

DESENVOLVIMENTO DE PLATAFORMA COM MÚLTIPLOS EXPERIMENTOS POR MEIO DE LABORATÓRIOS REMOTOS NA ÁREA DE CONTROLE DE PROCESSOS

Júlia Catarina Leme de Oliveira ¹; Wânderson de Oliveira Assis ²

¹ Aluno de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *Este trabalho apresenta uma proposta de desenvolvimento de laboratório remoto destinado a aplicações didáticas em engenharia. Trata-se de uma modernização de aplicações remotas da área de controle de processos, tendo como diferencial o acesso por meio de plataforma online responsiva, podendo ser acessada diretamente utilizando o browser de qualquer dispositivo com acesso à Internet, incluindo computadores e smartphones. Os testes foram realizados com dados obtidos de um sensor e disponibilizados na interface.*

Introdução

O emprego de tecnologias na educação é atualmente mais do que um complemento e um recurso adicional. Trata-se de uma necessidade, que tornou-se ainda mais emergente com a ocorrência da pandemia do COVID-19. Devido à importância das atividades de caráter prático e com a interrupção das aulas presenciais ocorreu uma revolução na forma como utilizamos as tecnologias. Houve um grande aumento da popularidade de aulas por videoconferência e aulas a distância, que passaram a ser realizadas em várias instituições, e se tornou factível graças à publicação da Portaria nº 438 do Ministério da Educação, a qual autorizou a substituição das aulas presenciais por modelos a distância (Ortega e Rocha, 2020).

Dentre as tecnologias que podem ser empregadas na educação, destaca-se a utilização de WebLabs, os laboratórios de acesso remoto. Tratam-se de exemplos claros de utilização de computação na educação, mas com um grande diferencial pois propiciam a possibilidade de aprendizagem real com controle à distância. Isso é possível porque os WebLabs proporcionam a realização de experimentos práticos, acessando remotamente pela Internet equipamentos em laboratórios reais e permitindo seu controle e aquisição de dados em tempo real.

Os WebLabs vêm sendo implantados em diversas instituições desde os anos 90 apresentando soluções para operação remota geralmente utilizando software disponível comercialmente ou redes dedicadas (Fidalgo *et al.*, 2014) (Rodrigues-Gil *et al.*, 2012) (Cruz *et al.*, 2005) (Landin *et al.*, 2015).

No Instituto Mauá de Tecnologia diversos laboratórios remotos foram produzidos, sendo alguns deles direcionados para aplicações de controle de processos (Assis *et al.*, 2013) (Assis *et al.*, 2019) enquanto outros mostram aplicações de Internet das Coisas (IoT) (Assis *et al.*, 2014(a)), aplicações industriais na área petrolífera (Matta *et al.*, 2014), engenharia biomédica (Castagnoli *et al.*, 2015) e robótica com enfoque na atração de alunos para cursos de engenharia (Assis *et al.*, 2014(b)).

Todas estas aplicações utilizam software comercial de elevado custo e exigem do usuário a instalação de um *plugin* num procedimento demorado e que consome considerável espaço da memória. Além disso, a interface apresenta algumas limitações tais como dificuldade para controlar o acesso dos usuários bem como obter o cadastro para identificar o perfil dos mesmos.

Com o custo elevado de softwares de controle que permitem a integração com aplicações controladas pela *web* e suas limitações na questão de adaptabilidade a diferentes dispositivos, bem como suas dificuldades para cadastro de usuários, como descrito anteriormente, surgiu a ideia de desenvolver uma plataforma de iniciativa independente para solucionar essas questões. Além disso, deseja-se que o sistema seja compatível com o *Smart Campus* da Mauá, uma plataforma integrada para desenvolvimento e implementação de soluções de Internet das Coisas e WebLabs.

Nesse sentido, o propósito deste trabalho é desenvolver um sistema que permita a comunicação com sensores e introdução dos sinais numa plataforma de acesso remoto que permita sua adaptação para aplicações de controles.

