Sistema de Monitoramento e Controle para Desumidificadores de Ar Industriais



Erik de Oliveira Rosa[1]; Lincoln Cezar Grabarski[1]; Marcos Fernando Fragoso[1]; Carlos Alexandre Gouvea da Silva[1]
[1]*Centro Universitário Unifacear*

RESUMO

A constante evolução tecnológica tem permitido avanços significativos e à melhorias nos processos das indústrias, principalmente em áreas que necessitam maior controle e eficiência do ambiente. O presente trabalho tem por objetivo apresentar o desenvolvimento de um sistema para desumidificador de ar industrial, que possibilite o monitoramento e controle de suas falhas remotamente a partir de uma comunicação de dados confiável entre os equipamentos envolvidos no processo. O sistema terá como base um equipamento da empresa Munters, onde possibilitará o aperfeiçoamento e um desempenho com maior eficiência do desumidificador. O controle desse equipamento será realizado através de um aplicativo mobile para Sistema Operacional Android. O sistema permitirá realizar paradas programadas para manutenção preventiva e/ou corretiva, monitoramento de todos os sensores do equipamento e através da função registros, poderá ser visualizado quais falhas estão ativas. O protótipo deste trabalho foi desenvolvido, os componentes são os mesmos utilizados nos equipamentos da empresa Munters, sendo assim é possível simular o funcionamento real do equipamento Munters.

Palavras chave: Desumidificador de ar, Comunicação de dados, Aplicativo, Manutenção preventiva.

ABSTRACT

The constant technological evolution has allowed significant advances and improvements in industrial processes, especially in areas that need greater control and environmental efficiency. The present work aims to present the development of a system for industrial air dehumidifier, which allows remote monitoring and control of its faults through reliable data communication between the equipment involved in the process. The system will be based on equipment from the company Munters, where it will enable the improvement and performance with greater efficiency of the dehumidifier. Control of this equipment will be performed through a mobile application for Android Operating System. The system will allow scheduled shutdowns for preventive and / or corrective maintenance, monitoring of all sensors of the equipment and through the function records, which faults are active. The prototype of this work was developed, the components are the same used in the equipment of the company Munters, so it is possible to simulate the actual operation of the equipment Munters.

Key Words: Air dehumidifier, Data communication, Application, Preventive maintenance.

1.INTRODUÇÃO

Na qualidade do ar, a automação industrial tem evoluído e proporcionado um número crescente de técnicas para controle de temperatura, umidade e remoção de resíduos poluentes de ambientes industriais. As consequências de uma má qualidade do ar variam de gravidade, pois podem causar doenças respiratórias nas pessoas, no qual em ambiente industrial pode gerar um número maior de ausências por motivo de doenças ou queda na capacidade produtiva dos colaboradores. Os desumidificadores de ar em geral são importantes para manter uma apropriada umidade do ar nos ambientes, em especial naqueles onde esse controle é exigido por normas técnicas e regulatórias externas, ou procedimentos internos das indústrias (MULTITEC, 2018).

Segundo a Thermomatic (2019) esses equipamentos tem o objetivo de modificar as características de umidade de um ambiente, retirando o excesso de umidade do ar e filtrando-o para se obter um ambiente saudável. Um dos principais desafios nesses ambientes industriais é como realizar o monitoramento dos desumidificadores através de uma rede confiável de comunicação de dados, evitando a necessidade de deslocamento de técnicos até o local onde se encontra o equipamento.

O objetivo desse artigo é implementar um sistema de monitoramento para desumidificadores de ar industriais para a empresa Munters Brasil, que auxilie o cliente a diminuir o tempo ocioso do equipamento no seu processo, através de um aplicativo Android o usuário poderá visualizar em tempo real as falhas e alertas, além de possibilitar o monitoramento do equipamento.

