

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
BACHARELADO EM ENGENHARIA DE TELECOMUNICAÇÕES

JOÃO PEDRO MENEGALI SALVAN BITENCOURT

SISTEMA DE MONITORAMENTO E AVISO DE EQUIPAMENTOS LIGADOS SEM USO

SÃO JOSÉ, 2023

SISTEMA DE MONITORAMENTO E AVISO DE EQUIPAMENTOS LIGADOS SEM USO

Pré-projeto integrador 2 apresentado ao Curso de Bacharelado em Engenharia de Telecomunicações do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como requisito à obtenção de nota parcial na disciplina de Comunicação e Expressão.

Professor orientador: Arliones Stevert Hoeller Junior

SÃO JOSÉ, 2023

RESUMO

Este relatório objetiva analisar o desenvolvimento de um sistema de monitoramento de equipamentos em ambientes fechados, como salas de aula, escritórios, entre outros, com a finalidade de prevenir que os mesmos sejam esquecidos ligados desnecessariamente e, como consequência, ocasionem desperdício de energia e sofram maior desgaste. Nos testes realizados, os locais monitorados foram as salas de aula do Instituto Federal de Santa Catarina do câmpus São José. Para realizar o monitoramento, foi concebido um equipamento que conta com sensores de luminosidade, temperatura, umidade, presença e infravermelho. Além disso, foi desenvolvido um sistema que captura os dados dos sensores citados e os consolida de forma a determinar se algum equipamento foi esquecido ligado, além de oferecer ao operador uma interface *web*, a qual disponibiliza os dados consolidados, permitindo que as ações necessárias sejam tomadas, de modo que o tempo gasto com a verificação manual, indo de sala em sala, pode ser utilizado em outras funções. Além da função principal, a plataforma on-line conta com gerenciamento de usuários, de tal maneira que é possível arranjar vários operadores simultaneamente, cada um com seus dispositivos de monitoramento.

Palavras-chave: Monitoramento. Sistema. Economia.

SUMÁRIO

1 INTRODUÇÃO.....	5
1.1 OBJETIVO GERAL.....	6
1.2 OBJETIVOS ESPECÍFICOS.....	6
2 REFERENCIAL TEÓRICO.....	7
2.1 INTERNET DAS COISAS.....	7
2.2 PLATAFORMA ESP32.....	7
2.3 MQTT.....	8
2.4 JAVA.....	8
2.5 LDR.....	9
2.6 SENSOR DHT11.....	9
3 METODOLOGIA.....	9
3.1 DESENVOLVIMENTO NA PLATAFORMA ESP32.....	9
3.2 MQTT E ANJO.....	10
3.3 O SISTEMA.....	12
4 RESULTADOS E DISCUSSÃO.....	14
5 CONSIDERAÇÕES FINAIS.....	15
REFERÊNCIAS.....	16

1 INTRODUÇÃO

Ao analisar-se alguns aspectos do Instituto Federal de Santa Catarina, câmpus São José, foi observado que, após o uso, vários equipamentos acabam sendo esquecidos ligados, consumindo energia desnecessariamente. Dessa forma, foi planejado um sistema que constitui em um equipamento que ficará na sala monitorada e um serviço em nuvem que receberá os dados dos sensores desse equipamento para avaliar se algum dispositivo ficou ligado de forma desnecessária.

O equipamento físico, chamado de AnJo, ficará presente na sala monitorada e contará com sensores de temperatura, luminosidade, umidade e presença, conexão sem fio através da tecnologia Wi-Fi, e um LED infravermelho. Ao detectar movimento, o AnJo começará a capturar os dados dos sensores e os enviará a um servidor. Tanto o dispositivo monitorador quanto o servidor, farão a consolidação dos dados. O aparelho de ar condicionado será monitorado através das variações de temperatura. Já a luminosidade será monitorada através do sensor que identifica a intensidade luminosa. O LED infravermelho permitirá dar comandos remotamente para o ar-condicionado. Os dados são transmitidos através da conexão de rede sem fio.