Para isso foi desenvolvida uma aplicação de teste para monitoramento de sinais de um LDR e utilizando uma placa Arduino para permitir a aquisição desses sinais e a transferência desses dados para um NodeMCU por comunicação I²C. Os dados serão recebidos numa plataforma desenvolvida usando MQTT e disponibilizadas para acesso a partir de um *browser* de Internet.

Adicionalmente a solução desenvolvida permitir o acesso por qualquer dispositivo com acesso à Internet, sendo compatível com qualquer sistema operacional e permitirá ao usuário interagir com o sistema enviado e recebendo dados por meio da interface.

No futuro espera-se que a aplicação possa ser adaptada para aplicações da área de controle de processos podendo ser utilizada como ferramenta para ensino de Engenharia.

Smart Campus Mauá

Smart Campus é uma plataforma do Instituto Mauá de Tecnologia (IMT) para desenvolvimento e implantação de aplicações de Internet das Coisas. Por meio da interface do *Smart Campus* (Figura 1), é possível monitorar e controlar diversas aplicações reais que incluem: sistema de automação para estacionamento (*Smart Parking*), sistema para monitoramento de chuvas intensas, fábrica 4.0 sensorizada, sistema de monitoramento com luz de emergência, câmara climática automatizada, *Smart Plug*, análise comportamental de pets, horta automatizada, dentre outros.

O acesso e monitoramento em tempo real do sistema pode ser feito pelo link <http://smartcampus.maua.br/dash>



Figura 1 – Página com *DashBoards* do *Smart Campus* Mostrando Aplicações Desenvolvidas no IMT

Uma das vertentes do *Smart Campus* é o desenvolvimento de WebLabs, ou seja, laboratório remotos dentro do Campus da Mauá e que podem ser utilizados como plataforma para ensino à distância, mas acessando equipamentos reais instalados nos laboratórios da Instituição.

WebLab de Controle Nível

O laboratório remoto de controle de nível de um tanque utiliza o equipamento *Process Control System* (PCS), ilustrado na Figura 2, é um sistema de controle de fluidos da Festo, fabricante de origem alemã, sendo constituído de um kit didático para simulação de controle de processos. O kit permite implementar o controle do nível, vazão e da temperatura do fluido de trabalho, no caso água destilada.



Figura 2 – Planta de controle de nível PCS Festo

O kit é composto basicamente por duas cubas por onde circula água destilada através de dutos e válvulas, três sensores que realizam as leituras do nível, da temperatura e da vazão, dois atuadores que regulam a temperatura e o fluxo e uma unidade de controle onde, como o próprio nome diz, é realizado todo o controle das variáveis do sistema.

Para desenvolvimento da aplicação de controle de nível utilizar-se-á o sensor de proximidade ultrassônico da Rockwell Automation / Allen Bradley, modelo 873-P-DBAC1-D4. O sensor tem alcance de 100 a 600 mm e produz saída de 4 a 20 mA.

O atuador para o controle de nível é uma bomba centrífuga controlada por tensão que permite bombear água para o tanque superior. A bomba trabalha com tensão de até 24 V e vazão máxima de 10 l/min. A bomba centrífuga é apropriada para a recirculação de água para resfriamento ou em sistemas de aquecimento, podendo ser acionada continuamente.

Para desenvolver o WebLab descrito no trabalho (Assis *et al.*, 2019) no sistema PCS - Festo, o controlador da Bürkert originalmente disponível no equipamento foi desconectado, sendo substituído por um sistema com plataforma microcontrolada (placa Arduino) permitindo que o controle possa ser realizado externamente, por computador.

No projeto uma placa Arduino é utilizada para monitorar o sensor de nível e controlar o acionamento da bomba (utilizando amplificador transistorizado com ponte H). Portanto o diagrama de blocos da aplicação incluindo planta PCS-Festo, placa de condicionamento de sinal, placa NodeMCU, *driver* para acionamento da bomba utilizando sinal PWM (*Pulse Width Modulation*) e interface desenvolvida em computador está ilustrada na Figura 3.

Materiais e Métodos

O projeto consiste em uma modernização do sistema já existente de controle para o WebLab de Controle de Nível, com a criação de uma plataforma *online* para que o acesso possa ser realizado em diferentes dispositivos como computadores, celulares, *tablets* etc.

