

# Engenharia de Telecomunicações

Sistema de Monitoramento e Aviso de Equipamentos Ligados Sem Uso

João Pedro Menegali Salvan Bitencourt

joao.ms@aluno.ifsc.edu.br

26/06/2023



**INSTITUTO  
FEDERAL**  
Santa Catarina

---

Câmpus  
São José

## 1 Introdução

## 2 O Sistema

- Desenvolvimento do AnJo
- Desenvolvimento do Servidor

## 3 Resultados

# Introdução

- 1 Equipamentos esquecidos ligados
  - Luz, ar-condicionado, projetor, computador etc.
- 2 Elaboração de um sistema de monitoramento através de sensores
  - Dispositivo monitorador
  - Servidor para consolidação dos dados
- 3 Impedir que equipamentos fiquem ligados desnecessariamente
  - Sinalizar os responsáveis por desligá-los
  - Desligamento remoto via infravermelho



O Sistema

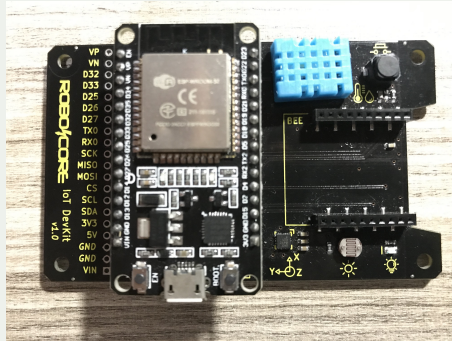
# O Sistema

Desenvolvimento do AnJo

- IoT DevKit LoRaWAN
  - ESP32-WROOM-32
  - WiFi
  - Sensor de temperatura e umidade (DHT11)
  - FOTOSensor
  - Botão
  - LED



Figura 1: Placa utilizada durante o desenvolvimento



Fonte: Autoria própria



- Fotorresistor (LDR - *Light Dependent Resistor*)
  - Reduz a resistência quanto maior for a presença de fótons
  - Os dados são obtidos através de um conversor analógico para digital
  - Os valores variam de 4095 a 0, o qual zero indica a maior incidência de luz
  - Possui baixa sensibilidade
- Temperatura e Umidade - DHT11
  - Temperatura em graus Celsius
  - Umidade em porcentagem
  - Os dados são obtidos através pulsos digitais e interpretados por uma biblioteca própria



- Utiliza-se o módulo Wi-Fi para conectar a placa de desenvolvimento à rede.
- As informações para a conexão são informadas no momento da compilação do programa que será executado.
- O módulo utiliza o conversor analógico para digital como interface de comunicação
- Há conflito entre o LDR e o dispositivo sem fio



- Programado utilizando a plataforma PlatformIO
- Linguagem C++
- Criação de classes C++ para cada componente da aplicação:
  - Lógica de detecção de luminosidade
  - Sensor de temperatura e umidade
  - Lógica para detectar se o ar-condicionado está ligado
  - Configuração da conexão sem fio
  - Envio dos dados coletados



- A linguagem de programação C++ permite separar um grande programa em pedaços, que podem ser chamados de classes. Por exemplo, abaixo, está a declaração da classe "LDR":

```
1 #ifndef __LDR_H__
2 #define __LDR_H__
3
4 #include <Arduino.h>
5 #include <list>
6
7 class LDR {
8     public:
9         LDR(int pin = 15);
10        void measure();
11        void measureBaseValue();
12        int getCurrent();
13        int getBaseValue();
14        bool getStatus();
15        ~LDR() {};
16    private:
17        int pin = 0;
18        int baseValue = 0;
19        int currentValue = 0;
20 };
21
22 #endif
```



Com isso, foram criadas as seguintes classes:

- LDR
- AC
- AnjofiMqtt
- AnjofiWifi
- TemperatureHumidity



- 1 Mede o valor inicial de luminosidade, a qual deve ser a maior possível de se obter. Dessa forma, é preciso que a luz do ambiente esteja ligada;
- 2 Obtém os valores atualizados de temperatura, umidade e luminosidade;
- 3 Conecta-se na rede sem fio;
- 4 Conecta-se no servidor MQTT;
- 5 Envia os dados obtidos e consolidados;
- 6 Desconecta-se da rede sem fio e desliga o módulo Wi-Fi;
- 7 Volta ao passo 2.



