COORDENADORIA DE ENGENHARIA DA COMPUTAÇÃO

ANDRESSA BRAGA VIEIRA RODRIGUES

DIOGO VITAL VIEIRA

GABRIEL COELHO CRISPI

GUILHERME SIQUEIRA DE GRADO PEREIRA

JOÃO VITOR MICHEL SILVA

LUCAS GARCIA FRAGOSO

AUTOMAÇÃO DE JANELAS COM CENTRAL DE NOTIFICAÇÕES

SOROCABA - SP

2023

ANDRESSA BRAGA VIEIRA RODRIGUES
DIOGO VITAL VIEIRA
GABRIEL COELHO CRISPI
GUILHERME SIQUEIRA DE GRADO PEREIRA
JOÃO VITOR MICHEL SILVA
LUCAS GARCIA FRAGOSO

AUTOMAÇÃO DE JANELAS COM CENTRAL DE NOTIFICAÇÕES

Relatório de entrega pra disciplina de Usina de Projetos Experimentais apresentado ao Centro Universitário Facens como parte da avaliação continuada na disciplina UP013LTIN3.

Orientador:
Prof. Dr. Gabriel Lobão Vasconcelos Fré

SOROCABA - SP
2023

Agradecimentos

Ao professor Gabriel Lobão Vasconcelos Fré, a coordenadora Andréa Lucia Braga Vieira Rodrigues, ao LINCE Facens e aos funcionários Kariston Henrique de Oliveira e Gustavo Abreu.

”Automação não é apenas sobre máquinas, mas também sobre melhorar a qualidade de vida das pessoas.”

(Indra Nooyi)

Resumo

O presente trabalho aborda problemas relacionados à conforto e mobilidade do usuário, baseando-se em cenários com pequenas ou grandes chuvas, que necessitam do desenvolvimento de variadas soluções. No entanto, o uso preponderanmente de sistemas manuais demandam exuberante alteração em sua estrutura, além de alterosa supervisão. Ao propor um sistema de automatização de janelas para dias chuvosos, a partir de um sensor de umidade, um sensor de fim de curso, um servo motor e um Arduino Uno, juntamente com um software, permite identificar, registrar e tomar as devidas medidas de maneira eficiente e versátil. Desse modo, a segurança dos ambientes que foram adaptados com tal solução é aprimorada, sendo possível controlar o sistema de funcionamento da janela pelo celular, tal como identificar o inicio de chuvas, para a execução do processo. Além de receber um alerta via Telegram, se está chovendo em sua residencia e se as medidas foram tomadas. Além disso, o projeto pode contribuir para a otimização de processos e facilitar a locomoção de seu consumidor ao evitar que precise retornar a sua residência para tomar as medidas necessárias contra a chuva.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

A automação de residências e edifícios tem se tornado cada vez mais popular, permitindo a implementação de soluções inteligentes e inovadoras que proporcionam segurança e conforto aos usuários. Nesse contexto, um exemplo relevante é o sistema de fechamento automático de janelas em caso de chuva, que utiliza um sensor que detecta chuva , um motor de passo, uma chave fim de curso e dois microcontroladores o Arduino e o esp01 para controlar o processo de fechamento das janelas.

Embora a ideia de um sistema automatizado para fechamento de janelas em situações de chuva não seja recente, como pode ser visto em MATOS 2009 ”Janela Residencial Automatizada”, os avanços na tecnologia dos microcontroladores alcançaram sua implementação mais acessível e viável para residências e edifícios. O microcontrolador Arduino foi escolhido como a plataforma central desse sistema devido à sua popularidade e versatilidade.