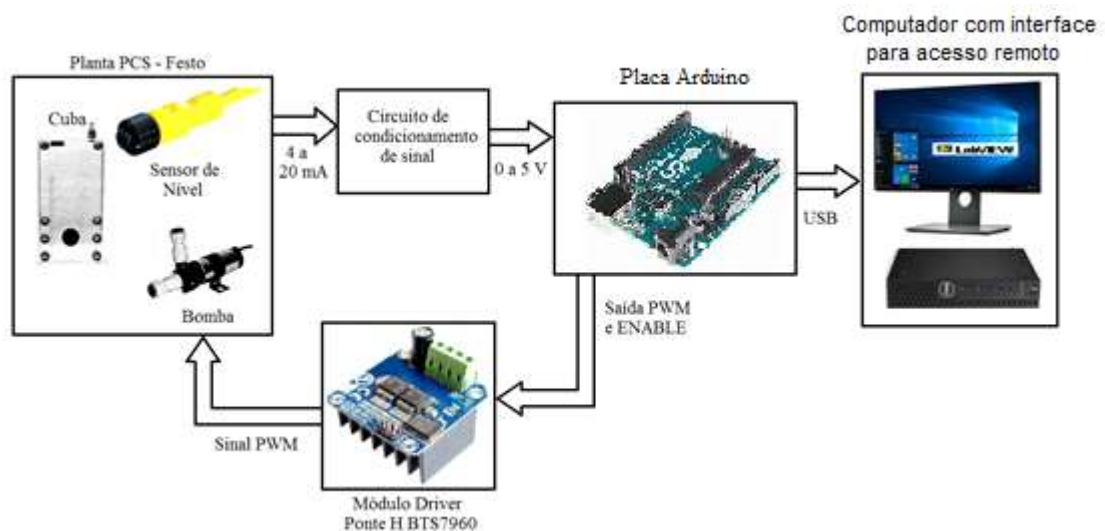


Figura 3 – Diagrama de blocos da aplicação de controle de nível

Como consequência da pandemia e devido à impossibilidade de acesso contínuo aos laboratórios do IMT, devido ao risco de contágio com o COVID-19 e considerando que o principal autor desse projeto pertence ao grupo de risco, optou-se por desenvolver uma solução mais limitada para o projeto, sem o uso efetivo do WebLab de Controle de Nível.

Por isso, o **hardware** do sistema é constituído dos seguintes dispositivos:

- um sensor de temperatura utilizando LDR (*Light Dependent Resistor*), ou seja resistor dependente de luz; a proposta nesse caso é demonstrar a aquisição do sinal do LDR permitindo seu monitoramento pela interface do WebLab, de forma similar ao que seria obtido com o sensor de nível do tanque, ou mesmo com o sensor de temperatura do fluido que pode também ser incorporado ao WebLab;

- uma placa Arduino Uno será utilizada para monitorar os sinais do sensor LDR transmitindo-os para uma placa Node MCU usando comunicação I²C;

- a placa NodeMCU (Hackster.io, 2020) (Electronic Wings, 2020) comunica-se por meio do protocolo MQTT com uma plataforma na Internet; MQTT (MQ Telemetry Transport) é um protocolo de mensagens utilizados para sensores e pequenos dispositivos móveis para redes TCP/IP. As iniciais MQ vêm de uma linha de produtos da IBM chamada “MQ Series”; o *broker* Mosquitto (Eclipse Foundation, 2020) é utilizado para realizar a transferência de dados para o microcontrolador, já que a comunicação com o NodeMCU é bidirecional.

- uma aplicação simples desenvolvido em *JavaScript* recebe os dados da plataforma e do NodeMCU e armazena os dados em no banco de dados elaborado em *PostgreSQL*.

Assim, o diagrama de blocos do hardware da aplicação foi desenvolvido como ilustrado na Figura 4. A comunicação entre o Node-MCU e o Arduino é realizada por meio das conexões mostradas na Figura 5.