Além dessa seção introdutória, a seção 2 apresenta o desenvolvimento com apresentação de fundamentos técnicos e científicos importantes para o projeto e a metodologia onde os procedimentos de planejamento e execução do projeto são apresentados. Os resultados e a discussão desses são mostrados na seção 3. Por fim, na seção 4 a conclusão é apresentada.

2. DESENVOLVIMENTO E METODOLOGIA

2.1. DESUMIDIFICADORES DE AR INDUSTRIAIS

Os desumidificadores de ar em geral são importantes para manter uma umidade do ar apropriada nos ambientes onde estão instalados. Segundo a Thermomatic (2019) esses equipamentos tem o objetivo de transformar um ambiente, retirando o excesso de umidade do ar e filtrando-o para se obter um ambiente saudável. A umidade relativa do ar ideal para a saúde das pessoas deve estar entre 50% a 60%, sendo os

desumidificadores essenciais nesse controle em ambientes como comércios, indústrias e residências (THERMOMATIC, 2019).

2.2. CONTROLADOR LÓGICO PROGRAMÁVEL

Equipamentos e componentes são cada vez mais utilizados para automatizar diversos processos na indústria, como por exemplo, aqueles destinados a controle da qualidade do ar, e entre os diversos equipamentos disponíveis o controlador lógico programável (CLP) é um dos mais utilizados. Segundo Capelli (2008, p. 20) “O primeiro CLP foi desenvolvido no final de 1960. Até então, os controladores eram grandes armários de relés eletromecânicos com vários quilômetros de fio, [...]”. Um exemplo de CLP pode ser observado na FIGURA 1.



FIGURA 1 – CLP DA MARCA ELIWELL
FONTE: ELIWELL (2019).

2.3. SENSORES

Existe uma grande variedade de sensores que são utilizados nas indústrias, comércios e nas residências. Para Groover (2011, p. 93), “Um sensor é um transdutor, no qual o dispositivo que converte uma variável física de uma forma para outra mais útil para a aplicação em questão”.

2.3.1. Sensor de temperatura Pt1000

Segundo Citisystems (2019) “O sensor de temperatura é um dispositivo de medição que detecta a temperatura a partir de uma característica física correspondente do dispositivo, como uma resistência elétrica, o campo eletromagnético (EMF) ou radiação térmica”.

2.3.2. Sensor de umidade

Nexxto (2017) esclarece que “O higrômetro, ou sensor de umidade, é um equipamento capaz de mensurar e exibir em um *display* a umidade relativa do ar. Ele pode ser utilizado tanto ao ar livre como também em ambientes fechados”, este instrumento é muito empregado em comércios, indústrias alimentícias, entre outros ambientes que necessitem de uma umidade controlada.

2.4. INTERFACE HOMEM–MÁQUINA

A Interface Homem-Máquina, também conhecida como IHM, é definida como todas as partes que fornecem informações e controle de um sistema interativo, ao usuário para realizar determinada tarefa, como sendo um ponto de interação entre o usuário e a máquina (COPADATA, 2019). Para Cordeiro, Oliveira e Chanquini (2009) IHM é resultado de boas práticas de programação inseridas no processo de desenvolvimento de software durante a construção da interface gráfica de qualquer sistema. A interface gráfica é uma parte importante do software, pois é por meio dela que o usuário se comunica.

2.5. SISTEMA EMBARCADO ARDUINO

De acordo com Martinazzo *et al.* (2014, p. 24), atualmente a diversidade de placas microcontroladas de baixo custo é ampla, contudo a que tem maior destaque é o Arduino baseado no microcontrolador AVR da Atmel. O desenvolvimento das aplicações é de código aberto e a linguagem padrão utilizada é C/C++. Possuindo diversas portas analógicas e digitais, o Arduino permite a leitura de dezenas de sensores e de acordo com o conhecimento do programador essas informações podem ser tratadas em programas de tratamentos de dados para análises posteriormente.

