

USO DE ARDUINO PARA AUTOMAÇÃO DE UMIDIFICAÇÃO DE UM QUARTO

USING AN ARDUINO TO AUTOMATE THE HUMIDIFICATION OF A ROOM

Gabriel Fonseca de Souza¹
Alex Franco Ferreira²

RESUMO: Este artigo apresenta o desenvolvimento de um sistema de automação de baixo custo para o controle da umidade do ar em ambientes fechados, visando proporcionar um ambiente mais saudável, especialmente para pessoas com rinite alérgica. A metodologia empregada envolveu aplicar de conhecimentos em *Arduino*, pesquisa sobre o sensor DHT, testes em simulador, construção de um protótipo e experimentação. Os resultados esperados incluíam a capacidade de ligar e desligar umidificadores ou desumidificadores de forma autônoma, mantendo a umidade entre 40% e 60%. A conclusão demonstra a viabilidade e eficácia da solução proposta, destacando sua acessibilidade e o potencial da prototipagem eletrônica para resolver problemas cotidianos de forma inovadora e econômica.

Palavras-Chave: Rinite alérgica. *Arduino*. Umidade. Automação.

ABSTRACT: This article presents the development of a low-cost automation system for controlling indoor air humidity, aiming to provide a healthier environment, especially for people with allergic rhinitis. The methodology employed involved applying knowledge in *Arduino*, research on the DHT sensor, simulator testing, prototype construction, and experimentation. Expected results included the ability to autonomously turn humidifiers or dehumidifiers on and off, maintaining humidity between 40% and 60%. The conclusion demonstrates the feasibility and effectiveness of the proposed solution, highlighting its accessibility and the potential of electronic prototyping to solve everyday problems in an innovative and cost-effective way.

Keywords: Allergic rhinitis. *Arduino*. Humidity. Automation.

INTRODUÇÃO

A humanidade cria invenções para manipular o clima há muito tempo, uma das primeiras invenções que se assemelha a um umidificador moderno é o vaporizador elétrico, um umidificador portátil que usa um aquecedor elétrico para esquentar água e gerar vapor, assim aumenta a umidade num ambiente fechado (KATZMAN, 1927)

¹Discente do Curso de Pós-Graduação em Automação Industrial. Universidade de Vassouras.

²Docente do Curso de Pós-Graduação em Automação Industrial. Universidade de Vassouras.

Uma pessoa que precise de uma certa umidade pode o fazer manualmente, ligando um umidificador quando quiser, mas além de precisar verificar a umidade por conta própria, também não o poderá fazer se a mesma estiver ocupada ou dormindo. Um sistema automatizado poderia resolver o problema. Pessoas com rinite alérgica podem ser particularmente mais interessadas em tal sistema. Este trabalho foca em construir um sistema que seja capaz de ligar e desligar o umidificador baseando-se em dados de umidade coletados no ambiente, e com um custo baixo.

A metodologia utilizada neste estudo envolveu a aplicar conhecimentos prévios em *Arduino*, pesquisa aprofundada sobre como o sensor DHT funciona, a realização de testes em simulador, a construção de um protótipo e, por final, a execução de um experimento com o protótipo desenvolvido.

Rinite alérgica e umidade

A umidade relativa do ar afeta pessoas com rinite alérgica, sendo que níveis de umidade relativa abaixo de 40% podem ressecar a mucosa nasal, causar irritações, coceira, sangramento nasal e tosse seca, e níveis acima de 60% podem ajudar na proliferação de fungos, ácaros, bactérias e outros alérgenos (ROSENFIELD *et al.*, 2024). Portanto o nível recomendado ficaria entre 40% e 60% (GUARNIERI *et al.*, 2023). Uma pessoa com rinite alérgica que pode ficar muito tempo do dia em um quarto específico, seja dormindo ou num escritório trabalhando pode aproveitar bem uma umidade constante.