O **software** do sistema incluem as seguintes aplicações de programação:

- IDE Arduino – plataforma de desenvolvimento para programação de placas Arduino;
- NodeRed (OpenJS Foundation, 2020) – ferramenta de desenvolvimento baseada no paradigma de programação baseada em fluxo, foi criada pela IBM com a principal função de unir hardware e software e integrar o conceito de Internet das Coisas (IOT); por meio do NodeRed foi possível criar os *dashboards* da aplicação, gráficos com registro histórico, imagens em tempo real do equipamento do laboratório, além de informações do sistema de controle de nível e temperatura, tais como *Set Point* (SP), parâmetros do controlador PID (ganho K_P e constantes de tempo T_I e T_D); por meio da interface será também selecionar o tipo de operação a ser realizada (ensaio em “malha aberta” ou com “controle PID”);

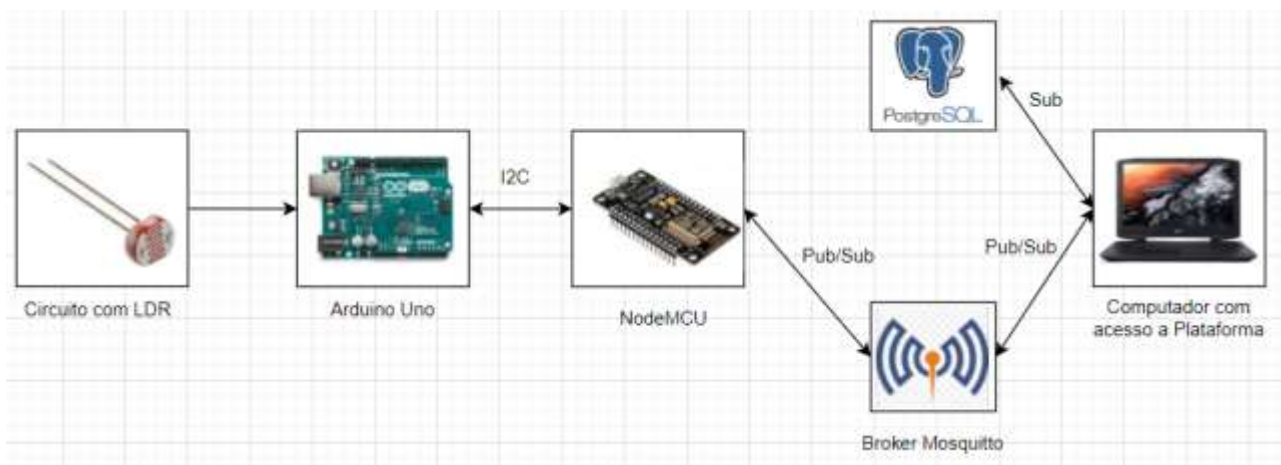


Figura 4 – Diagrama de Blocos do Sistema Desenvolvido

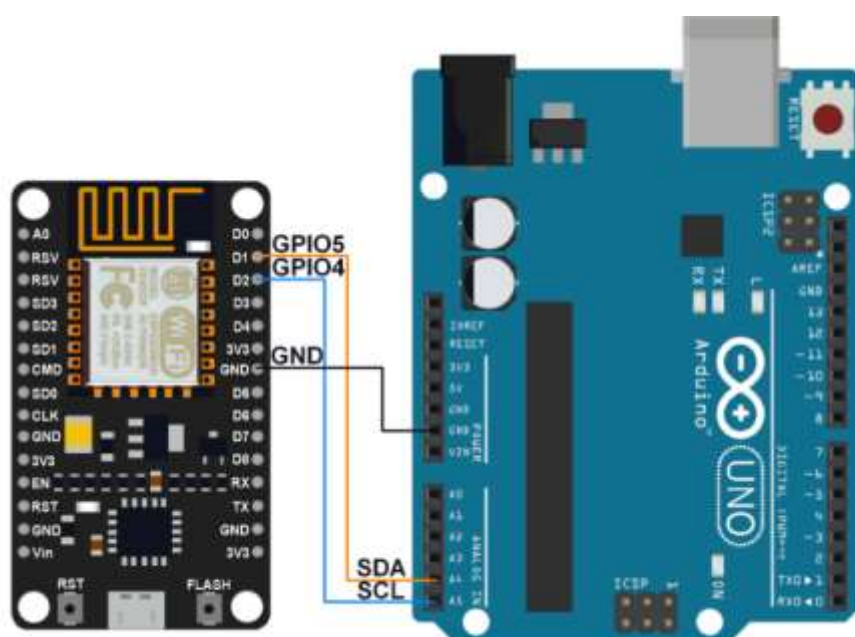


Figura 5 – Conexão I²C NodeMCU-Arduino

- MQTT / Mosquito – é um *broker open source* que implementa o protocolo MQTT para garantir o envio/recebimento de mensagens, ele atua como intermediário no processo de comunicação entre sensores e a plataforma *online*.