2.5.1. Linguagem C

Segundo Mizrahi (1990, p. 02), “A linguagem C foi criada inicialmente por Dennis M. Ritchie e Ken Thompson no laboratório Bell em 1972 baseando-se na linguagem de programação B criada por Thompson a partir da evolução da antiga linguagem BCPL”. A linguagem de programação é muito popular entre as arquiteturas, uma vez que, são poucas as que não possuem compiladores para essa linguagem.

2.6. AMBIENTE DE DESENVOLVIMENTO ANDROID

Lançado em 2008 e baseado no Linux o Android é um sistema operacional projetado para dispositivos móveis. De acordo com Pereira e Silva (2012, p.01) “ao longo dos anos, a quantidade de telefones celulares vem aumentando cada vez mais. Atualmente, o celular é o produto de consumo mais utilizado no mundo, sendo a quantidade existente correspondente a mais da metade da população” mundial. No Android é possível gerenciar o processamento do dispositivo e realizar outras funções de *software* e *hardware*, possibilitando o funcionamento dos dispositivos.

2.6.1. Software Development Kit (SDK) / Java Development Kit (JDK)

Aplicativos dos mais variados podem ser instalados nos *smartphones* com sistema operacional Android, IOS ou *Windows Phone*, o que inclui uma variedade de aplicações como: agenda, calendário, conversor de moeda, navegação em sites da Internet, clima, sistemas de geoposição (GPS, *Global Position System*), entre diversos outros. De acordo com Lecheta (2013, p. 32) “o Android SDK é o *software* utilizado para desenvolver aplicações do Android, que tem um emulador para simular a tela do celular, ferramentas utilitárias e uma API completa para a linguagem Java”.

2.6.2. Android Studio

O Android Studio é um ambiente de desenvolvimento integrado para desenvolvimento de aplicativos para o sistema Android. Segundo o site Developers (2019) o Android Studio é baseado no *IntelliJ IDEA* (*Integrated Development Environment Apache*) que oferece recursos como o editor de código fonte e as ferramentas de desenvolvedor avançadas do *IntelliJ.*

É bem evidente que o número de dispositivos Android está aumentando cada vez mais, para site de tecnologia AndroidPro (2019), “Claramente, há uma demanda de desenvolvimento de aplicativos Android muito grande, [...]. Podemos dizer que é uma plataforma empolgante e o espaço para fazer aplicativos é gigantesco”.

2.7. MODBUS

O protocolo Modbus foi desenvolvido pela Modicon em 1979, que consiste numa estrutura de mensagem aberta, sendo o protocolo mais utilizado em equipamentos de

automação industrial para a comunicação entre dispositivos, devido à simplicidade na sua implementação (EMBARCADOS, 2014).

Com relação ao funcionamento do Modbus, Moraes e Castrucci (2013, p. 170) explicam que “A tecnologia de comunicação no protocolo Modbus é o mestre-escravo, sendo que somente um mestre e no máximo 247 escravos podem ser conectados à rede”.

2.8. COMUNICAÇÃO

O projeto consiste em um sistema que facilite o operador a identificar possíveis falhas e também visualizar os valores dos sensores de controle do equipamento. Devido que muitas das vezes o equipamento se encontra em um local dentro da industria aonde o acesso é restrito, é necessário um tempo de resposta adequado da equipe de manutenção de forma que não ocasione atrasos para realizar manutenções e ligar o equipamento. Nesta seção é apresentado o processo metodológico para planejamento, elaboração e execução do projeto.

É apresentada na FIGURA 2 toda a comunicação dos componentes, sendo iniciada pelo CLP que faz a leitura dos sensores do equipamento enviando para a IHM e a mesma informação é transmitida para o módulo MAX485, que possibilita a comunicação com o Arduino recebendo esses dados e processando para enviá-los ao aplicativo, por meio do módulo Ethernet, que em conjunto com o roteador se conecta na rede WiFi.

