

Campus Porto Velho Calama
Coordenação do Curso de Engenharia de Controle e Automação

ALFREDO JUSTINIANO PAES
THIAGO LIMA BARBOSA

**PROTOCOLO MODBUS COM ARDUINO VIA SCADABR COMO ALTERNATIVA
DIDÁTICA PARA O ENSINO DE ENGENHARIA**

PORTO VELHO
2025

**ALFREDO JUSTINIANO PAES
THIAGO LIMA BARBOSA**

**PROTOCOLO MODBUS COM ARDUINO VIA SCADABR COMO ALTERNATIVA
DIDÁTICA PARA O ENSINO DE ENGENHARIA**

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus Porto Velho Calama*, como requisito parcial para obtenção do grau de bacharel, junto ao Curso de Engenharia de Controle e Automação, sob a orientação do professor Me. Kariston Dias Alves.

PORTO VELHO
2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Paes, Alfredo Justiniano.

Protocolo Modbus com Arduino via Scadabr como alternativa didática para o ensino de engenharia / Alfredo Justiniano Paes, Thiago Lima Barbosa. - Porto Velho, 2025.

32 f. : il.

Orientador(a): Prof. Dr. Kariston Dias Alves.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho, 2025.

ISBN 1678-0817

1. ScadaBR. 2. Arduino. 3. Modbus. 4. Supervisório. I. Barbosa, Thiago Lima. II. Alves, Kariston Dias (orient.). III. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. IV. Título.

Bibliotecário(a) Responsável: Miria Santana Veiga, CRB-11/898

ALFREDO JUSTINIANO PAES

THIAGO LIMA BARBOSA

**PROTOCOLO MODBUS COM ARDUINO VIA SCADABR COMO ALTERNATIVA
DIDÁTICA PARA O ENSINO DE ENGENHARIA**

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus Porto Velho Calama*, como requisito parcial para obtenção do grau de bacharel, junto ao Curso de Engenharia de Controle e Automação, sob a orientação do professor Me. Kariston Dias Alves.

Aprovado em: 28/11/2025 pela banca examinadora.

Eduardo Araujo de Sousa
Coordenador

Kariston Dias Alves
Orientador

Rafael Pissinati de Souza
Coorientador

PROTOCOLO MODBUS COM ARDUINO VIA SCADABR COMO ALTERNATIVA DIDÁTICA PARA O ENSINO DE ENGENHARIA

Engenharias, Volume 29 - Edição 151/OUT 2025 / 24/10/2025

MODBUS PROTOCOL WITH ARDUINO VIA SCADABR AS A DIDACTIC ALTERNATIVE FOR ENGINEERING TEACHING

REGISTRO DOI: 10.69849/revistaft/ra10202510241412

Alfredo Justiniano Paes¹

Thiago Lima Barbosa²

Kariston Dias Alves³

Rafael Pissinati de Souza⁴

Resumo

Este trabalho apresenta uma proposta como alternativa didática para o ensino de redes industriais em cursos de engenharia voltadas para automação, com o uso de plataformas livres e de código aberto, integrando o software supervisor ScadaBR com o microcontrolador Arduino UNO e aplicando o protocolo Modbus. O objetivo foi criar uma alternativa prática, acessível e replicável que permitisse aos alunos compreender a comunicação mestre-escravo e a troca de dados em sistemas SCADA. A abordagem incluiu pesquisas bibliográficas e a criação

de um protótipo experimental capaz de monitorar e gerenciar LEDs e sensores usando registradores Modbus. O sistema implementado possibilitou uma comunicação estável entre o Arduino e o ScadaBR, permitindo a leitura e a escrita em tempo real de variáveis analógicas e digitais. Os resultados mostram que a combinação de hardware aberto e ferramentas de código livre é eficiente para superar as limitações muitas vezes encontradas em laboratórios acadêmicos, facilitando o aprendizado prático de conceitos de automação industrial e redes de comunicação.

Palavras-chave: ScadaBR, Arduino, Modbus, supervisorio.

Abstract

This paper presents a proposal as a didactic alternative for teaching industrial networks in automation-focused engineering courses, using free and open-source platforms, integrating the ScadaBR supervisory software with the Arduino UNO microcontroller and applying the Modbus protocol. The objective was to create a practical, accessible, and replicable alternative that would allow students to understand master-slave communication and data exchange in SCADA systems. The approach included literature research and the creation of an experimental prototype capable of monitoring and managing LEDs and sensors using Modbus registers. The implemented system enabled stable communication between the Arduino and ScadaBR, allowing real-time reading and writing of analog and digital variables. The results show that the combination of open hardware and open-source tools is effective in overcoming the limitations often encountered in academic laboratories, facilitating practical learning of industrial automation and communication network concepts.