- PgAdmin4 – é um gerenciador de banco de dados *open source* para *PostgreSQL*, onde é possível visualizar o banco de dados e gerenciar os dados obtidos.

- Visual Studio Code – plataforma de desenvolvimento para diversas linguagens de programação, utilizado para escrever a aplicação em *JavaScript*.

Resultados e Discussão

A plataforma *online* elaborada em Node-Red pode ser vista na Figura 6, onde há diversos campos de interação com o usuário. No campo “Câmera” será possível visualizar uma imagem em tempo real do laboratório em funcionamento, ainda não implementada pela ausência da câmera. Tratando-se do campo “Seleção de Controle” é possível selecionar a variável para a qual desejamos configurar o controle no momento, podendo ser Nível ou Temperatura, onde podem ser inseridos valores de SP (Set Point), K_P (ganho proporcional), T_I (constante de tempo de integração)

e TC (constante de tempo de derivação T_D) para o controle selecionado, como pode ser visualizado nos campos inferiores direito. Em “Seleção de ensaio” pode-se optar pelo cálculo do controle PID (não implementado), “Malha Aberta” ou “Curva Sensor” onde é possível visualizar um gráfico com os valores obtidos do *LDR* em função do tempo no campo “Chart”. No campo “Imagem do Tanque” será possível visualizar melhor os parâmetros atuais do tanque como nível e temperatura da água (não implementado).