1958

Uma maneira de manter a umidade constante, ao menos em um quarto, é usando um umidificador de ar, mas para manter a umidade num nível adequado seria necessário que este desligasse quando a umidade estiver alta. Muitos umidificadores avançados já possuem esse sistema, porém estes tendem a ser mais caros, além de umidificadores mais antigos não possuírem um tipo de sistema como esse. Sendo necessário substituir um modelo mais simples por um mais avançado, ou a montagem dum sistema de ligamento automatizado por conta própria. O sistema ainda pode funcionar no caso oposto, em que um quarto é úmido demais, automatizando o desligamento e religamento de um desumidificador em vez de um umidificador.

Arduino

É facilmente possível montar um sistema em que um umidificador de ar é automaticamente ligado e desligado dependendo da umidade numa sala usando um *Arduino*.

O *Arduino*, conforme Figura 1, consiste plataforma para prototipagem eletrônica de código livre, criado por profissionais do *Interaction Design Institute Ivrea*, na Itália, estes criaram uma empresa também chamada *Arduino*, que mantém e desenvolve a marca e modelos de *Arduino* até os dias de hoje. Um *Arduino* consiste basicamente de uma placa de circuitos com um microcontrolador e pinos de saída e entrada digitais e analógicos. Esses pinos podem ler diversos tipos de sinais de entrada, como um botão, um sinal sonoro, um sensor de umidade. E o *Arduino* consegue transformar sinais de entrada em sinais de saída, como acender um LED, escrever num *Display LCD*, ligar um relé. A linguagem usada pelo *Arduino* se chama *Arduino IDE*, criada pela própria *Arduino*. Há outros tipos de microcontroladores, como o *ESP32*, porém o *Arduino* é um dos de maior simplicidade e, também, um dos mais baratos sendo o bom o bastante para a automatização necessária nesse projeto. Tendo uma grande comunidade que pode te ajudar com problemas ao usar o *Arduino* (BANZI, 2021).

Figura 1: *Arduino UNO R3*



Fonte: Arduino Store (2019)

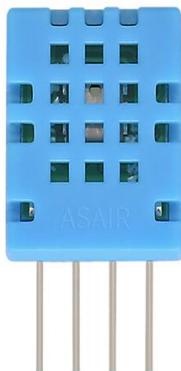
1959

Detector de umidade

O dispositivo usado nesse sistema é o DHT, modelo DHT11, como se observa na Figura 2. O DHT11 inclui um microcontrolador junto de um tipo de termistor Coeficiente de temperatura negativo, é um tipo de resistor que com o aumento da temperatura a resistência diminui, e para umidade um elemento capacitivo formado por dois eletrodos e um substrato higroscópico no meio. A umidade altera a constante dielétrica do material, modificando sua

capacitância, através do microcontrolador essa variação é convertida em um sinal elétrico (ASAIR, 2023). O DHT₁₁ tem a vantagem de ser barato, pequeno e fácil de usar.

Figura 2: DHT₁₁



Fonte: ASAIR (2024)

Relé

1960

Um relé, observado na Figura 3, é um componente eletromecânico que age como um interruptor, geralmente contém uma bobina eletromagnética, um comutador, um sistema de mola e terminais: comum, normalmente aberto e normalmente fechado. Um campo magnético é criado quando uma corrente elétrica passa pela bobina, o campo atrai o comutador que altera sua posição, se estava desconectado (normalmente aberto), fecha o circuito, se estava conectado (normalmente fechado), abre o circuito. O sistema de rota retorna o relé a posição original quando a bobina deixar de receber energia (BOYLESTAD, 1987).