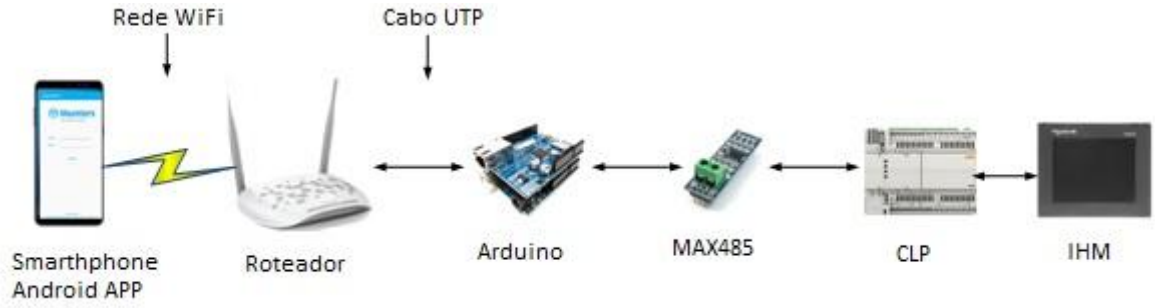


FIGURA 2 – COMUNICAÇÃO GERAL DO SISTEMA
FONTE: OS AUTORES (2019)

2.9. CONSTRUÇÃO DA BANCADA

A FIGURA 3 apresenta o protótipo elaborado para o monitoramento do sistema de ar. Esse protótipo é composto por quatro partes principais: a IHM (1) que mostra os dados do equipamento; o conjunto de LEDs, potenciômetros e chaves (2) que terão a

função de simular as falhas e alertas da máquina; os disjuntores (3) que protegem o sistema contra curto-circuito e sobrecarga; o CLP (4) que está sendo alimentado pela fonte 110 Vca para 24 Vcc, é responsável por toda a lógica do equipamento e o Arduino (5) contém toda a lógica encarregada por receber os dados do CLP e transmitir para o *smarthphone, juntamente com o MAX485 (5) que converte o sinal RS485 para TTL.*

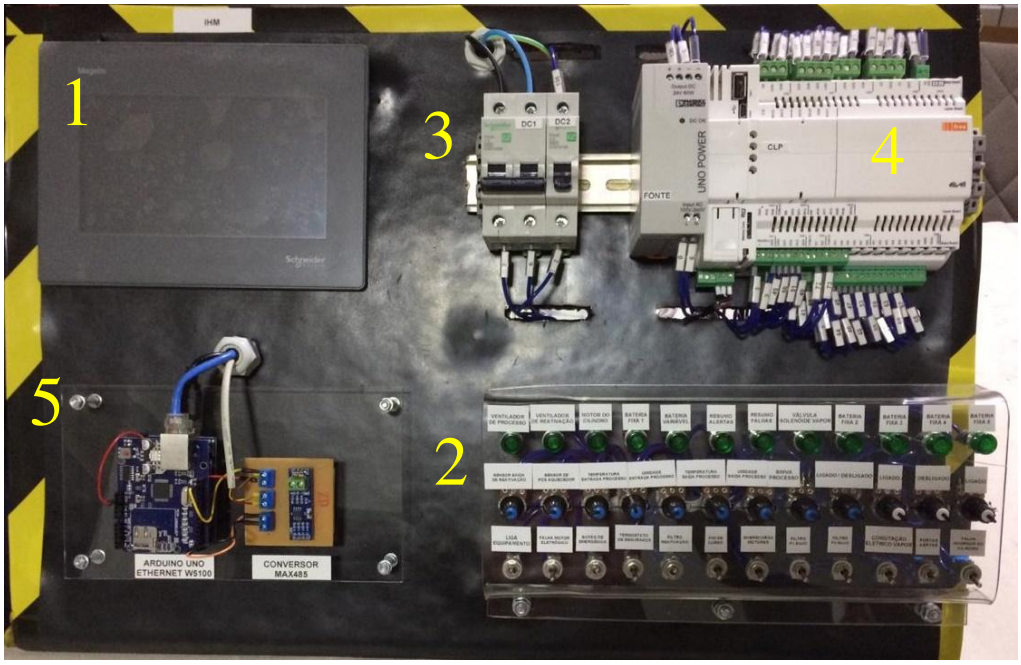


FIGURA 3 – PARTE FRONTAL DA BANCADA
FONTE: OS AUTORES (2019)