Keywords: ScadaBR, Arduino, Modbus, supervisor.

1 INTRODUÇÃO

O desenvolvimento da Indústria 4.0 no mundo, devido à sua capacidade de solucionar os desafios globais de sistemas produtivos como a eficiência energética e de recursos, tem consolidado as redes industriais como pilar da automação industrial. De acordo com Kagermann (2013), o avanço da Indústria 4.0 permite ganhos contínuos de produtividade e eficiência de recursos em toda a rede de valor. Para que o controle do processo industrial ocorra com excelência, a arquitetura da rede industrial deve garantir que os dados trafeguem com confiabilidade e agilidade dos elementos que fazem esse controle, como sensores, atuadores, válvulas e chaves, até o nível de informação gerencial, representado pelo sistema supervisório (LUGLI; SANTOS, 2019). Os protocolos de comunicação são os responsáveis por estabelecer os parâmetros de troca dados entre os equipamentos, de forma ordenada, evitando erros e informando cada acontecimento (SENAI, 2014). Neste contexto, compreender a arquitetura e o funcionamento dos protocolos de comunicação é indispensável para profissionais das engenharias voltadas para automação, não somente na teoria, mas principalmente na prática.

No cenário acadêmico, a realização de atividades práticas usando equipamentos industriais, muitas vezes é limitada, devido ao custo elevado de controladores, bancadas insuficientes em relação ao número de discentes, falta de manutenção por fatores burocráticos e a indisponibilidade licenças de softwares supervisórios. Tais limitações tornam o ensino prático de redes industriais um desafio para instituições de ensino, gerando uma lacuna entre o conhecimento teórico adquirido em sala de aula e a vivência prática, restringindo a consolidação do conhecimento.

O uso de plataformas abertas tem se mostrado eficaz para suprir essa lacuna, permitindo a realizações de simulações para complementar ensino teórico com experiências práticas. Moromisato et al. (2007), afirmam que, os programas livres estão em franca expansão e são uma solução para esse problema. Entre as plataformas livres, o Arduino foi optado como microcontrolador para este trabalho, devido à sua

simplicidade de programação, ampla documentação e a capacidade de integração com diversos sensores e atuadores (ARTERO, 2013).

De acordo com Nascimento e Lucena (2003), no campo das redes industriais, o protocolo Modbus talvez possui a mais larga escala de utilização, isto se deve a sua simplicidade e facilidade de implementação, além de ser o preferido de diversos controladores e ferramentas para o desenvolvimento de sistemas supervisórios. Essas características do Modbus, tornam o protocolo uma ferramenta não apenas aplicável em sistemas industriais, mas o consagra como um excelente recurso pedagógico.

Uma alternativa à softwares supervisórios proprietários, como Elipse E3, Wonderware Intouch e iFIX, que são amplamente utilizados na indústria, mas que, devido ao alto custo, dificultam aplicações em ambientes acadêmicos, apresenta-se o ScadaBR que possui código aberto, multiplataforma e compatível com protocolos mais utilizados (SCADABR, 2025).

Contudo este trabalho visa apresentar uma proposta para o ensino de redes industriais, que proporciona uma experiência prática aos alunos dos cursos de engenharia, utilizando plataformas livres, um Arduíno UNO configurado como escravo Modbus e o ScadaBR como sistema supervisório, consolidando conceitos de comunicação mestre-escravo, troca de dados entre registradores e interação em sistemas SCADA.

Dessa forma, o presente artigo busca responder a seguinte questão: É possível viabilizar o ensino de redes industriais de maneira acessível, eficaz e replicável, utilizando plataformas abertas? Alinhando-se as necessidades da formação de profissionais preparados e com bagagem prática para os desafios da Indústria 4.0, a proposta desenvolvida insere-se no contexto de iniciativas que buscam democratizar o acesso ao ensino prático de automação.