O usuário do sistema poderá gerenciar os monitoradores através uma página *web*, que conterà as informações de cada um, como o local onde estão posicionados, se estão atualmente conectados no servidor, visualizar os dados dos sensores, visualizar os dados consolidados e compará-los com os dados consolidados do servidor. A consolidação dos dados estipulará se algum equipamento ficou ligado sem que o ambiente monitorado esteja em uso. O critério para determinar se o local está em uso é a detecção de movimento. Caso o sensor não identifique movimento após um determinado tempo, através dos dados capturados dos outros sensores, tanto o equipamento monitorador, quanto o servidor, farão a consolidação dos dados, que alertará sobre a possibilidade de equipamentos estarem ligados e sem uso do ambiente. Diante da análise, o operador do sistema poderá enviar um comando pré-cadastrado para desligamento do ar-condicionado.

Como todo o sistema funciona de forma não invasiva, monitorando apenas as condições do ambiente, usa-se certos padrões para deduzir se algum equipamento ficou ligado. Dessa forma, é possível utilizar a solução em ambientes genéricos.

1.1 OBJETIVO GERAL

Impedir que equipamentos fiquem ligados desnecessariamente, sinalizando mais rapidamente os responsáveis por desligá-los, ou ainda, realizar o desligamento remoto dos mesmos.

1.2 OBJETIVOS ESPECÍFICOS

- Analisar o padrão de variação de temperatura após o ar-condicionado permanecer ligado de forma a estabilizar a mesma.
- Analisar o padrão de variação de temperatura de acordo com a distância do sensor para com o equipamento.
- Analisar o padrão de inércia da variação de temperatura ao desligar o ar-condicionado.
- Avaliar o alcance do sensor infravermelho.
- Avaliar os modelos de ar-condicionado que possuam o comando para desligamento diferente do comando para ligá-lo.
- Determinar as condições iniciais o qual o equipamento monitorador deve ser ligado.
- Determinar a sensibilidade do sensor de luminosidade.

2 REFERENCIAL TEÓRICO

2.1 INTERNET DAS COISAS

Segundo Oliveira, Assis e Nolli (2018, p. 52),

Com os avanços da tecnologia, sistemas embarcados, microeletrônica, comunicação e sensoriamento, surge-se um novo conceito: a internet das Coisas (IoT - Internet of Things). Atualmente, ela é uma das principais novidades tanto na academia quanto na indústria devido a sua potencialidade de aplicação nas mais diversas áreas das atividades humanas.

Esse tipo de tecnologia permite maior presença de equipamentos cujos objetivos são diversos, sendo um deles o de monitoramento através do uso de sensores.

2.2 PLATAFORMA ESP32

Uma das plataformas utilizada nesse cenário para conceber o dispositivo de monitoramento chamado “AnJo” é a “ESP32-WROOM-32”, presente no *chip* “ESP32-D0WDQ6”. Segundo (ROBOCORE; on-line), as especificações técnicas da placa são:

- Processador: Xtensa® Dual-Core 32-bit LX6
- Memória *Flash* programável: 4 MB
- Memória RAM: 520 KBytes
- Memória ROM: 448 KBytes
- *Clock* máximo: 240 MHz
- Pinos Digitais GPIO: 25 (todos com PWM)
- Resolução do PWM: até 16 bits (ajustável via código)
- *Wireless* 802.11 b/g/n - 2.4GHz (antena integrada)
- Modos de operação: *Access Point* / Estação / *Access Point* + Estação
- *Bluetooth Low Energy* padrão 4.2 integrado
- Tensão de alimentação externa: 4,5 V a 9 V (o módulo possui regulador integrado para 3,3 V).

Foi utilizado o conjunto “IoT DevKit - LoRaWAN”, fornecido pela “ROBOCORE”, que conta com os sensores de temperatura, umidade, luminosidade, além de um botão que foi utilizado para simular a presença de movimento na sala monitorada.