Os componentes do protótipo como a IHM, através de sua interface, mostram os valores dos sensores de temperatura e umidade que variam em uma faixa de acordo com sua característica, altera o *setpoint*, recebe alarmes e falhas do equipamento. O disjuntor bipolar de 6A é responsável pela proteção de sobrecarga da fonte. O disjuntor monopolar de 6A protege contra sobrecarga da linha 24 Vcc de comando após a fonte. A fonte de tensão reduz a tensão de 110 para 24 Volts e converte a tensão de alternada (Vac) para contínua (Vcc). Esta fonte é utilizada para alimentar o CLP e a IHM. O CLP contém toda a programação do equipamento, fornece as informações para a IHM e para o Arduino. O sinaleiro LED tem a função de sinalizar quando uma saída digital do equipamento é acionada.

2.10. MÓDULO MAX485

Após a bancada finalizada foi iniciado os trabalhos de comunicação entre o Arduino e o CLP. Para isso, foi utilizado o meio físico serial RTU RS-485, no qual para essa comunicação foi utilizado o módulo max485 que realiza a conversão de níveis lógicos de TTL para RS-485. A FIGURA 4 apresenta o módulo de comunicação:

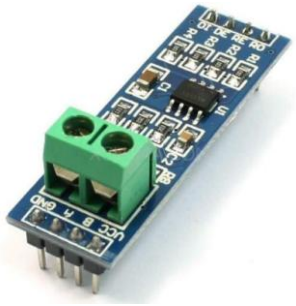


FIGURA 4 – MÓDULO MAX485
FONTE: OS AUTORES (2019)

A partir do uso do MAX485 foi necessário realizar a ligação entre o Arduino e o CLP via comunicação serial, no qual o diagrama de ligação utilizado é apresentado na FIGURA 5.

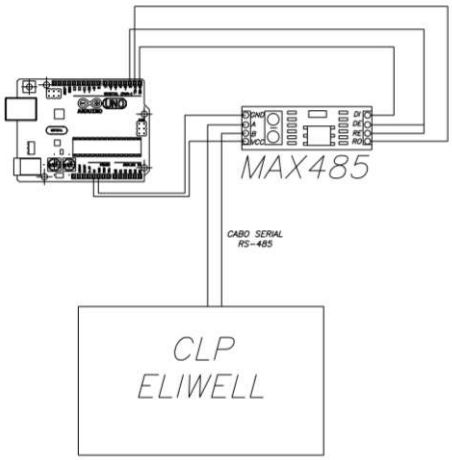


FIGURA 5 – DIAGRAMA DE LIGAÇÃO
FONTE: OS AUTORES (2019)

Para ligação do CLP no módulo MAX485 foi utilizado a saída RS485-2 do CLP interligando na entrada A e B no módulo, já para na alimentação foi utilizado os pinos VCC e GND do módulo e para a transmissão dos dados foi utilizado o RE e DE conectados juntos e ligado no pino 02 do Arduino.

Após a ligação do módulo de comunicação serial foi utilizado à interface de programação do Arduino de forma que o Arduino recebesse as informações dos sensores

da bancada através do CLP. Os dados de parâmetros que foram acessados no CLP ficam armazenados em elementos de memória chamados de registradores, e dentro desses registradores existem subconjuntos que exercem uma função dependendo de sua característica. Neste trabalho foram utilizados os *holding registers,* aonde a sua principal função é fazer a leitura e escrita dos dados com uma mensagem de 16 bits em um endereço de 40001 a 49999.

