Servidor com Raspberry Pi, demos continuidade numa nova possibilidade do projeto, onde poderá utilizar esse mesmo sistema autônomo e monitoramento, em Aviários onde não chegam a internet. Então seria montada uma rede de Sensores e Atuadores, interna dentro apenas do aviário, utilizando os NodeMCUs com Wi-fi, um roteador Wi-fi e um Raspberry Pi como Servidor compacto. Sendo assim com os mesmos fundamentos e programações dos Sensores, porém com um Servidor e Broker MQTT na rede interna, onde este Servidor Raspberry Pi estaria instalado os Serviços

necessários, com uma aplicação mais simples e leve, que receba as leituras e armazene no banco de dados, e quando for necessário analisar os dados, acessado remotamente de um notebook ou colocar um monitor ou display gráfico para visualização sem necessidade de levar um outro equipamento até lá, e então exportar os dados SQL em planilhas do Excel, podendo exportar para um pendrive, ou até mesmo no notebook, para aí sim levar ao laboratório e fazer as análises necessárias dos resultados das leituras.

Criado então esse novo Servidor em Raspberry Pi para aviários sem internet. Como tinha a ideia de utilizar um display gráfico até o momento, foi então instalado o Raspbian no modo gráfico, com os principais serviços, como o anterior, MySQL, PHP, Apache, Mosquitto e PhpMyAdmin. Liberado também um usuário para acessar via SSH, para futuras manutenções e configurações, e também o serviço VSFTPD para transferência de arquivos via FTP, para que desenvolvedores possam atualizar arquivos do sistema Web. Para ser feita essa atualização, foi criado um grupo de usuários “desenvolvedores”, com um usuário para o desenvolvedor, para cada desenvolvedor que tiver irá ter seu usuário e adicionado a esse grupo, esses usuários deste grupo tem acesso para fazer alterações nos arquivos da pasta WWW, através de sua pasta Home, que na pasta “/home” de cada usuário terá um link direto redirecionando a pasta “/var/www” do sistema, até pq esse usuário acessando via FTP não conseguirá sair de sua pasta home, somente conseguirá ir para a www e através desse link. A montagem deste dentro do aviário sem internet, com sua refrigeração, ficou de ser montada para o próximo semestre ao retorno das férias.

3.1. Possibilidades de Melhoria na Segurança

Em meio a toda essa infraestrutura, percebi que há possíveis falhas de segurança, para que pessoas mal-intencionadas possam utilizar recursos da rede, enviar dados incorretos dos Sensores, acessar a informação indevidamente que se está sendo enviada, acionamento indevido dos Atuadores, portanto podem ser dotadas algumas medidas de segurança em cada equipamento.

Roteador: o roteador do aviário pode conter umas configurações de segurança, evitando o acesso indevido a rede de Sensores, podendo roubar banda de internet, ou acessar as informações trafegadas. No momento temos algumas configurações básicas que poderemos manter, que seria a segurança em WPA2 com senha e desativar recurso de conexões WEP, mas algumas configurações que poderiam aumentar a segurança, seria: alterar senha de administrador das configurações do roteador, evitando que alguém acesse as configurações pela senha padrão e altere as configurações, liberando outros acessos; esconder o SSID do Wi-fi, para dificultar a conexão de possíveis pessoas mal intencionadas; bloquear conexão de mais hosts, permitindo apenas os MACs dos NodeMCUs dos Sensores e Atuadores, impedindo que, por mais que consigam acessar a senha do wi-fi, não consigam se conectar à rede, por o roteador não permitir conexões de outros hosts.

Sensor: como o protocolo MQTT há possibilidade de se autenticar com usuário e senha para tráfego de dados com seu Broker, poderia ativar essa opção, onde o Sensor iria se autenticar no serviço do Mosquitto no servidor, com a autorização aí sim ele enviaria as informações por MQTT. Sendo assim isso evitaria que caso invadam a rede, para enviar informações pelo MQTT, além de saberem o qual tópico está sendo utilizado precisariam saber também o usuário e senha do Mosquitto no Broker, evitando assim que qualquer um tenha controle de enviar mensagem para o Atuador ligar ou desligar em horas indevidas.

Também precisaria de uma criptografia para o tráfego desses dados, para trafegar esses dados por MQTT e HTTP criptografado. Foi tentado desdobrar esse projeto para tentar criptografar esses dados, porém não foi encontrado nada que possa fazer trafegar informações em HTTPS pelo NodeMCU, aparentemente este micro controlador usado não suporta esse tipo de recursos, por limite de processamento. Mas então poderia ser desenvolvido algum código que criptografe as leituras e mensagem ao Atuador, e envie para o servidor criptografados, sendo assim caso alguém consiga capturar o pacote, não entenderá do que se trata. Esta ideia foi tentada desenvolver também por mim, mas até o momento, por questões de tempo, não foi possível desenvolver em tempo hábil.