Para a programação da placa, foi utilizado o conjunto de ferramentas do *PlatformIO*, que é uma plataforma profissional, disponível para vários sistemas operacionais, suportando várias arquiteturas de processador, múltiplas estruturas de trabalho, criado para desenvolvedores de *software* que escrevem aplicações direcionadas à dispositivos embarcados (PLATFORMIO; on-line).

As bibliotecas e o programa que é executado no monitorador foram escritos na linguagem de programação C++. O programa, faz a leitura dos sensores, e os envia para um servidor através do protocolo MQTT.

2.3 MQTT

É um protocolo concebido para a transferência de mensagens, que usa o modelo de publicação e inscrição, em inglês, *publish and subscribe*. No MQTT, que necessita de um *broker* central, a mensagem é publicada em um tópico e o cliente deve estar inscrito naquele tópico para ver a mensagem (STEPHEN, 2021). Para Santos (2022; on-line), o *broker* é um gerenciador de mensagens, que atua como intermediador entre o assinante e o publicador, já que não conexão direta entre ambos.

No caso do sistema em discussão, o dispositivo AnJo é o publicador e o servidor desenvolvido para esse projeto será o assinante. Um servidor extra será o *broker* que intermedeia essa comunicação. Para tal, foi utilizado o “Eclipse Mosquitto”, que é um *broker* de código aberto, que implementa o protocolo MQTT na versão 5.0, 3.1.1 e 3.1 (MOSQUITTO, 2023; on-line).

2.4 JAVA

É uma linguagem de programação e computação lançada pela Sun Microsystems, em 1995 (JAVA; on-line). Para maior facilidade na construção do sistema, foi utilizada a plataforma JAVA *Gradle*, que é uma ferramenta de código aberto focada em automação, flexibilidade e desempenho (GRADLE; on-line).

Para compor todo sistema, juntamente às duas ferramentas citadas anteriormente, foi utilizado o *Spring Boot*. Este, torna fácil criar aplicações independentes que podem ser executadas como serviço (BOOT; on-line). Toda a lógica de programação que dita como o servidor desenvolvido opera, bem como a interface que é utilizada para o usuário, estão contidas em uma única aplicação gerada com o *Spring Boot*.

2.5 LDR

LDR, sigla em inglês para *Light Dependent Resistor*, que traduzido significa resistor dependente de luz, também é conhecido como fotoresistor e é um tipo de resistor que tem a capacidade de variar a sua resistência em função da intensidade de luz que incide sobre ele (ALVES; on-line). Quando as partículas de luz (fótons) incidem sobre a superfície do sensor, os elétrons que estão no material semicondutor são liberados, dessa forma a condutividade do LDR aumenta e a sua resistência diminui (ALVES; on-line).

2.6 SENSOR DHT11

Segundo ROBOCORE (2023), é um sensor de temperatura e umidade que produz um sinal digital de 8 bits.

3 METODOLOGIA

Para elaborar o sistema, foram utilizadas tecnologias vistas ao longo do curso de Engenharia de Telecomunicações, como as linguagens de programação C++ e Java, e um sistema embarcado na plataforma ESP32.

3.1 DESENVOLVIMENTO NA PLATAFORMA ESP32

Para a utilização dos sensores, foi criado um código-fonte próprio implementando o uso de cada sensor. Dessa forma, proporcionou-se maior controle no tratamento dos dados

obtidos. Todo o desenvolvimento na placa ESP32 utiliza a linguagem de programação C++, que, segundo Stroustrup (1986; p. 4), provê suporte para abstrair dados, de modo que o programador pode definir tipos personalizados e utilizá-los como se fossem nativos.