Para a programação foi adicionado á biblioteca *modbusslave*, no qual representa um trecho do software *modbus* já previamente pronto. Depois, foi escrito via linha de programação todas as linhas de código na qual foram requisitados os valores de parâmetros necessários. Para cada tipo de dado acessado é associado o registrado correto para acesso àquele dado, como mostrado na FIGURA 6. No código *enum* se declara MB_4000 que indica o registrador do CLP que será lido.

```
enum {
  //ENTRADAS ANALÓGICAS
  MB_40000,       //A0
  MB_40001,       //SRT_TEMPERATURA_SAIDA_REATIVACAO
  MB_40002,       //PA_TEMPERATURA_POS_AQUEC_REATIV
  MB_40003,       //ST1_TEMPERATURA_ENTRADA_PROCESSO
  MB_40004,       //SU1_UMIDADE_ENTRADA_PROCESSO
  MB_40005,       //ST2_TEMPERATURA_SAIDA_PROCESSO
  MB_40006,       //SU2_UMIDADE_SAIDA_PROCESSO
  MB_40007,       //PRE_TEMPERATURA_PRE_RESFRIAMENTO
  MB_40008,       //POS_TEMPERATURA_POS_RESFRIAMENTO
  MB_40009,       //PSI1_SENSOR_PRESSAO_SIMVA_I
```

FIGURA 6 – CÓDIGO IDE
FONTE: OS AUTORES (2019)

2.11. MÓDULO ETHERNET W5100

Em plantas industriais existem uma grande quantidade de maquinários e processos que por natureza geram um alto índice de ruído que pode acarretar em altas taxas de perdas de pacotes (SILVA, SANTOS, OSINSKI, 2018). Por este motivo foi utilizado para a comunicação do Arduino para o aplicativo o módulo Ethernet, o cabo trás maior confiabilidade na transmissão de dados do que outros tipos. O módulo é encaixado em cima do Arduino e por este motivo não necessita de nenhuma ligação externa, isso pode ser visto na FIGURA 7:



FIGURA 7 – ETHERNET W5100
FONTE: REMOTEXY (2019)

Com este módulo é possível criar 04 *sockets* para a comunicação com o aplicativo, que através de um *Acess Point* como intermédio em modo *bridge*, a aplicação estabelece a conexão com o Arduino. Para essa comunicação, utilizando o programa IDE do Arduino o primeiro passo é incluir as bibliotecas SPI.h e Ethernet.h, elas já vem instaladas no IDE, depois declara-se informações de endereço IP no qual o aplicativo também estará configurado.

Um exemplo pode ser visto na FIGURA 8:

```
//Informacoes de endereco IP, gateway, mascara de rede
byte mac[] = { 0xA4, 0x28, 0x72, 0xCA, 0x55, 0x2F };
byte ip[] = { 192, 168, 0, 150 };
byte gateway[] = { 192, 168, 0, 1 };
byte subnet[] = { 255, 255, 255, 0 };
```

FIGURA 8 – CONFIGURAÇÃO DO MÓDULO ETHERNET
FONTE: OS AUTORES (2019)

Na Figura 8 mostra a configuração do módulo Ethernet na IDE do Arduino UNO, o *mac* é o endereço físico do módulo, no *ip* é o rótulo que atribuímos para o módulo, no *gateway* é inserido a faixa de *ip* utilizada para fazer a ponte entre o módulo e o aplicativo Android e no *subnet* é inserido a máscara de sub-rede da rede *ip*.

No próximo passo foi usado o método EthernetServer, que está incluso na biblioteca Ethernet.h, e indicar qual a porta que será utilizada para o envio dos pacotes de dados, como mostrado na FIGURA 9:

```
EthernetServer server(12345);
EthernetServer server2(12346);
EthernetServer server3(12347);
EthernetServer server5(12344);
```

FIGURA 9 – CONFIGURAÇÃO DOS SERVIDORES
FONTE: OS AUTORES (2019)

Depois de configurado os servidores é necessário iniciá-los, e para isso utiliza-se o comando server.begin(), como mostra a FIGURA 10:

```
server5.begin();
server.begin();
server2.begin();
server3.begin();
```