- Como o módulo da rede sem fio é conflitante com o uso do fotosensor, em decorrência de utilizarem o mesmo conversor analógico para digital, a rede Wi-Fi é desligada durante a obtenção dos dados dos sensores
- Os dados são montados no formato JSON, conforme exemplificado abaixo:

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



# Gravação do Programa na Placa

- É utilizado o PlatformIO para compilar e gravar na placa;
- Todas as especificações são informadas no arquivo de configuração "platformio.ini":

```
1 [env:esp32dev]
2 platform = espressif32
3 board = esp32dev
4 framework = arduino
5 lib_deps =
6   adafruit/DHT sensor library@^1.4.4
7   adafruit/Adafruit Unified Sensor@^1.1.9
8   knolleary/PubSubClient@^2.8
9   arduino-libraries/ArduinoMqttClient@^0.1.7
10
11 build_flags =
12   -DBROKER_SERVER="jpmsb.ddns.net"
13   -DBROKER_PORT=51439
14   -DBROKER_USER="anjofi"
15   -DBROKER_PASS="pji29006"
16   -DWIFI_SSID="${sysenv.WIFI_SSID}"
17   -DWIFI_PASS="${sysenv.WIFI_PASS}"
18   -DDEVICE_ID="${sysenv.DEVICE_ID}"
```





- Por fim, a gravação é realizada com o comando abaixo:

```
1 WIFI_SSID="Nome da Rede" WIFI_PASS="senha" DEVICE_ID="$(xxd -l 4 -p /dev/random)" pio run -t upload
```

- As informações para conexão de rede são informadas no momento da compilação, obtidas das variáveis de ambiente "WIFI\_SSID" e "WIFI\_PASS";
- O código identificador do dispositivo é gerado aleatoriamente e gravado na variável de ambiente "DEVICE\_ID";



- Todos os dados coletados e consolidados são enviados para um servidor externo, utilizando o protocolo MQTT;
- MQTT opera no modelo publicador/inscrição, ou *Publish/Subscribe* em inglês, cujo servidor é chamado de *broker*;
- Toda mensagem é publicada em um tópico;
- O dispositivo atua como publicador e o sistema desenvolvido atua como um cliente inscrito;
- O dado é publicado em um subtópico definido como "AnJoFi/ID\_do\_dispositivo";
- O servidor desenvolvido para este projeto, fica inscrito no tópico geral "AnJoFi/#", que permite obter as mensagens de todos os subtópicos;



# O Sistema

Desenvolvimento do Servidor

- Utiliza linguagem de programação JAVA;
- Usa a plataforma *Spring Boot* com *Gralde* como base para o servidor;
- Obtém os dados provenientes do servidor MQTT;
- Disponibiliza uma interface *web* para interação com o usuário;
- Possui controle de acesso, exigindo usuário e senha;



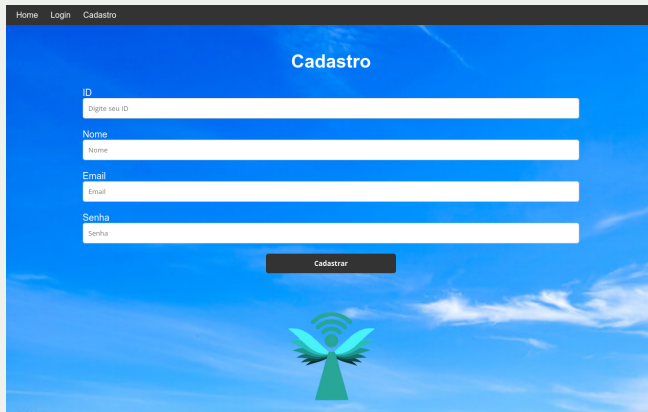
Figura 2: Página inicial.



Fonte: Autoria própria



Figura 3: Página para cadastro de usuário.



The image shows a web registration page titled "Cadastro". At the top, there is a dark navigation bar with links for "Home", "Login", and "Cadastro". The main content area has a blue background with a light blue sky and clouds. It contains four white input fields for "ID", "Nome", "Email", and "Senha", each with a small placeholder text. Below the fields is a dark "Cadastrar" button. At the bottom center is a green logo of a bird with its wings spread, emitting a signal wave.