Para cada sensor, foi escrita uma biblioteca, em formato de classes. Além disso, foi escrita uma biblioteca, na mesma linguagem, para todos os cálculos referentes ao padrão de variação de temperatura ocasionado pelo ar-condicionado, com o objetivo de determinar se o mesmo encontra-se ligado ou não. Abaixo, estão as bibliotecas criadas:

- AC.h: utilizada para a lógica que determina se o ar-condicionado está ligado, baseando-se no padrão de variação de temperatura ao longo do tempo;
- AnjofiMqtt.h: utilizada para o envio de dados coletados dos sensores para o *broker*;
- AnjofiWifi.h: utilizada para realizar a conexão com a Internet, através de uma rede sem fio;
- LDR.h: utilizada para a lógica de cálculo da luminosidade;
- TemperatureHumidity.h: utilizada para a lógica de cálculo da temperatura e umidade;

A placa conecta-se à Internet através da rede sem fio Wi-Fi, que permite o envio dos dados dos sensores e dados consolidados na própria placa para o servidor *broker*, o qual receberá os mesmos através do protocolo MQTT.

3.2 MQTT E ANJO

O programa gravado no AnJo gera um dado consolidado que é enviado através do protocolo MQTT para um servidor cadastrado. Abaixo, está um exemplo de um dos envios:

Quadro 1
Dado, no formato JSON, gerado e enviado via MQTT

```
1 {  
2     temperature:16.274000,  
3     humidity:65.000000,  
4     acStatus:false,  
5     lightStatus:true,  
6     lightCurrentValue:2386,  
7     lightBaseValue:2783  
8 }
```

No Quadro 1, as informações estão divididas em seis categorias:

- *temperature*: contém a informação da temperatura atual do ambiente monitorado. No exemplo, o valor de temperatura é de 16,27 graus Celsius;
- *humidity*: contém a informação relacionada à umidade do ambiente monitorado. No exemplo, o valor de umidade é de 65%;
- *acStatus*: indica se o ar-condicionado está ligado ou não. O valor gerado é do tipo *booleano*, ou seja, assume verdadeiro caso o aparelho esteja ligado e falso, caso contrário;
- *lightStatus*: indica se a iluminação está ligada ou não. Assim como *acStatus*, o valor gerado em *lightStatus* também é *booleano* e assume verdadeiro quando a luminosidade está acima do parâmetro de base e falso, caso contrário;
- *lightCurrentValue*: indica o valor atualizado de luminosidade, que é menor quando a luminosidade está alta e maior, caso contrário. No exemplo, o valor é 2386. O maior valor que esse parâmetro assume é 4095;
- *lightBaseValue*: indica o valor base de iluminação. É utilizado para a calibragem inicial do sistema, de forma que valores acima dessa calibragem indicam que a luminosidade está ligada e abaixo indicam que a mesma está desligada.

Para facilitar a interpretação, por parte do servidor JAVA desenvolvido, dos dados gerado pelo dispositivo monitorador, o formato de dados escolhido foi o JSON. Segundo Pezoa (2016, p. 1), o JSON desempenha um papel importante em aplicações *web*, principalmente no uso de APIs, sigla inglesa para *Application Programming Interface*, dado que é uma linguagem facilmente compreendida por desenvolvedores e máquinas, tornando-se o formato mais popular para enviar requisições e respostas através do protocolo HTTP.

Os dados dos sensores começam a ser obtidos e enviados assim que o monitorador detectar movimento no ambiente, e permanece nesse estado após um certo tempo sem haver movimentação. Tal comportamento após o fim da presença de movimento no ambiente é necessário para acompanhar a variação de temperatura, que é consolidada para deduzir se o ar-condicionado está ligado ou não. Com isso, o dispositivo foi projetado para ser completamente não invasivo, tentando detectar as condições do ambiente através de seus sensores e sem interação direta com o objeto de observação.