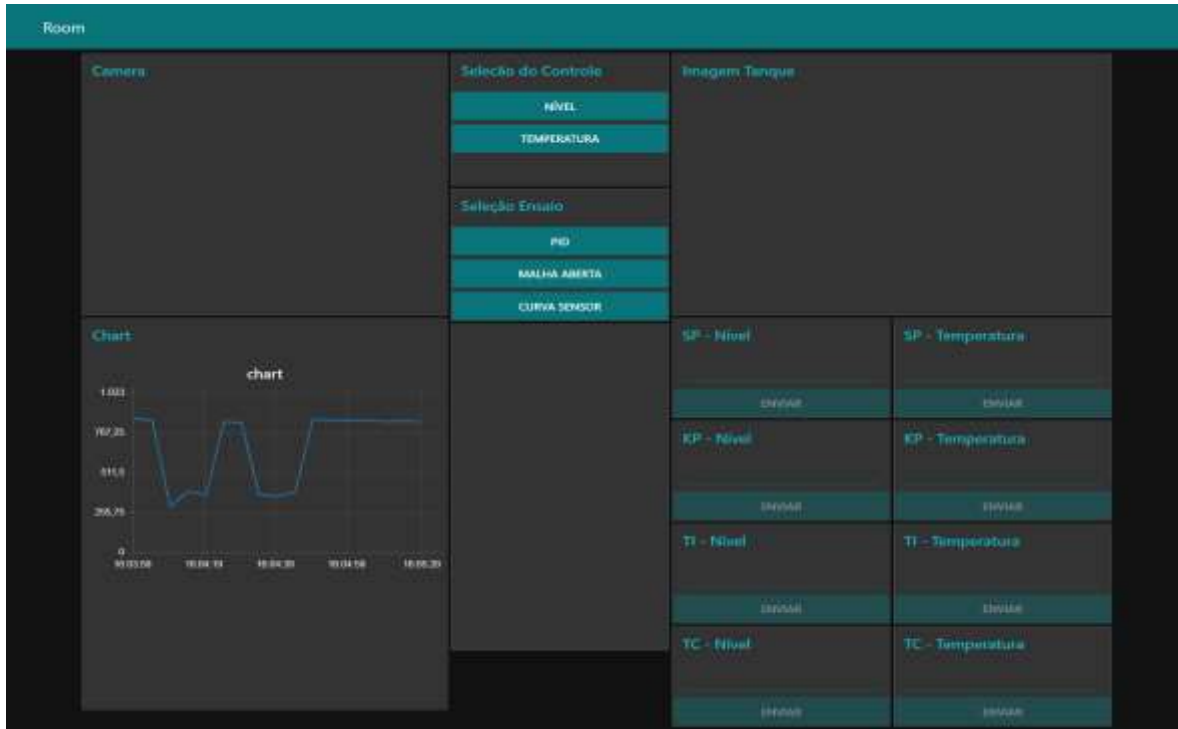


Figura 6 – Plataforma *Online*

Os dados inseridos e selecionados na plataforma são enviados ao *NodeMCU* por meio da comunicação *MQTT*, os dados também são direcionados para a aplicação *JavaScript* que envia os dados para um banco de dados em *PostgreSQL* onde eles podem ser armazenados para consultas futuras. Os dados no banco podem ser visualizados pelo gerenciador de banco de dados do *PgAdmin 4*, como apresentado na Figura 7.

	Data Output	Explain	Messages	Notifications
	dado character varying (20)	valor double precision	id integer	
7	room/kplv		30	7
8	room/splv		96	8
9	room/splv		96	9
10	room/splv		96	10
11	room/splv		96	11
12	room/chart		27	12

Figura 7 – Dados no *PgAdmin4*

Os dados enviados ao *NodeMCU* podem ser vistos no monitor serial e são enviados diretamente ao *Arduino Uno* por meio de comunicação I²C (Figura 8). Não foram registrados atrasos significativos na transmissão dos dados nos dois protocolos. Os valores do *LDR* que são lidos pelo *Arduino Uno* são enviados ao *NodeMCU* por meio da I²C, e enviados à plataforma por meio do *MQTT* pelo *NodeMCU*, onde os valores são inseridos no gráfico. Um esquemático do funcionamento da aplicação pode ser visto na Figura 9.

```

19:43:26.578 -> Conectando a rede aa
19:43:27.120 -> .....
19:43:34.987 -> WiFi conectado! - ESP8266 IP: 192.168.1.44
19:43:35.022 -> Tentando conectar ao MQTT...Conectado!
23:00:00.447 -> Mensagem no tópico: room/ensaio. Mensagem: Curva Sensor
23:00:00.481 -> 834
23:00:05.442 -> 815
23:00:10.436 -> 819
23:00:15.447 -> 101
23:00:20.430 -> 119
23:00:25.435 -> 816
23:00:30.440 -> 830
23:00:35.447 -> 811
23:00:40.469 -> 831
23:00:45.469 -> 811
23:00:50.456 -> 824
23:00:55.479 -> Mensagem no tópico: room/splv. Mensagem: 100
23:00:55.512 -> Mensagem no tópico: room/led. Mensagem: Nível
23:01:00.675 -> Mensagem no tópico: room/ensaio. Mensagem: Malha Aberta

```

Figura 8 – Monitor Serial NodeMCU.