Home Login Cadastro

## Cadastro

ID  
Digite seu ID

Nome  
Nome

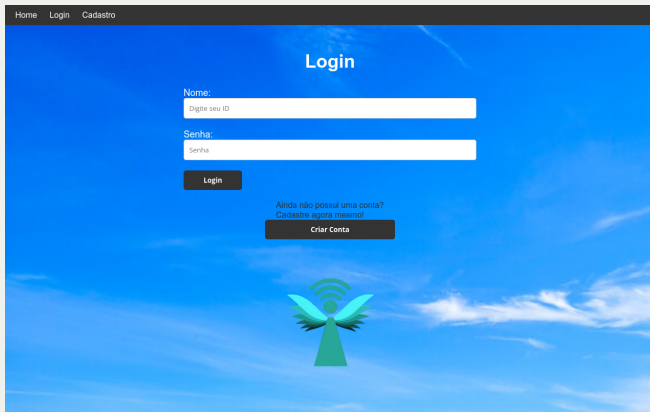
Email  
Email

Senha  
Senha

Cadastrar

**Fonte:** Autoria própria

Figura 4: Página de acesso.



The image shows a web login page with a blue sky background. At the top, there is a dark navigation bar with links for 'Home', 'Login', and 'Cadastro'. The main heading is 'Login'. Below it, there are two input fields: 'Nome:' with a placeholder 'Digite seu ID' and 'Senha:' with a placeholder 'Senha'. A 'Login' button is positioned below the password field. To the right of the login button, there is a link that says 'Ainda não possui uma conta? Cadastre agora mesmo!'. Below this link is a 'Criar Conta' button. At the bottom center, there is a green logo of an angel with wings and a halo.

**Fonte:** Autoria própria

Figura 5: Página do usuário

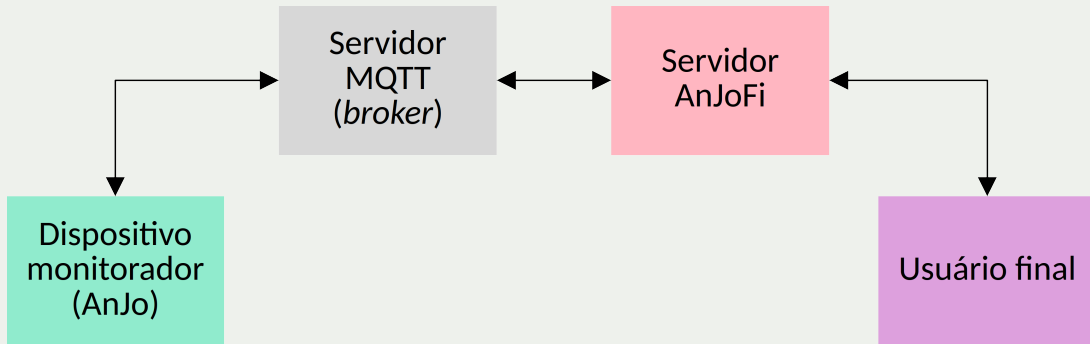


Fonte: Autoria própria





Figura 6: Estrutura de comunicação entre as partes do sistema.



**Fonte:** Autoria própria

## Resultados

## O que foi realizado

- Foi possível conceber um dispositivo que, através do sensor de luminosidade consegue identificar se a iluminação no ambiente está ligada ou não;
- Os dados coletados são enviados para um servidor externo, o que permite o processamento dos mesmos;
- Foi viabilizado um servidor que provê acesso *Web* para a visualização desses dados. Dessa forma, o responsável por verificar se algum equipamento ficou ligado, consegue economizar tempo indo apenas nos locais indicados;
- O servidor utiliza comunicação através de *API Rest*, que permite a criação de programas para interagir com o servidor. Por exemplo: aplicativo móvel.



- Neste projeto, o sensor de temperatura DHT11 demonstrou-se impreciso, de tal forma que o algoritmo para detectar se o ar-condicionado está ligado foi falho;
- O sensor de luminosidade precisa estar próximo à fonte de luz;
- O conflito de uso entre o fotosensor e o módulo de rede sem fio impedem que o dispositivo fique continuamente conectado. Isso dificulta algumas implementações, como o comando remoto para desligar o ar-condicionado.



- Ajustar o algoritmo de detecção de ar-condicionado ligado;
- Interface *Web* para uma configuração inicial do dispositivo físico;
- Possibilidade de envio de comando remoto para desligar o ar-condicionado via infravermelho;





Automação Industrial.

**Protocolo mqtt: O que é, como funciona e vantagens.**

<https://www.automacaoindustrial.info/mqtt/>, data não fornecida.

Acesso em: 28 mai. 2023.



Manual da Eletrônica.

**LDR - O que é e como funciona!**

<https://www.manualdaeletronica.com.br/ldr-o-que-e-como-funciona/>, data não fornecida.

Acesso em: 1 jun. 2023.



Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgo.

**Foundations of json schema.**

In *PUC Chile*, page 11, 2016.



Bjarne Stroustrup.

**An overview of c++.**

*AT&T Bell Laboratories*, page 4, 1986.