3.3 O SISTEMA

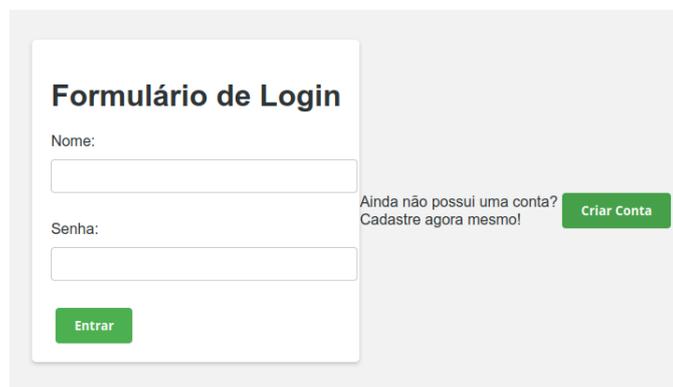
Coleta e consolida os dados, de forma a serem exibidos de forma mais clara ao usuário. Os obtidos via protocolo MQTT são processados de acordo com cada tipo. Aqueles referentes à luminosidade, consideram o valor de intensidade luminosa inicial, que deve ser obtido com as luzes acesas, ou seja, o AnJo deve colocado em operação com as luzes do ambiente já ligadas para que se realize a calibragem inicial. Dessa forma, caso o valor de luminosidade reduza abaixo de 30% do valor inicial, considera-se que a luz está apagada. Essa tolerância é adicionada para possíveis pequenas variações no valor lido, já que a medição é realizada através de um conversor de sinal analógico para digital, sujeito a interferências externas, como picos na rede elétrica.

Os dados referentes à temperatura são processados ao longo do tempo para determinar se o ar-condicionado está ligado ou não. É considerado que o ar-condicionado pode tanto aumentar quanto diminuir a temperatura de um ambiente.

A janela de medição ocorre entre o início da detecção de movimento, até a detecção de pequenas variações de temperatura. O objetivo é apenas sinalizar o usuário caso o dado consolidado indique que um equipamento ficou ligado sem uso.

Como a interação com o usuário é on-line, o mesmo pode acessar através de várias plataformas, como dispositivos móveis, computadores, entre outros. O cliente pode acessar o sistema através de um formulário de acesso, conforme mostrado abaixo:

Figura 1 - Página de acesso ao sistema, a qual o usuário insere as credenciais cadastradas.



A imagem mostra uma interface web para login. No topo, há o título "Formulário de Login". Abaixo dele, há dois campos de entrada: "Nome:" e "Senha:". Cada campo tem um botão "Entrar" verde ao lado. À direita dos campos, há um texto que diz "Ainda não possui uma conta? Cadastre agora mesmo!" e um botão "Criar Conta" verde.

Fonte: autoria própria.

Caso o usuário não possua acesso ao sistema, o mesmo pode se cadastrar, conforme mostrado abaixo:

Figura 2 - Página de cadastro do novo usuário.



O formulário de cadastro, intitulado "Cadastro", apresenta um fundo escuro com elementos em branco. No topo, o título "Cadastro" é exibido em uma fonte grande e bold. Abaixo dele, há quatro campos de entrada de texto, cada um com um rótulo cinza claro: "ID", "Nome", "Email" e "Senha". Os campos são retangulares com cantos arredondados. Na base do formulário, um botão cinza claro com o texto "Cadastrar" em negrito está centralizado.

Fonte: autoria própria.

Na Fig. 2, o campo *ID* refere-se ao nome de usuário único que será criado, ou seja, não é possível a existência de dois ou mais usuário com mesma informação de *ID*. As outras informações, como Nome, E-mail e Senha, podem ser repetidas entre os utilizadores.

4 RESULTADOS E DISCUSSÃO

Dados os objetivos firmados inicialmente, o equipamento monitorador foi desenvolvido e capaz de realizar as medições, além de identificar se a iluminação estava acesa ou não. Além disso, foi possível extrair os dados do AnJo e processá-los em um sistema externo, que pode ser acessível através de uma interface *web*.