Figura 3: Um módulo de relé com LED



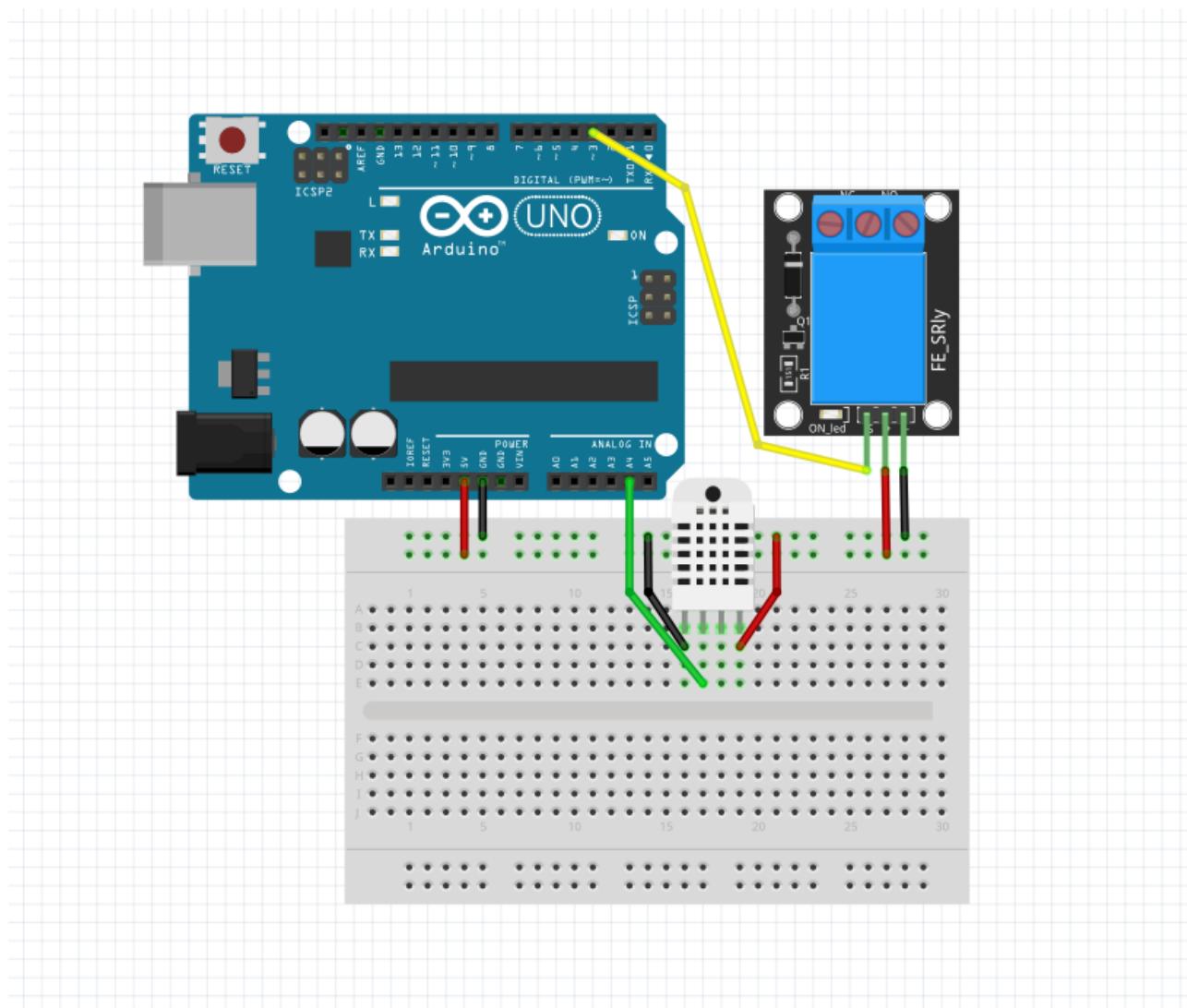
Fonte: Mercado Livre (2025)

Sistema

O Software usado para demonstrar o sistema é chamado de *Fritzing*, uma plataforma de código aberto criada por um time da *University of Applied Sciences Potsdam* na Alemanha em 2007. É um software de design e simulação de componentes eletrônicos (FRITZING, 2025).

Usando um sensor de umidade DHT11 este consegue enviar um sinal para um *Arduino*, e este aciona um relé para ligar um umidificador de ar. O modelo pode ser visto na Figura 4

Figura 4: Modelo do sistema feito no software *Fritzing*

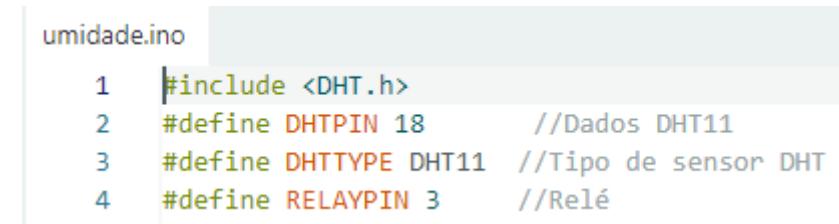


Fonte: Autores (2025)

O *software* usado para programação se chama *Arduino IDE*, um *software* para programação de *Arduino* feito pela própria *Arduino* (BANZI, 2021). Para rodar o programa são necessárias algumas etapas.