No decorrer deste trabalho, serão vistos detalhes sobre os componentes utilizados, o funcionamento do sistema, os benefícios oferecidos e os possíveis aprimoramentos que podem ser implementados, baseado nos trabalhos de BRAGA 2010 ”Janela Automatizada para Smart Houses com Sensor de Chuva e Aviso por SMS”e CANUTO 2021 ”Janela Automatizada que Aciona com Aproximação e Chuva”. Acredita-se que essa solução de automação de janelas contribuirá para otimizar os processos, melhorar a segurança e facilitar a vida dos usuários ao evitar a necessidade de retornar à residência para tomar medidas necessárias em caso de chuva.

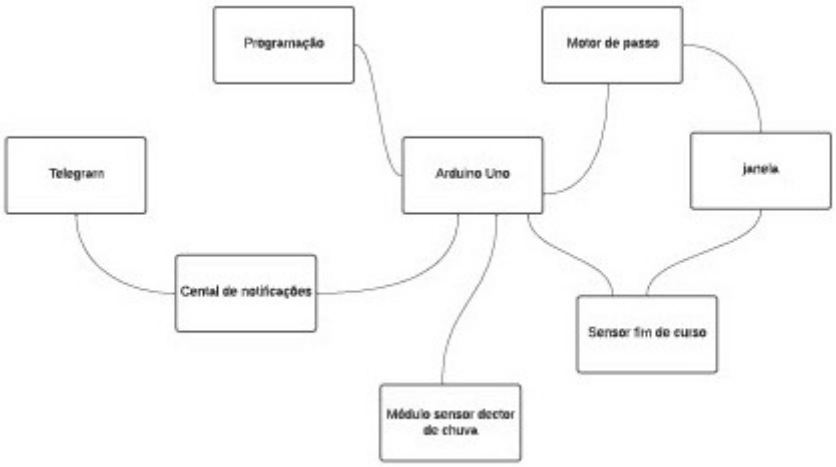

Figura 1: Diagrama de Blocos

2 Funcionamento

O sistema é composto por dois microcontroladores principais: o Arduino e o ESP01. Cada um desempenha funções específicas e trabalham em conjunto para o funcionamento adequado do sistema. O Arduino é responsável pelo controle dos componentes, como o fim de curso, o sensor de chuva e o motor de passo. Ele recebe e envia dados para o ESP01 por meio de comunicação serial. Essa interação é explicada de forma mais detalhada no tópico 2.1.

Por outro lado, o ESP01 tem a função de estabelecer a conexão via Wi-Fi do sistema. Ele envia e recebe os dados da API do Telegram, que são detalhados no tópico 2.2. Após o recebimento dos dados da API, o ESP01 realiza o tratamento necessário e envia os dados processados para o Arduino, também por meio de comunicação serial.

Essa estrutura de comunicação e colaboração entre o Arduino e o ESP01 é essencial para o correto funcionamento do sistema, garantindo que os componentes sejam controlados adequadamente e que os dados sejam transmitidos de maneira eficiente e precisa.

Ao abordar os componentes utilizados, são identificados:

- **Protoboard:** Para a montagem do circuito do sistema. Essa escolha foi feita levando em consideração que o projeto tinha como objetivo principal a pesquisa da viabilidade do sistema, e não a produção em massa. Dessa forma, optou-se pela utilização do protoboard devido à sua flexibilidade e facilidade de prototipagem, permitindo ajustes e alterações rápidas durante o desenvolvimento. Em função dessas considerações, não foi desenvolvida uma placa de circuito impresso (PCB) estruturada especificamente para o projeto.

- **Módulo Sensor Detector de Chuva:** Tem a função de verificar a presença de chuva. Quando o sensor detecta chuva, ele envia um sinal ou dado correspondente para o Arduino. Essa informação é importante para o sistema, pois desencadeia ações específicas, como acionar o mecanismo de proteção e enviar a notificação ao usuário. A partir do sinal recebido, o Arduino executa as instruções necessárias para lidar adequadamente com a detecção de chuva.