FIGURA 10 – INICIALIZAÇÃO DOS SERVIDORES
FONTE: OS AUTORES (2019)

3. ANÁLISES E RESULTADOS

Para a validação do sistema de monitoramento do desumidificador de ar industrial, foram realizados testes de transmissão de dados entre o CLP e o Arduino através do conversor MAX485. Com os dados recebidos no Arduino utilizou-se o *shield ethernet* para o envio de pacotes para o aplicativo Android. Como esperado, o aplicativo recebeu as informações que foram solicitadas do CLP e foram realizados os testes de funcionamento de um desumidificador. Esses resultados serão apresentados a seguir.

Na FIGURA 11 foi simulado o teste inicial com o equipamento no estado ideal de funcionamento, no qual o equipamento não apresentou nenhuma falha ou alarme ativo.



FIGURA 11 – ESTADO IDEAL DO EQUIPAMENTO
FONTE: OS AUTORES (2019)

Quando o equipamento detecta uma falha, automaticamente o equipamento entra em modo de desligamento, na FIGURA 12 foram simuladas as falhas do equipamento, no qual acionando as chaves foi causado os erros de emergência acionada, termostato de segurança, sobrecarga motores e aumentando a resistência do potenciômetro, geramos o erro do sensor de reativação e do pós aquecedor.



FIGURA 12 – TODOS OS REGISTROS ACIONADOS
FONTE: OS AUTORES (2019)

O fluxo de ar na reativação está operando pelo motivo que ele só entra em falha com o equipamento ligado, está falha se refere à chave de pressão diferencial, igual á dos filtros, porém ela é considerada uma falha porque está localizada após as resistências e por esse motivo é muito importante para a segurança do equipamento.

Outra função exercida pelo aplicativo é o monitoramento em tempo real, a FIGURA 13 mostra os valores recebidos do CLP através dos sensores de temperatura e umidade, que são simulados por meio da variação da resistência, onde utilizou-se os potenciômetros.



FIGURA 13 – MONITORAMENTO DO EQUIPAMENTO
FONTE: OS AUTORES (2019)

A FIGURA 13 mostra os valores em tempo real, como a temperatura de entrada de processo que está em 54,50ºC e após passar pelo cilindro do desumidificador chega ao ambiente com uma umidade de 51,90%, o mesmo acontece com a saída de processo, a leitura do sensor da saída de reativação interfere diretamente na umidade de entrada do processo, pois com base nela que o PID do CLP usa como referência para o controle. Essas informações são importantes para o usuário saber como está o funcionamento do equipamento e fazer as devidas modificações de acordo com as necessidades do processo.

4. CONCLUSÃO E CONSIDERAÇÕES FINAIS

O projeto atendeu a proposta inicial, principalmente na realização da comunicação entre o desumidificador e o aplicativo desenvolvido para Android, sendo possível a visualização em tempo real dos dados, onde o técnico responsável pela manutenção não necessitaria se deslocar até o equipamento verificar a possível falha.

Utilizou-se o *Shield Ethernet* W5100 como solução para a comunicação do Arduino, devido ao fato de receber menos interferências e de menor custo. O protótipo proporcionou um desempenho confiável, e de desenvolvimento acessível, cumprindo com os objetivos propostos.

O monitoramento do desumidificador desenvolvido apresentou um resultado aceitável nos testes, onde os erros apresentados na IHM, também são visualizados no aplicativo, assim como os valores dos sensores lidos pelo CLP é enviado em segundos para a tela de monitoramento do aplicativo Munters.

No futuro, é almejado realizar melhorias na comunicação, verificando a possibilidade de alterar o módulo ethernet para um *shield* que possui o envio de informações através da rede GSM/GPRS, onde não seria necessária uma rede local de comunicação via cabo. Seria interessante a implementação de *setpoint* no aplicativo, dando opção ao usuário setar variáveis sem a necessidade de deslocamento até a IHM.