O comando “include” inclui a biblioteca do DHT, necessária para o programa rodar as funções do DHT. O comando “define” defina qual pino do *Arduino* será usado por cada componente e também os atribui um nome de variável, e também define qual tipo de DHT está sendo usado. A primeira parte do código pode ser vista na Figura 5.

Figura 5: Primeira parte do código no *software Arduino IDE*

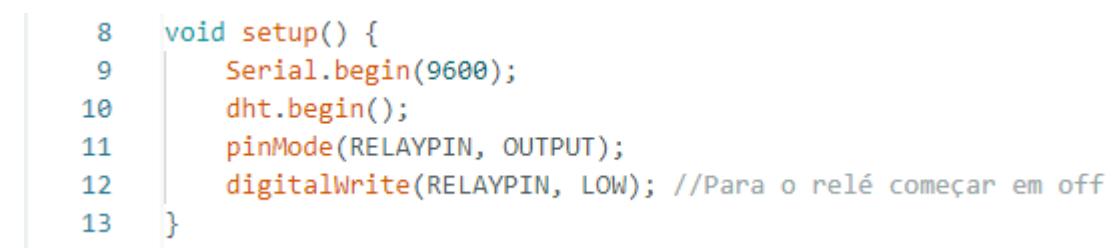


```
umidade.ino
1 #include <DHT.h>
2 #define DHTPIN 18      //Dados DHT11
3 #define DHTTYPE DHT11 //Tipo de sensor DHT
4 #define RELAYPIN 3    //Relé
```

Fonte: Autores (2025)

Os comandos dentro do “void setup” serão rodados apenas uma vez. O comando “pinMode” é usado para definir se o relé é uma entrada ou saída, “OUTPUT” é para saída. O comando “Serial.begin” é necessário para iniciar o Monitor Serial que é usado para monitorar o funcionamento do programa dentro do software. O comando “digitalWrite(RELAYPIN, LOW)” garante que o relé comece desligado. A estrutura de comando pode ser vista na Figura 6.

Figura 6: Segunda parte do código no *software Arduino IDE*



```
8 void setup() {
9   Serial.begin(9600);
10  dht.begin();
11  pinMode(RELAYPIN, OUTPUT);
12  digitalWrite(RELAYPIN, LOW); //Para o relé começar em off
13 }
```

Fonte: Autores (2025)

Os comandos dentro do “void loop” rodarão continuamente em um loop, com o comando “delay” decidindo o tempo de intervalo entre cada loop, o tempo escolhido é de 10 segundos, ou 10000 milissegundos. O comando “float humidity = dht.readHumidity” cria uma

variável com o nome “humidity” atribui os valores dessa variável como sendo números com casas decimais, e os lê. O comando “if isnan(humidity)” é para caso o Arduino não consiga ler o DHT escrevendo no Monitor Serial através do comando “Serial.println” a seguinte mensagem “Falha ao ler o sensor DHT11!”. Os comandos de “Serial.print” em “("Umidade: ")”, “(humidity)” e “("%")” são para escrever no Monitor Serial o valor da umidade lida. O comando “if (humidity < 40)” verifica se o nível de umidade lida é menor que 40 e o comando “digitalWrite(RELAYPIN, HIGH)” liga o relé, após isso, o comando “Serial.println("Relé LIGADO!")” escreverá no Monitor Serial que o relé foi ligado. Enquanto o comando “else if (humidity => 60)” verifica se o nível de umidade lida é maior ou igual que 60 e o comando “digitalWrite(RELAYPIN, LOW)” desliga o relé, após isso, o comando “Serial.println("Relé DESLIGADO!")” escreverá no Monitor Serial que o relé foi desligado.

O código funciona de uma maneira que o relé para ligar o umidificador apenas acione quando a umidade for abaixo que 40% e desligue se chegar a 60%, esse intervalo é para evitar que o umidificador fique ligando e desligando continuamente, se o relé for ligado, porém a umidade não chegar a 60% ele continuará ligado, da mesma forma se estiver desligado, mas não abaixar de 40% ele continuará desligado. Esta etapa do código pode ser visualizada na Figura 7.