- **Motor de Passo:** É utilizado em conjunto com a engrenagem e a cremalheira para realizar a abertura e o fechamento da janela. Após receber um comando do Arduino, o motor de passo começa a rotacionar para um lado específico, rotacionando a engrenagem para posicionar corretamente a janela conforme desejado. Essa interação entre o Arduino e o servomotor permite o controle preciso e automatizado do movimento da janela, garantindo sua abertura e fechamento adequados.

- **Chave Fim de Curso:** Desempenha um papel fundamental no sistema ao possibilitar a identificação precisa do estado da janela, ou seja, se está aberta ou fechada. Essa informação é de extrema importância, pois evita redundâncias nas ações executadas, como tentar fechar uma janela que já está fechada. Com base no sinal emitido pelo fim de curso, o sistema pode tomar

as decisões apropriadas e realizar as ações necessárias, garantindo o correto funcionamento e evitando operações desnecessárias.

Para uma melhor compreensão, serão apresentados dois modos de execução do sistema. O primeiro modo é acionado quando o sensor de chuva detecta a presença de chuva. O segundo modo é ativado através de comandos enviados pelo Telegram, que podem ser para abrir, fechar ou verificar o estado da janela.

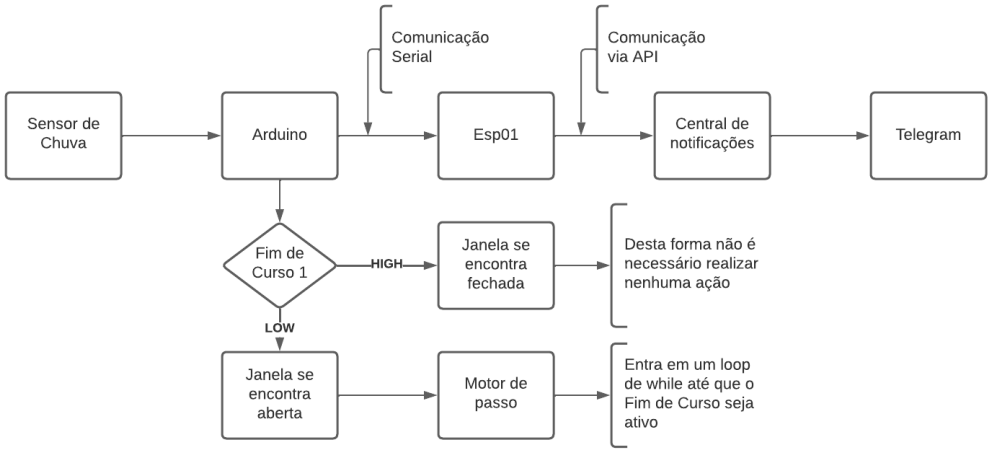

Figura 2: Fluxograma Processo 1

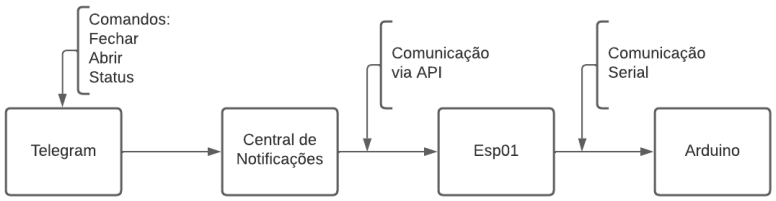

Figura 3: Fluxograma Processo 2

2.1 Comunicação Serial

A comunicação serial é um método de transferência de dados sequencial, em que os bits de informação são transmitidos em uma sequência ordenada, um após o outro, através de um único canal de comunicação.

A comunicação serial envolve dois componentes principais: um transmissor (ou emissor) e um receptor. O transmissor converte os dados em uma sequência de bits e os envia para o receptor. O receptor, por sua vez, recebe essa sequência de bits e decodifica os dados para utilização ou processamento posterior.

Para realizar a comunicação serial de maneira eficiente e confiável, é necessário definir e seguir um

protocolo de comunicação. Esse protocolo inclui acordos sobre a velocidade de transmissão (baud rate), formato dos dados, bits de paridade (opcional), controle de fluxo (opcional) e outros parâmetros relevantes.