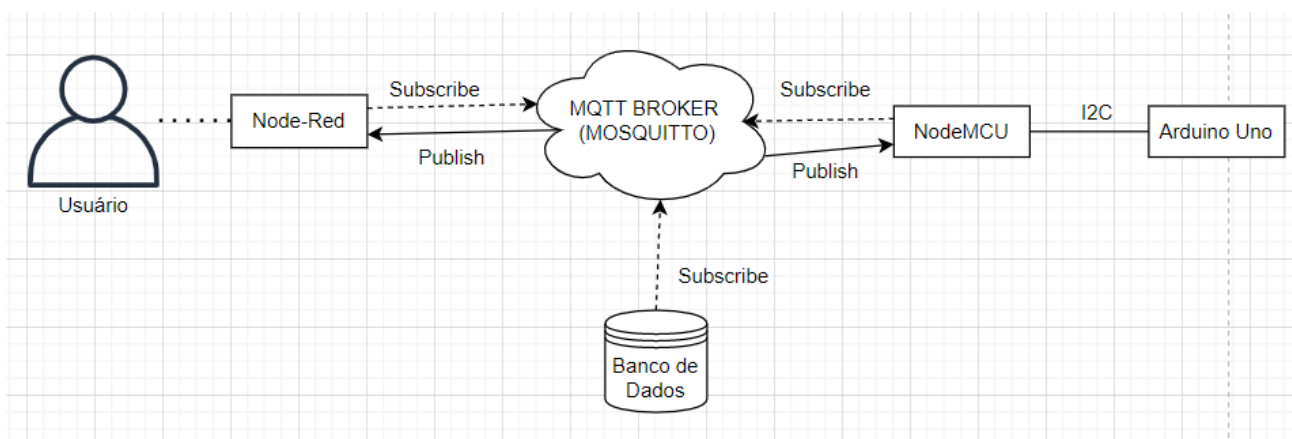


Figura 9 – Esquemático do Sistema de Comunicação Desenvolvido

Para continuação do projeto serão adicionadas as funcionalidades das opções de PID no qual é realizada a implementação e cálculo do controlador, e realizados ensaios em “Malha Aberta”, visando realizar a modelagem do sistema. Adicionalmente será realizada a exibição da imagem na câmera e imagem do tanque com uma representação dos valores de nível e temperatura, melhoria do gráfico e adição de novos parâmetros.

Conclusões

O projeto em sua versão atual garante comunicação *MQTT* entre a plataforma *online* em Node-Red, com possibilidade de envio e recebimento de dados para o Arduino por meio da comunicação I²C com o NodeMCU. Os tempos de resposta de transmissão são rápidos, porém com um aumento muito grande no fluxo de dados alguns atrasos podem ser visíveis, o principal atraso acontece na atualização do gráfico com os valores do sensor *LDR*; como a comunicação com a plataforma necessita de internet, pode ser um problema para conexões mais lentas. Tratando-se da plataforma do Node-Red, observou-se que possui uma limitação quando o assunto é customização da página. Por padrão são poucas as opções de mudança e a aplicação permite apenas código *CSS* para criar uma personalização, mas não oferece o nome de classes e identificações *HTML* para tornar a alteração viável. A aplicação em banco de dados está em um código a parte por limitações no Node-Red para acesso ao banco de dados *PostgreSQL*. Por ser uma plataforma muito simples é necessária a instalação de diversas bibliotecas a parte para garantir algumas funcionalidades, porém nem todas são funcionais.

Para continuação do projeto é recomendado uma análise para adicionar a aplicação de banco de dados ao sistema. Como o *Node-Red* não permite personalização é viável verificar possíveis melhorias nesse quesito bem como buscar por bibliotecas que aprimorem a solução. Para melhor implementação do sistema de controle com controlador PID ou realização de ensaio em “Malha Aberta” é recomendado a continuação do trabalho por um aluno que tenha conhecimento prévio na área de controle de processos.

Referências Bibliográficas

Assis, W. O., Coelho, A. D., Júnior, J. C. S., Kunigk, L., Morais, A. S., Lemos, G. M., Gedraite, E. S., Takahashi, E. K., Gedraite, R. **“WebLab for Control Applications in Engineering Education”** WBE 2013 - The 10th IASTED International Conference on Web-Based Education. **Anais**. Innsbruck, Áustria, 2013.

Assis, W. O.; Coelho, A. D.; Mathias, M. A.; Matta, E. N.; Gedraite, R. **“A Practical Approach to Teaching During the First Grade of Electronic Engineering Course – A Weather Station With Wireless Transmission and Interface by Web”**, ICSE 2014 – 2nd International Congress of Science Education, Foz do Iguaçu, PR, 2014(a).