1963

Figura 7: Terceira parte do código no *software Arduino IDE*

```
15 void loop() {  
16     delay(10000); //10 segundos de intervalo entre as medições  
17     float humidity = dht.readHumidity();  
18     if (isnan(humidity)) {  
19         Serial.println("Falha ao ler o sensor DHT11!");  
20         return;  
21     }  
22     Serial.print("Umidade: ");  
23     Serial.print(humidity);  
24     Serial.println("%");  
25     if (humidity < 40) {  
26         digitalWrite(RELAYPIN, HIGH); //Liga o relé abaixo de 40%  
27         Serial.println("Relé LIGADO!");  
28     }  
29     else if (humidity >= 60) {  
30         digitalWrite(RELAYPIN, LOW); //Desliga o relé acima de 60%  
31         Serial.println("Relé DESLIGADO!");  
32     }  
33 }
```

Fonte: Autores (2025)

O código completo incluindo o *void setup* e o *void loop* pode ser observado na Figura 8.

Figura 8: Código completo do sistema no *software Arduino IDE*

```
umidade.ino
1  #include <DHT.h>
2  #define DHTPIN 18      //Dados DHT11
3  #define DHTTYPE DHT11 //Tipo de sensor DHT
4  #define RELAYPIN 3     //Relé
5
6  DHT dht(DHTPIN, DHTTYPE);
7
8  void setup() {
9      Serial.begin(9600);
10     dht.begin();
11     pinMode(RELAYPIN, OUTPUT);
12     digitalWrite(RELAYPIN, LOW); //Para o relé começar em off
13 }
14
15 void loop() {
16     delay(10000); //10 segundos de intervalo entre as medições
17     float humidity = dht.readHumidity();
18     if (isnan(humidity)) {
19         Serial.println("Falha ao ler o sensor DHT11!");
20         return;
21     }
22     Serial.print("Umidade: ");
23     Serial.print(humidity);
24     Serial.println("%");
25     if (humidity < 40) {
26         digitalWrite(RELAYPIN, HIGH); //Liga o relé abaixo de 40%
27         Serial.println("Relé LIGADO!");
28     }
29     else if (humidity >= 60) {
30         digitalWrite(RELAYPIN, LOW); //Desliga o relé acima de 60%
31         Serial.println("Relé DESLIGADO!");
32     }
33 }
```

1965

Fonte: Autores (2025)

É possível ter o código facilmente modificado para comportar desumidificadores em vez de umidificadores, sendo necessário apenas uma pequena mudança nos comandos para assim o sistema funcionar ao contrário. Alterar para que o relé desligue quando a umidade estiver abaixo de 40 e ligar quando a umidade estiver maior ou igual que 60. Tal modificação pode ser vista na Figura 9.