No contexto do projeto em questão, a comunicação serial foi utilizada entre o Arduino e o ESP01, com uma velocidade de transmissão (baud rate) de 9600. Um exemplo aplicado no projeto é quando ocorre a detecção de chuva pelo sensor de chuva, um sinal é enviado para o Arduino. Em resposta, o Arduino inicia o fechamento da janela. Simultaneamente, o Arduino envia uma string (texto) de valor ”5”via comunicação serial. O ESP01 recebe essa string e envia uma notificação para o usuário através do aplicativo Telegram.

2.2 Comunicação Via API

O método utilizado para informar ao usuário sobre as alterações feitas na janela foi por meio da API do Telegram. Essa API possui algumas funionalidades como:

- Envio de mensagens - A API permite enviar mensagens de texto, mídia, como fotos e vídeos, documentos, áudios e até mesmo adesivos. As mensagens podem ser enviadas para usuários individuais, grupos ou canais.
- Gerenciamento de grupos e canais - A API permite criar, configurar e gerenciar grupos e canais no Telegram. É possível adicionar ou remover membros, definir permissões de administrador, configurar restrições de privacidade, obter informações sobre o grupo/canal e muito mais.
- Interação com bots - Os bots são uma parte essencial da API do Telegram. É possível criar bots usando a API e programá-los para responder a comandos, processar mensagens recebidas, fornecer informações, realizar tarefas automatizadas e interagir com os usuários de forma programática.
- Teclados personalizados - A API oferece suporte à criação de teclados personalizados, que permitem aos usuários interagir com botões predefinidos na interface do Telegram. Esses teclados facilitam a interação com os usuários e fornecem opções e ações rápidas.
- Consultas Inline - A API permite que os bots processem consultas inline, ou seja, consultas feitas diretamente na caixa de mensagem de chat. Com isso, os usuários podem obter informações rápidas, pesquisar conteúdo e realizar ações sem a necessidade de abrir uma conversa separada com o bot.
- Gerenciamento de notificações - A API oferece suporte a notificações push para aplicativos, permitindo enviar notificações para os usuários mesmo quando eles não estão ativamente usando o aplicativo Telegram. Isso pode ser útil para enviar alertas, atualizações ou mensagens importantes.

A comunicação entre a janela e o usuário foi programada por meio do token dado pelo BotFather (gerador oficial de bot do telegram). Com isso, a janela oferece as seguintes opções para o usuário: abrir e fechar a janela a qualquer momento, saber o status (se esta chovendo ou não), além de avisos sobre o fechamento e abertura automático a partir do sensor de chuva.

2.3 Telegram

A plataforma selecionada para funcionar como central de notificações foi o Telegram, devido à sua disponibilidade de um sistema oficial de criação de bots com acesso à API. Com o intuito de proporcionar facilidade de uso, o bot aceita 4 comandos: /start, /fechar, /abrir e /status. O comando /start apresenta uma mensagem de boas-vindas e, em seguida, fornece informações sobre os comandos disponíveis, como exemplificado na imagem a seguir:

Figura 4: Comando start no telegram

Prosseguindo com os comandos, os comandos /fechar e /abrir possuem o mesmo processo, enviam a solicitação para a API e, se a ação for concluída com sucesso, o usuário recebe a mensagem ”Janela fechando...”para o /fechar e ”Janela abrindo...”para o /abrir, conforme ilustrado nas imagens abaixo.