Com relação a lógica para determinar se o ar-condicionado está ligado ou não, baseando-se no padrão de variação de temperatura, foi observado que os resultados não são precisos por conta do sensor DHT11 possuir tal imprecisão. Para diminuir o problema, na biblioteca "TemperatureHumidity.h" criada, foi aplicado um fator de correção, tendo como referência um termômetro de mercúrio vermelho. Até a finalização deste relatório, a lógica não estava pronta para que uma nova análise fosse realizada.

Já relacionado a luminosidade, houve sucesso em determinar se a iluminação de um ambiente está ligada ou não. Foi constatado que o sensor LDR possui baixa sensibilidade, necessitando estar próximo à fonte de luz para que se conseguisse o resultado desejado. Entretanto, esse fator atenuou a interferência da iluminação externa, proveniente de portas e janelas, o que garantiu que o resultado obtido estivesse dentro do esperado. Como o dispositivo monitorador fica localizado próximo teto, a incidência de luz é suficiente para alterar o valor de resistência do sensor.

O sistema desenvolvido não foi finalizado a tempo. Dessa forma, apenas o controle de usuário foi implementado.

5 CONSIDERAÇÕES FINAIS

Os resultados obtidos permitiram concluir que é possível montar um sistema organizado que possa auxiliar na sinalização de equipamentos que tenham sido esquecidos ligados e sem uso. O caso em que houve maior assertividade foi o da iluminação, bastando apenas que o dispositivo monitorador seja iniciado com as luzes ligadas para a realização da calibragem.

Ainda que não tenha sido finalizado, foi validada a possibilidade de um sistema para monitoramento através de uma interface *web*, que pode ser acessada por qualquer navegador de Internet. Isso otimiza o tempo do operador, que não precisará mais realizar a verificação manual de cada ambiente.

REFERÊNCIAS

- OLIVEIRA, Bruno A. S.; Assis, Servílio S.; Nolli, Carlos, R. **Desenvolvimento de um protótipo de sistema de monitoramento de energia elétrica via Internet**. Departamento de Engenharia e Computação, IFMG, Petrópolis, v. 12, n. 1, p. 48-61, Nov. 2018
- ESP32 - WiFi + Bluetooth. **RoboCore**. Disponível em: <https://www.robocore.net/wifi/esp32-wifi-bluetooth>. Acesso em: 21 mai. 2023.
- What is PlatformIO?. **PlatformIO**. Disponível em: <https://docs.platformio.org/en/latest/what-is-platformio.html>. Acesso em: 22 mai. 2023.
- How MQTT Works -Beginners Guide. **Steve's Internet Guide**. Disponível em: <http://www.steves-internet-guide.com/mqtt-works/>. Acesso em: 20 mai. 2023.
- Protocolo MQTT: O Que é, Como Funciona e Vantagens. **Automação Industrial**. Disponível em: <https://www.automacaoindustrial.info/mqtt/>. Acesso em: 28 mai. 2023.
- Eclipse Mosquitto: An open source MQTT broker. **MOSQUITTO**. Disponível em: <https://mosquitto.org/>. Acesso em: 29 mai. 2023.
- What is Java technology and why do I need it?. **JAVA**. Disponível em: https://www.java.com/en/download/help/whatis_java.html. Acesso em: 20 mai. 2023.
- Gradle User Manual. **Gradle**. Disponível em: <https://docs.gradle.org/current/userguide/userguide.html>. Acesso em: 19 mai. 2023.
- Spring Boot. **Spring**. Disponível em: <https://spring.io/projects/spring-boot>. Acesso em: 20 mai. 2023.
- LDR – O que é e como funciona!. **Manual da Eletrônica**. Disponível em: <https://www.manualdaeletronica.com.br/ldr-o-que-e-como-funciona/>. Acesso em: 1 jun.2023.
- STROUSTRUP, Bjarne. An Overview of C++. AT&T Bell Laboratories.Murray Hill, New Jersey. 1986.
- PEZOA, Felipe; Reutter, Juan L; Suarez, Fernando; Ugarte, Martín; Vrgoč, Domagoj. **Foundations of JSON Schema**. PUC Chile. p. 11, 2016