Figura 9: Código para desumidificadores no software Arduino IDE

```

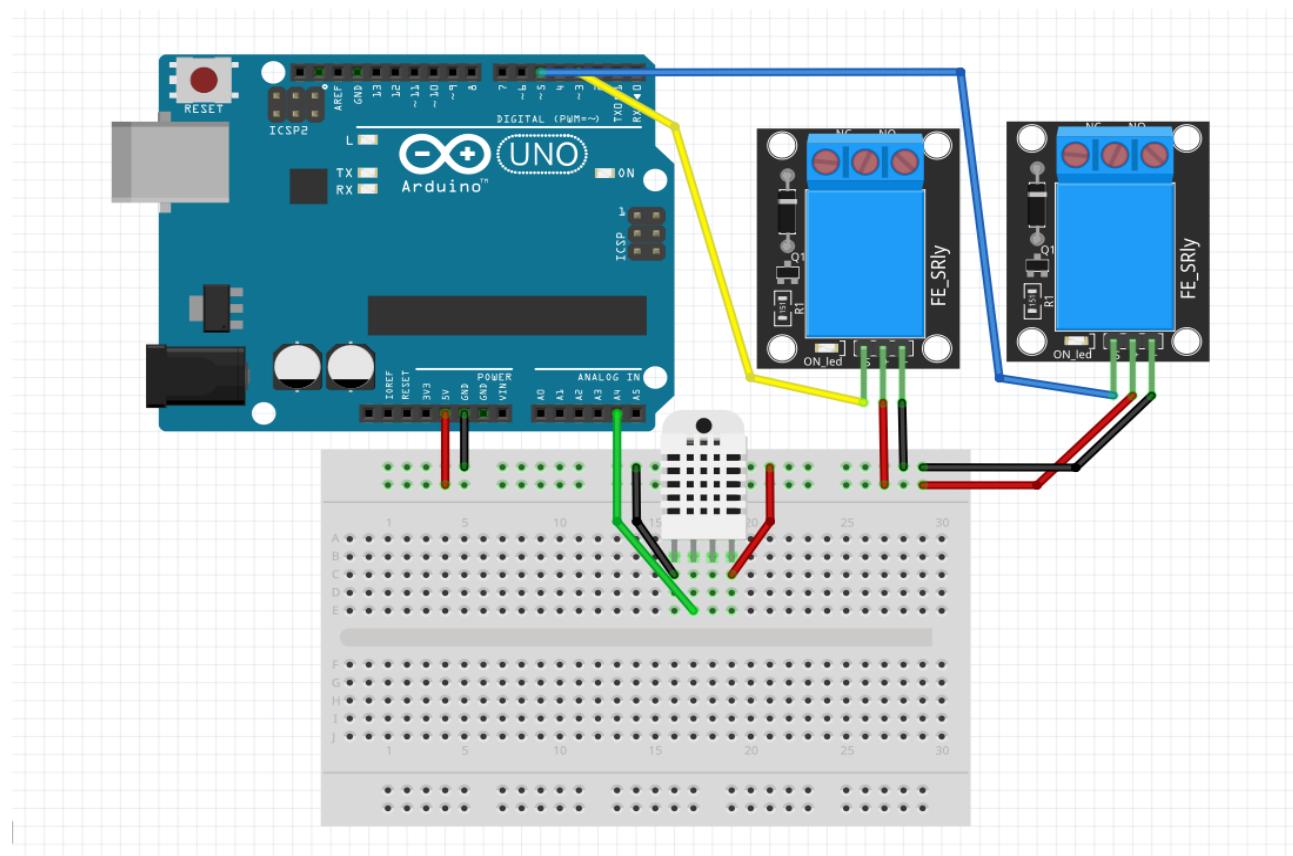
25      if (humidity < 40) {
26          digitalWrite(RELAYPIN, LOW); //Desliga o relé abaixo de 40%
27          Serial.println("Relé DESIGADO!");
28      }
29      else if (humidity >= 60) {
30          digitalWrite(RELAYPIN, HIGH); //Liga o relé acima de 60%
31          Serial.println("Relé LIGADO!");
32      }
33  }

```

Fonte: Autores (2025)

E ainda se pode fazer de um jeito a acionar ambos um umidificador e um desumidificador, ao adicionar mais um relé, conforme Figura 10, e modificando levemente o código.

Figura 10: Modelo do sistema com dois relés



1966

Fonte: Autores (2025)

Realizando assim o acréscimo de um segundo relé, para o código funcionar é necessário definir um segundo relé no código, conforme pode se observar na Figura 11.

Figura 11: Primeira parte do código com dois relés no software Arduino IDE

```
umidade.ino
1  #include <DHT.h>
2  #define DHTPIN 18      //Dados DHT11
3  #define DHTTYPE DHT11 //Tipo de sensor DHT
4  #define RELAY1 3      //Relé 1
5  #define RELAY2 5      //Relé 2
6
7  DHT dht(DHTPIN, DHTTYPE);
8
9  void setup() {
10    Serial.begin(9600);
11    dht.begin();
12    pinMode(RELAY1, OUTPUT);
13    pinMode(RELAY2, OUTPUT);
14    digitalWrite(RELAY1, LOW);
15    digitalWrite(RELAY2, LOW); //Para ambos relés começarem em off
```

Fonte: Autores (2025)

Quando a umidade for abaixo de 40%, o relé do umidificador é ligado, e é desligado quando a umidade for acima de 55%. E também quando a umidade for acima de 70% o relé do desumidificador é ligado e é desligado quando a umidade for abaixo que 60%. Ainda mostrando o valor da umidade no Monitor Serial e quando cada relé desliga e liga. Tal parte do código é complementada na Figura 12.