Figura 5: Comando fechar no telegram

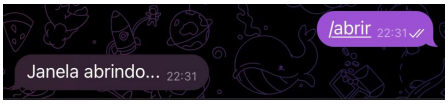

Figura 6: Comando abrir no telegram

Em caso de qualquer erro durante o processo, uma mensagem de erro será retornada ao usuário. Em relação ao último comando, o /status possui um procedimento um pouco diferente em comparação aos comandos anteriores. Ao enviá-lo, a API recebe a solicitação e a envia para o esp01, que, por sua vez, requisita ao arduino o estado do sensor de chuva por meio da comunicação serial. Quando o arduino responde ao esp01 com o estado, este informa à API, que retorna as mensagens ”Não está chovendo!”e ”Está chovendo!”, como exemplificado na figura abaixo:

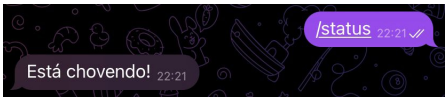

Figura 7: Comando status no telegram

3 Componentes e Orçamento

- Estrutura da Janela:
  - Quadro de Janela de 23cms*5,7cms
  - Quadro de Janela 60cms*5,7cms
  - Placa de MDF 30cms*20cms
  - Esquadrias para Janela
  - Trilho para Janela de Reta Flexível
  - Kit Rodas de Silicone gel
- Filamento - Utilizado para imprimir a engranegem e a cremalheira.
- Arduino Uno R3 - Placa de desenvolvimento eletrônico utilizada para criação de projetos.
- Protoboard 400 pontos - Placa de ensaio utilizada na montagem de circuitos eletrônicos.
- Cabo Wire Jumper 20cm 40 fios Fêmea-macho - Conjunto de cabos utilizados para fazer as conexões entre os componentes eletrônicos no protoboard.
- Cabo Wire Jumper 20cm 40 fios Macho-macho - Conjunto de cabos utilizados para fazer as conexões entre os componentes eletrônicos no protoboard.

- Módulo Sensor Detector de Chuva - Dispositivo eletrônico utilizado para detectar as gotículas de água da chuva.
- Motor de Passo 28byj-48 - Motor elétrico que se move em passos ao invés de girar continuamente, utilizado para fechar e abrir a janela.
- Chave Micro Switch Fim de Curso Kw11-7-3 - Interruptor mecânico utilizado para sinalizar que a janela chegou ao fim do percurso.
- Conversor de Nivel Lógico I2c 4 Canais Bidirecionais 5v 3.3v - Dispositivo eletrônico utilizado para converter sinais digitais de um nível de tensão lógica para outro.
- Esp-01 Wifi Esp8266 + Adaptador Usb Serial Ch340g Arduino - Microcontrolador utilizado para mandar sinal Wifi, para contatar o usuário.
- Bateria 9V - Bateria com tensão de 9 volts, utilizada para fornecer energia ao arduino.

Tabela 1: Orçamento dos componentes utilizados

Quantidade	Componente	Valor
2	Quadro da Janela 23cms*5,7cms	R$1,10
2	Quadro da Janela 60cms*5,7cms	R$2,85
2	Placa de MDF 30cms*20cms	R$5,00
2	Esquadrias para Janela	R$10,00
1	Trilho para Janela de Reta Flexível	R$10,00
1	Kit Rodas de Silicone gel	R$25,00
1	Filamento Abs 1.75mm 1kg	R$59,24
1	Arduino Uno R3	R$70,00
1	Protoboard 400 pontos	R$15,25
1	Cabo Wire Jumper 20cm 40 fios Fêmea-macho	R$15,29
1	Cabo Wire Jumper 20cm 40 fios Macho-macho	R$15,29
1	Módulo Sensor Detector de Chuva	R$23,19
1	Motor de Passo 28byj-48	R$29,90
1	Chave Micro Switch Fim de Curso Kw11-7-3	R$42,21
1	Conversor de Nivel Lógico I2c 4 Canais Bidirecional 5v 3.3v	R$20,19
1	Esp-01 Wifi Esp8266 + Adaptador Usb Serial Ch340g Arduino	R$29,99
1	Bateria 9V	R$30,00
Valor Total		R$404,50

4 Conclusão

Conclui-se que o projeto obteve o resultado esperado, atendendo ao objetivo inicialmente idealizado para sua utilização e implementação. Inicialmente a ideia era apenas um dispositivo para o controle de abertura e fechamento de janelas, porém ao decorrer do projeto, foi aprimorado com algumas ideias, tendo em sua versão final, uma comunicação via API para o aviso ao usuário, referente as janelas de sua casa, e se estavam fechadas ou abertas, podendo variar, dependendo das alterações climáticas.