Outra sugestão seria a inclusão de um banco de dados no aplicativo para que seja disponibilizado um histórico de erros e falhas ocorridas, facilitando uma manutenção mais precisa e reduzindo custos com gastos desnecessários.

REFERÊNCIAS

ANDROIDPRO. **Android Studio: configurando o ambiente de desenvolvimento Android.** 2019. Disponível em: <https://www.androidpro.com.br/blog/android-studio/android-studio-configurando-ambiente>. Acesso em: 20 abr. 2019.

CAPELLI, Alexandre. **Automação industrial: controle do movimento e processos contínuos.** 2. ed. São Paulo: Érica, 2008.

CITISYSTEMS. **Sensor de temperatura.** 2019. Disponível em: <https://www.citisystems.com.br/sensor-de-temperatura/>. Acesso em: 05 jun. 2019.

COPADATA. **Interface homem-máquina.** 2019. Disponível em: <https://www.copadata.com/pt/solucoes-hmi-scada/interface-homem-maquina-hmi/>. Acesso em: 05 jun. 2019.

CORDEIRO, Renato; OLIVEIRA, M. R.; CHANQUINI, T. **Utilização de conceitos de interface homem-máquina para adaptação da disciplina de requisitos do RUP**. São Paulo, 2009.

DEVELOPERS. **Conheça o Android Studio.** 2019. Disponível em: <https://developer.android.com/studio/intro>. Acesso em: 20 abr. 2019.

ELIWELL. **CLP AVC840006I500.** 2019. Disponível em: <https://www.eliwell.com/en/Partnumber/AVC840006I500.html>. Acesso em: 13 abr. 2019.

EMBARCADOS. **Protocolo Modbus.** 2014. Disponível em: <https://www.embarcados.com.br/protocolo-modbus/>. Acesso em: 05 jun. 2019.

GROOVER, Mikell. **Automação industrial e sistemas de manufatura.** 3. ed. São Paulo: Pearson Prentice Hall, 2011.

LECHETA, Ricardo R.. **Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK.** 3. ed. São Paulo: Novatec, 2013.

MARTINAZZO, Claodomir Antonio et al. Arduíno: Uma Tecnologia no Ensino de Física. **Perspectiva**, Erechim, v. 38, n. 143, p.21-30, set. 2014.

MIZRAHI, Victorine Viviane. **Treinamento em linguagem C.** 2. ed. São Paulo: Makron Books, 1990.

MORAES, Cícero C.; CASTRUCCI, Plínio de L.. **Engenharia de automação industrial.** 2. ed. Rio de Janeiro: LTC, 2013.

MULTITEC. **A importância da climatização no ambiente industrial.** 2018. Disponível em: <https://www.multtecriopreto.com.br/single-post/climatizacaoambienteindustrial>. Acesso em: 28 out. 2019.

NEXXTO. **Sensores de umidade.** 2017. Disponível em: <https://nexxto.com/sensores-de-umidade-como-funcionam-e-por-que-utiliza-los>. Acesso em: 06 abr. 2019.

PEREIRA, Lúcio C. O.; SILVA, Michel L. **Android para desenvolvedores.** 2. ed. Rio de Janeiro: Brasport, 2012.

REMOTEXY. **Ethernet W5100 shield.** 2019. Disponível em: <http://remotexy.com/en/help/w5100shield/>. Acesso em: 05 nov. 2019.

SILVA, Carlos A. G.; SANTOS, Edson L.; OSINSKI, Cristiano. Wireless Industrial Networks under Interference Conditions based on IEEE 802.15. 4. In: **2018 13th IEEE International Conference on Industry Applications (INDUSCON)**. IEEE, 2018. p. 169-173.

THERMOMATIC. **Como funciona um desumidificador de ar.** 2019. Disponível em: <https://www.thermomatic.com.br/fique-por-dentro/como-funciona-um-desumidificador-de-ar.html>. Acesso em: 20 abr. 2019.