1967

Figura 12: Segunda parte do código com dois relés no software Arduino IDE

```
28  if (humidity > 40) {
29    digitalWrite(RELAY1, HIGH); //Liga o relé abaixo de 40%
30    Serial.println("Umidificador LIGADO!");
31  }
32  else if (humidity <= 55) {
33    digitalWrite(RELAY1, LOW); //Desliga o relé acima de 55%
34    Serial.println("Umidificador DESLIGADO!");
35  }
36  if (humidity > 70) {
37    digitalWrite(RELAY2, HIGH); //Liga o relé acima de 70%
38    Serial.println("Desumidificador LIGADO");
39  } else if (humidity <= 60) {
40    digitalWrite(RELAY2, LOW); //Desliga o relé abaixo de 60%
41    Serial.println("Desumidificador DESLIGADO");
42  }
43 }
```

Fonte: Autores (2025)

O código completo com os dois relés pode ser observado na Figura 13.

Figura 13: Código completo com dois relés no *software Arduino IDE*

```
umidade.ino
1  #include <DHT.h>
2  #define DHTPIN 18      //Dados DHT11
3  #define DHTTYPE DHT11 //Tipo de sensor DHT
4  #define RELAY1 3      //Relé 1
5  #define RELAY2 5      //Relé 2
6
7  DHT dht(DHTPIN, DHTTYPE);
8
9  void setup() {
10    Serial.begin(9600);
11    dht.begin();
12    pinMode(RELAY1, OUTPUT);
13    pinMode(RELAY2, OUTPUT);
14    digitalWrite(RELAY1, LOW);
15    digitalWrite(RELAY2, LOW); //Para ambos relés começarem em off
16  }
17
18 void loop() {
19  delay(10000); //10 segundos de intervalo entre as medições
20  float humidity = dht.readHumidity();
21  if (isnan(humidity)) {
22    Serial.println("Falha ao ler o sensor DHT11!");
23    return;
24  }
25  Serial.print("Umidade: ");
26  Serial.print(humidity);
27  Serial.println("%");
28  if (humidity > 40) {
29    digitalWrite(RELAY1, HIGH); //Liga o relé abaixo de 40%
30    Serial.println("Umidificador LIGADO!");
31  }
32  else if (humidity <= 55) {
33    digitalWrite(RELAY1, LOW); //Desliga o relé acima de 55%
34    Serial.println("Umidificador DESLIGADO!");
35  }
36  if (humidity > 70) {
37    digitalWrite(RELAY2, HIGH); //Liga o relé acima de 70%
38    Serial.println("Desumidificador LIGADO");
39  } else if (humidity <= 60) {
40    digitalWrite(RELAY2, LOW); //Desliga o relé abaixo de 60%
41    Serial.println("Desumidificador DESLIGADO");
42  }
43 }
```

Orçamento

Os componentes principais desse sistema são um *Arduino*, um ou dois relés e um sensor de umidade, sendo possível comprá-los por um valor inferior a R\$104,00 em 2025 em lojas de eletrônicos na internet. Umidificadores de ar podem ser encontrados em diversos tamanhos e preços sendo os modelos simples cerca de R\$20,00, enquanto os mais avançados com sistema automatizado de desligamento e religamento são encontrados custando mais de R\$200,00. Os fios necessários para o sistema podem ser comprados avulsos em lojas de eletrônica presencialmente, ou também na internet em sites de vendas de materiais eletrônicos, porém estes vendem os fios em pacotes com 10, 20, 40 ou mais. Um pacote com 10 fios de 20 cm pode chegar a custar até R\$12,00. O orçamento pode ser observado no Quadro 1.