O projeto teve um custo mediano para ser produzido, porém atende as necessidades a qual foi projetado, apesar de ser produzido apenas uma unidade para teste e experimentos, concluiu-se que poderá ser implementado em maior escala dentro de um ambiente de maneira interligada.

5 Sugestões para Trabalhos Futuros

À medida que o projeto avança, torna-se relevante considerar as possíveis melhorias futuras que podem aprimorar os resultados e aumentar o impacto do projeto realizado. Algumas sugestões de melhorias que podem ser implementadas no futuro foram identificadas:

- Implementação em outras centrais de notificações, deixando a escolha do usuário decidir qual é a melhor para ele.
- Junção com API de previsões de tempo, aumentando as interações possíveis com o usuário.
- Possível implementação em varais de roupas, utilizaria o mesmo código só seria necessário mudar alguns componentes para melhorar a viabilidade.
- Realização de uma PCB para otimizar o sistema e o espaço utilizado, tirando a necessidade da protoboard.

Referências Bibliográficas

BRAGA, Maria Luiza Oliveira. 2010. “Janela automatizada para smart houses com sensor de chuva e aviso por sms”.

CANUTO, Diego Gabriel. 2021. “Janela automatizada que aciona com aproximação e chuva”.

MATOS, Bruno Moreira. 2009. “Janela residencial automatizada”.

Apendice A - Código utilizado no Arduino

Aqui está o código utilizado no arduino:

```
#include <Stepper.h>   //biblioteca para controle de motor de passo

// --- Mapeamento de Hardware ---
#define   in1   8      //entrada 1 do ULN2003
#define   in2   9      //entrada 2 do ULN2003
#define   in3  10      //entrada 3 do ULN2003
#define   in4  11      //entrada 4 do ULN2003

const int stepsPerRevolution = 200;  // change this to fit the number of steps per revolution
Stepper myStepper(stepsPerRevolution, in1,in3,in2,in4);

// Pinos Sensor chuva
#define pinSensorA A0
#define pinSensorD 12
#define fimDeCursoPin 2
#define fimDeCursoPin2 3
#define DELAY_LEITURA 5000

bool fimDeCursoState;
unsigned long ultimaLeitura;

void setup() {
  pinMode(fimDeCursoPin, INPUT);
  pinMode(pinSensorD, INPUT);
  myStepper.setSpeed(90);

  Serial.begin(9600);
}

void loop() {
  fimDeCursoState = digitalRead(fimDeCursoPin);

  if(millis() - ultimaLeitura > DELAY_LEITURA){
    if(digitalRead(pinSensorD) == LOW){
```

```
      while(digitalRead(fimDeCursoPin) == LOW){
        Serial.println("5");
        myStepper.step(-stepsPerRevolution);
      }
    }//else{
      //Serial.println("8");
    //}
    ultimaLeitura = millis();
  }

  if(Serial.available()){
    String mensagem = Serial.readString();
    int mensagem_int = mensagem.toInt();
    //Serial.println(mensagem);
    //Serial.println(mensagem_int);
    if(mensagem_int == 1){
      while(digitalRead(fimDeCursoPin) == LOW){
        myStepper.step(-stepsPerRevolution);
      }
    }
    if(mensagem_int == 6){
      while(digitalRead(fimDeCursoPin2) == LOW){
        myStepper.step(stepsPerRevolution);
      }
    }
    if(mensagem_int == 4){
      if(digitalRead(pinSensorD) == HIGH){
        Serial.println("0");
      }
      if(digitalRead(pinSensorD) == LOW){
        Serial.println("1");
      }
    }
  }

}
```