Uma outra medida a tomar com a segurança das informações que enviam para o Servidor, seria o Sensor tem um número de identificação, para poder sim se autenticar no Servidor, se bem que nos dados enviados para armazenamento no Banco de Dados, os Sensores já possuem um ID, como por exemplo “DHT010100102”, então poderia utilizar este ID mesmo para ser feita essa autenticação (que irei detalhar mais na parte do servidor).

Atuador: assim como no Sensor, para receber a mensagem via MQTT de ligar ou desligar. O Atuador primeiro se autenticaria no serviço do Mosquitto no Broker, para depois receber a mensagem e executar a ação. Antes ainda também trabalharia com criptografia, onde ele buscaria a mensagem da ação a se tomar, que está criptografada, então ele descriptografaria para saber o que deverá ser feito. Como ele também envia informações de status, que horas que ligou ou desligou, quando tempo ficou ligado, via HTTP como o Sensor, essas informações também seriam criptografadas do mesmo jeito, para então enviar essas informações ao Servidor Web e também lá ele seria autenticado pelo seu ID, que já possui também, como “ATU010100102”.

Servidor: na parte do MQTT, ele sendo o Broker, ativaria a *autenticação* pelo protocolo, onde para poder utilizar esse servidor, para se dar um PUB ou SUB, o host antes deve se autenticar a esse serviço, ou seja pra enviar alguma mensagem num Tópico, ou ler a mensagem desse Tópico, precisaria antes se autenticar no Broker do Mosquitto, para depois liberar esse acesso. A mensagem criptografada que o Sensor enviaria para o Atuador ouvir, continuaria criptografada, o Mosquitto apenas iria gerenciar o serviço de transmissão, não iria fazer nada quanto a criptografia.

As mensagens que chegassem pelo protocolo HTTP, que seriam as leituras, essas sim seriam descriptografadas. A mensagem chegaria com as leituras criptografadas, então o sistema PHP iria separar a mensagem e descriptografaria as leituras e armazenariam no banco de dados a informação clara. No caso de como descrito anteriormente, que no fim foi utilizado um Raspberry Pi como concentrador que transmite para um outro Servidor na Nuvem, os dados poderiam ser armazenados criptografados no arquivo TXT, no mesmo formato e distribuição, deixando assim para o Servidor da Nuvem quando abrir o TXT, *descriptografar* os dados para assim armazenar no banco de dados em texto claro. No caso também, já que seriam dois Servidores se comunicando, nesse caso esse arquivo poderia ser transferido via HTTPS, até porque nos dois, tanto o Raspberry Pi quanto o Servidor na Nuvem possuem o Apache, que suporta o protocolo HTTPS.

Também esse sistema iria antes de tratar essa mensagem, fazer uma *validação* se de fato a mensagem veio de algum Sensor conhecido no meio desse projeto, caso não

conseguisse identificar o sistema ignoraria a mensagem, evitando que armazene no banco de dados uma mensagem falsa, que seria de alguma pessoa ou host mal-intencionado.

Para proteger o Raspberry Pi, poderia também ter um serviço de *Firewall*, bloqueando todas as portas e liberando apenas as necessárias pelo seu uso. No caso ICMP para poder se fazer teste de conexão, mas com limites de “ping”, para evitar que o trave, derrubando todos os serviços com o “ping da morte”; portas de DNS para funcionar a conexão de internet; a porta do MQTT, para poder se fazer a comunicação entre o Sensor e Atuador; a porta de HTTP e HTTPS, para poder se comunicar com o meio externo da internet, receber as leituras dos Sensores e Status dos Atuadores, enviar os arquivos TXT para o Servidor na Nuvem.

Com dados sendo gravados constantemente no Banco de Dados, mesmo que no servidor na Nuvem tenha algum plano, que até então não sei se cobre backup, por eu não ter tido esse acesso, é bom que tenhamos um método de backup nosso, para que caso aconteça algo, tenhamos já de imediato um backup para corrigirmos mais rápido e subir os dados. Como dito, não tive acesso direto a configurações desse servidor e não sei o que pode ser feito, então até o momento, teria que ser feito um backup manualmente pelo, semanalmente, agora se fosse um servidor nosso, com a máquina física dentro do laboratório, como era anteriormente, poderia inserir uma instrução agendada dentro do Linux, para executar um script que efetua backup do Banco de Dados por comandos do MySQL. No Raspberry Pi, como há apenas arquivos TXT, que enviam esses arquivos pra Nuvem e eles são apagados, há ainda a possibilidade de caso fique sem internet o local por algum motivo, e então esse arquivo não seja enviado acumulando diversos TXTs, então nesse tempo haja também problemas com o cartão de memória do Raspberry Pi ou com ele próprio, com isso perder esses arquivos que ainda não foram enviados, então poderia ter uma mídia externa, como por exemplo um pendrive, que fique plugado sempre no Raspberry, sincronizando a pasta onde são criados os arquivos TXTs com uma outra pasta localizada no Pendrive, assim ao se criar um arquivo TXT já será automaticamente sincronizado com a pasta do pendrive criando uma cópia, caso numa instabilidade de conexão e posteriormente problemas no cartão, ainda terá os arquivos TXT no pendrive para poder enviar os dados ao Servidor na Nuvem. As aplicações Web como não sobrem atualizações frequentemente, só seriam feitos backup quando essas sofrerem alguma alteração.

4. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Tendo em vista do explicado sobre o projeto, verificado que é possível uma rede Sensores se comunicar com outra rede externa, enviando dados a um sistema Web para mensurar e analisar o comportamento do ambiente.

Visto também que se é possível desenvolver um Servidor Web, recebendo dados de Sensores também, utilizando um Raspberry Pi 3, onde seus recursos são suficientes para atender uma rede desse porte, com o devido refrigeração.

De acordo com um dos problemas relatados, como uma possibilidade de o firewall identificar um ataque de Dos, mas que esses inúmeros acessos consecutivos eram necessários, visto que é possível também prover uma solução com um concentrador de dados, não precisando diminuir a Segurança do Servidor por conta da necessidade e com isso abrindo brechas para reais ataques.

Com a análise e visão de Segurança adquirida pelo Curso de Segurança da Informação, consegui apontar onde precisaria de melhorias e o que poderia ser feito para executar essas melhorias, para tornar-se um sistema um pouco mais seguro. Podendo ser implementada numa futura continuação do projeto.

Poderão também num futuro, ser aplicadas as ideias de Segurança apresentadas anteriormente, com talvez possíveis melhorias, ou alterações de recursos para a utilizações melhores de Segurança, como por exemplo alguma placa de sensores que suporte o HTTPS facilmente.

REFERÊNCIAS BIBLIOGRÁFICAS:

AOSONG ELECTRONICS CO.LTD. **DATA SHEET: Digital-output relative humidity & temperature sensor/module DHT22**, 2013.

CARVALHO, T; SOUZA, T. Internet das Coisas e Sua Aplicação em Bibliotecas. **Revista Gestão.Org**, v. 13, Edição Especial, 2015. p. 264-270 ISSN 1679-1827

MARTINS, J; VIANNA, A. Análise sobre experimentos com potenciômetros para a introdução do uso de sensores em cursos de física. **Universidade Tecnológica Federal do Paraná – UTFPR - Programa de Pós-Graduação em Ensino de Ciência e Tecnologia – PPGET - II Simpósio Nacional de Ensino de Ciência e Tecnologia**. Artigo número: 61 ISSN 2178-6135

MENDES, J; STEVAN, S. LDR E SENSORES DE LUZ AMBIENTE: FUNCIONAMENTO E APLICAÇÕES. **Universidade Tecnológica Federal do Paraná – UTFPR - Semana de Eletrônica e Automação**. SEA 2013

International Business Machines Corporation (IBM) Eurotech. **MQTT V3.1 Protocol Specification**. Disponível em < <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html> >. Acessado em 10 de Fevereiro de 2019.

MOREIRA, L. Medição de Temperatura Usando-se Termopar. **Revista Cerâmica Industrial**, vol.7 n5, 2002

OLIVEIRA, M; OLIVEIRA, R; SOUZA, M; HARADA, E; TECH, A. Desenvolvimento de Sensores para Monitoramento de Ambiente Aviário com Ênfase em Controle Térmico. *IN: Brazilian Journal of Biosystems Engineering* v. 12(3): 234-240, 2018.

Romano, V. F. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos**. Edgard Blucher, 2002.

SENSE SENSORS & INSTRUMENTS. Medição de Temperatura em Processos Industriais. **Variáveis de Processo Temperatura**. 2000000300 Rev. A – 09/2015. Acessado em 09 de Fevereiro de 2019.

SOUZA, A; CARVALHO, P. Utilização de Sensores no ensino das ciências. **FCUP - Artigo em Revista Científica Nacional** Sala de professores/alunos vol. 34 nº 3/4, 2011 <<https://repositorio-aberto.up.pt/handle/10216/81262>>. Acessado em 06 de Fevereiro de 2019.

SOUZA, N. S. **Apostila de acionamentos elétricos**. Material do Professor. Instituto Federal de Educação, Ciência e Tecnológica do Rio Grande do Norte.2009. Disponível em < <https://docente.ifrn.edu.br/heliopinheiro/Disciplinas/maquinas-e-acionamentos-eletricosii/apostila-basica> >. Acessado em 10 de fevereiro de 2019.