Quadro 1: Orçamento

Material	Quantidade	Preço por unidade (R\$)
<i>Arduino UNO R3</i>	1	39,87
Módulo relé 5 V	2	19,01
Sensor DHT11	1	16,50
Kit cabos jumper	1	12,00
Preço Total		103,88

Fonte: Mercado Livre (2025)

1969

CONCLUSÕES

Este trabalho demonstrou com sucesso a concepção e implementação de um sistema de automação de custo baixo para o controle da umidade do ar em ambientes fechados. A meta de criar uma solução barata e eficaz que mantenha a umidade em níveis ideais, especialmente para pessoas com rinite alérgica, foi alcançada. Com o uso de componentes como o *Arduino*, o sensor DHT11 e um relé se foi possível construir um dispositivo que, de forma automática, liga e desliga umidificadores ou desumidificadores, garantindo um ambiente mais apropriado. A flexibilidade do sistema, demonstrada pela capacidade de adaptação para diferentes cenários e a possibilidade de controle simultâneo de umidificadores e desumidificadores, ressalta a robustez da solução. Além do mais, a análise orçamentária confirmou a viabilidade econômica do projeto, posicionando-o como uma alternativa significativamente mais barata em comparação com equipamentos comerciais de alta tecnologia.

Em síntese, este estudo provou que é possível desenvolver soluções tecnológicas práticas e de baixo custo para problemas cotidianos, também sendo um modelo funcional que pode ser replicado e aprimorado. A simplicidade e a eficácia do sistema proposto abrem caminho para futuras inovações, como a integração com redes sem fio e sistemas de monitoramento remoto, ampliando o alcance e a utilidade desta abordagem para a automação residencial.

REFERÊNCIAS

- ARDUINO STORE. Arduino Uno Rev3. Disponível em: <<https://store.arduino.cc/products/arduino-uno-rev3>>. Acesso em: 8 jul. 2025.
- ASAIR. ASAIR DHT11 Temperature and Humidity Sensor Module. Disponível em: <<https://aosong.com/en/Products/info.aspx?itemid=2257>>. Acesso em: 8 jul. 2025.
- BANZI, M. et al. About Arduino. Arduino, 15 set. 2021. Disponível em: <<https://www.arduino.cc/en/about/>>. Acesso em: 20 jun. 2025.
- BOYLESTAD, R. L.; NASHESKY, L. Electronic devices and circuit theory. 4. ed. Harlow, England: Longman Higher Education, 1987.
- GUARNIERI, G. et al. Relative humidity and its impact on the immune system and infections. International journal of molecular sciences, v. 24, n. 11, 2023.
- HOROWITZ, S. H.; PHADKE, A. G. Relay operating principles. Em: Power System Relaying. Chichester, UK: John Wiley & Sons, Ltd, 2008. p. 23–48. 1970
- MERCADO LIVRE. Placa Compatível Com Arduíno Uno R3 Atmega328 Smd. Disponível em: <<https://www.mercadolivre.com.br/placa-compativel-com-arduino-uno-r3-atmega328-smd/p/MLB37804532>>. Acesso em: 8 jul. 2025.
- MERCADO LIVRE. Módulo de relé de relé 1 canal 5v com Leds. Disponível em: <<https://www.mercadolivre.com.br/modulo-de-rele-de-rele-1-canal-5v-com-leds/p/MLB34957631>>. Acesso em: 8 jul. 2025.
- MERCADO LIVRE. Modulo / Sensor Dht11 Temperatura E Umidade - Arduino - Pic. Disponível em: <https://produto.mercadolivre.com.br/MLB-3801458330-modulo-sensor-dht11-temperatura-e-umidade-arduino-pic-_JM>. Acesso em: 8 jul. 2025.
- MERCADO LIVRE. Kit Jumper Macho Macho 20cm - 10 Fios. Disponível em: <<https://www.mercadolivre.com.br/kit-jumper-macho-macho-20cm-10-fios/p/MLB44219038>>. Acesso em: 8 jul. 2025.
- ROSENFIELD, L. et al. Allergic rhinitis. Allergy, asthma, and clinical immunology: official journal of the Canadian Society of Allergy and Clinical Immunology, v. 20, n. Suppl 3, p. 74, 2024.



THOMSEN, A. Monitorando Temperatura e Umidade com o sensor DHT11. *MakerHero*, 5 ago. 2013. Disponível em: <<https://www.makerhero.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>>. Acesso em: 4 jul. 2025.