Assis, W. O.; Coelho, A. D.; Pacheco, V. G.; Gedraite, R.; Pires, R. C.; Matta, E. N. **“Applying the Concept of WebLab in a Playful Way - The Electronic Engineering Hands On”**. ICSE 2014 – 2nd International Congress of Science Education, Foz do Iguaçu, PR, 2014(b).

Assis, W. O., Gonçalves, H. S. B., Silva, G. T., Coelho, A. D., **“Uma Aplicação de Controle de Nível com Acesso Remoto pela Web”** COBENGE 2019 - XLVII Congresso Brasileiro de Educação em Engenharia. **Anais**. Fortaleza, CE, Brasil, 2019.

Castagnoli, G., Toyota, L. T. S., Veríssimo, R. R., Theodoro, V. O. **“Tecnologia Assistiva – Estação Health Care para Idosos”**. Trabalho de Conclusão de Curso – Engenharia de Controle e Automação, 2015.

Cruz, A. J. G.; Jesus, C. D. F.; Giordano, R. C.; Nascimento, C. A. O.; Roux, G. A. C. L.; Loureiro, L. V. Loureiro **“Experimento de Controle de Nível Operado Remotamente Via Internet: Projeto WebLab”** XXXIII Congresso Brasileiro de Ensino de Engenharia, Campina Grande, PB, Brasil, v. 1, pp. 1-10, 2005.

Eclipse Foundation “**Eclipse Mosquitto – An Open Source MQTT Broker**”. Disponível em: <https://mosquitto.org/> Acesso em 27 de novembro de 2020.

Electronic Wings “**NodeMCU MQTT Client with Arduino IDE**” Disponível em: <https://www.electronicwings.com/nodemcu/nodemcu-mqtt-client-with-arduino-ide>

Fidalgo, A. V.; Alves, G. R.; Marques, M. A.; Viegas, M. C.; Costa-Lobo, M.C.; Hernandez-Jayo, U.; Garcia-Zubia J.; Gustavvsson, I. “**Adapting Remote Labs to Learning Scenarios: Case Studies Using VISIR and RemotElectLab**”, IEEE Revista Iberoamericana de Tecnologias del Aprendizage, 9(1), pp. 33-39, 2014.

Hakster.io “**Getting Start w/ NodeMCU ESP8266 on Arduino IDE**” Disponível em: <https://create.arduino.cc/projecthub/electropeak/getting-started-w-nodemcu-esp8266-on-arduino-ide-28184f>. Acesso em 27 de novembro de 2020.

Landin, P. N.; Gustafsson, S.; Fager, C.; Eriksson T. “**WebLab: A Web-Based Setup for PA Digital Predistortion and Characterization**”, IEEE Microwave Magazine, Vol. 16, I. 1, pp. 138-140, Feb., 2015.

Matta, E. N., Maranhão, G. F. G., Ortiz, R. S. L., Assis, W. O., Gedraite, R., Coelho, A. D., Kunigk, L. “**Caracterização de Partículas em Fluido Utilizado na Perfuração de Poços de Petróleo**”. CBA 2014 – 20º Congresso Brasileiro de Automática, Belo Horizonte, MG, 2014.

OpenJS Foundation “**Node-Red – Low-code Programming for Event-Driven Applications**” Node-Red. Disponível em: <https://nodered.org/>

Ortega, L. M. R.; Rocha, V. F. “**O Dia Depois de Amanhã – Na Realidade e Nas Mentes – O Que Esperar da Escola Pós-Pandemia?**” Pedagogia em Ação, v. 13, n° 1, p.302 – 314, Belo Horizonte – MG: Revista Eletrônica do Curso de Pedagogia da PUC Minas, 2020.

Rodrigues-Gil, L.; Orduña, P.; García-Zubia, J.; Lopes-de-Ipiña D. “**Advanced Integration of OpenLabs VISIR (Virtual Instrument Systems in Reality) with Weblab-Deusto**”, 2012 9th International Conference on Remote Engineering and Virtual Instrumentation, Bilbao, Spain, 2012.