Apendice B - Código utilizado no Esp01

Aqui está o código utilizado no esp01:

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>   // Universal Telegram Bot Library written by Brian Lough: http
Arduino-Telegram-Bot
#include <ArduinoJson.h>

// Replace with your network credentials
const char* ssid = "iPhone de Joao vitor";
const char* password = "12345678";

// Initialize Telegram BOT
#define BOTtoken "****"  // your Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
#define CHAT_ID "908161889"

#ifdef ESP8266
  X509List cert(TELEGRAM_CERTIFICATE_ROOT);
#endif

WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

const int ledPin = 2;
bool ledState = LOW;

// Handle what happens when you receive new messages
```

```
void handleNewMessages(int numNewMessages) {
  //Serial.println("handleNewMessages");
  //Serial.println(String(numNewMessages));

  for (int i=0; i<numNewMessages; i++) {
    // Chat id of the requester
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    //Serial.println(text);

    String from_name = bot.messages[i].from_name;

    if (text == "/start") {
      String welcome = "Welcome, " + from_name + ".\n";
      welcome += "Use os comandos abaixo para controlar suas saídas.\n\n";
      welcome += "/fechar para fechar a janela \n";
      welcome += "/abrir para abrir a janela \n";
      welcome += "/status para verificar o status da janela \n";
      bot.sendMessage(chat_id, welcome, "");
    }

    if (text == "/fechar") {
      bot.sendMessage(chat_id, "Janela fechando...", "");
      //ledState = HIGH;
      Serial.print("1\n");
      //digitalWrite(ledPin, ledState);
    }

    if (text == "/abrir") {
      bot.sendMessage(chat_id, "Janela abrindo...", "");
      //ledState = HIGH;
```

```
  Serial.print("6\n");
  //digitalWrite(ledPin, ledState);
}

/*if (text == "/led_off") {
  bot.sendMessage(chat_id, "LED state set to OFF", "");
  ledState = LOW;
  digitalWrite(ledPin, ledState);
}
*/

if (text == "/status") {
  Serial.println("4");
  delay(4000);
  if(Serial.available()){
    String mensagem = Serial.readString();
    int status = mensagem.toInt();
    if(status == 1){
      bot.sendMessage(chat_id, "Está chovendo!", "");
    }
    if(status == 0){
      bot.sendMessage(chat_id, "Não está chovendo!", "");
    }
    else{
      bot.sendMessage(chat_id, "Erro na resposta via Serial!", "");
    }
  }else{
    bot.sendMessage(chat_id, "Erro na comunicação!");
  }
  /*if (Serial.available()){
    String mensagem = Serial.readString();
    int status = mensagem.toInt();
    if(status == 1){
      bot.sendMessage(chat_id, "Está chovendo!", "");
    }
    if(status == 0){
      bot.sendMessage(chat_id, "Não está chovendo!", "");
```

```
        }
        else{
        bot.sendMessage(chat_id, "LED is OFF", "");
        }
      }*/
    }
  }
}

void setup() {
  Serial.begin(9600);

  #ifdef ESP8266
    configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
    client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
  #endif

  //pinMode(ledPin, OUTPUT);
  //digitalWrite(ledPin, ledState);

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  #ifdef ESP32
    client.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
  #endif
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }
  // Print ESP32 Local IP Address
  //Serial.println(WiFi.localIP());
}

void loop() {
  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
```

```
    while(numNewMessages) {
      //Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }

  if(Serial.available()){
    String chat_id = "908161889";
    String mensagem = Serial.readString();
    int teste = mensagem.toInt();
    //Serial.print(mensagem);
    if(teste == 5){
      Serial.println("Alerta chuva!");
      bot.sendMessage(chat_id, "Está chovendo, estou fechando!");
    }
  }
}
```