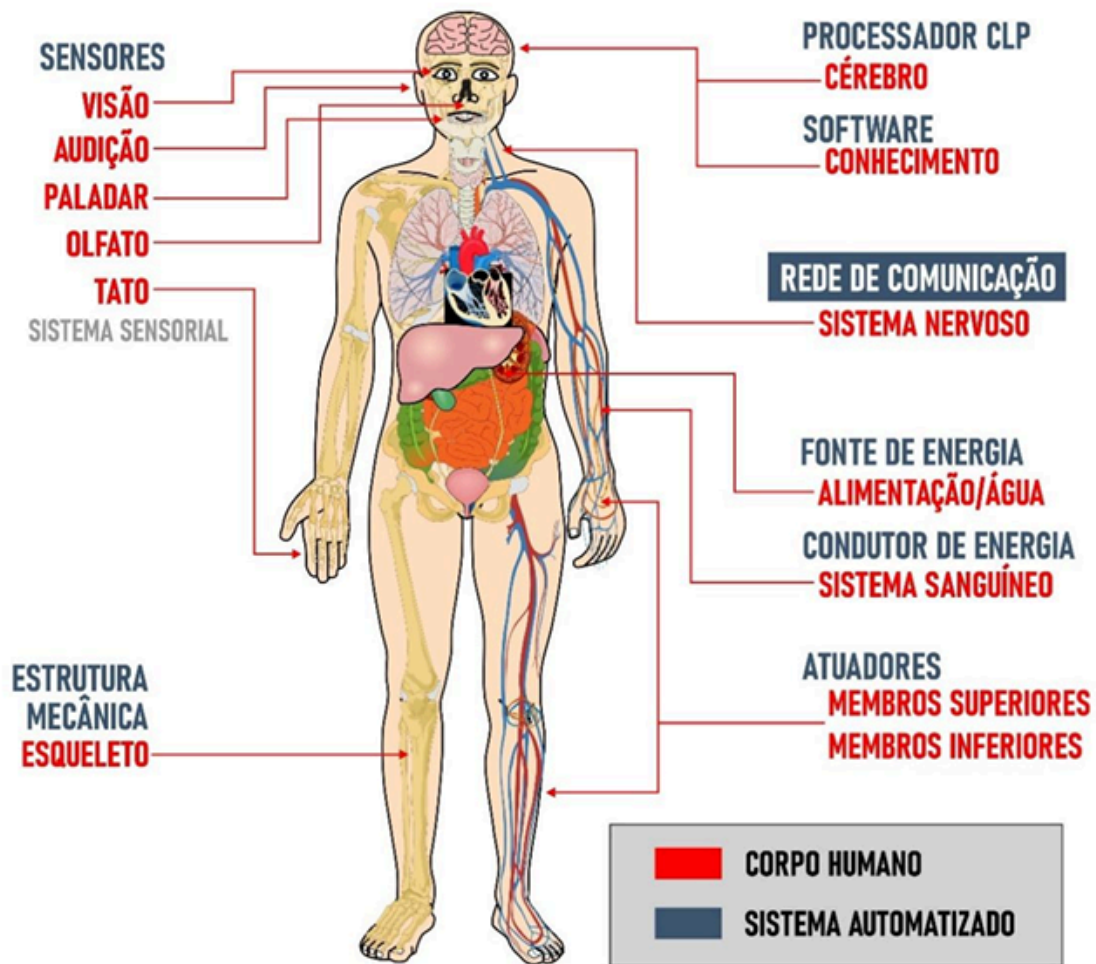
2 FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento da automação e do controle de processos no contexto da Indústria 4.0 está diretamente ligado ao progresso das redes industriais e dos sistemas supervisórios. Nesta seção, abordam-se conceitos fundamentais que embasam o desenvolvimento deste trabalho: redes industriais, protocolo Modbus, sistemas SCADA e a relevância de utilizar ferramentas de código aberto, como o ScadaBR, para o ensino prático de engenharia.

2.1 Redes industriais e a Indústria 4.0

De acordo com a Cleveland Clinic (2023), o sistema nervoso do corpo humano tem como principal função enviar e receber mensagens ou sinais elétricos entre o cérebro e todas as partes do corpo, transmitindo comandos como andar, falar ou ver, por exemplo. Com base nesse conceito, pode-se afirmar que as redes industriais, dentro de um processo automatizado, possuem função semelhante ao sistema de comunicação do corpo humano, evidenciando sua importância fundamental para a eficiência dos sistemas produtivos e o avanço da indústria 4.0. Na figura 1, ilustra-se uma comparação entre o sistema de comunicação do corpo humano e de um sistema automatizado.

Figura 1 – Corpo Humano/Sistema automatizado.



Fonte: Elaborado pelos autores.

Conforme Lugli e Santos (2019), as redes industriais são estruturas de comunicação projetadas para conectar os elementos responsáveis pelo controle do processo industrial, com sensores e atuadores, efetuando o tráfego de dados até o nível de informação gerencial.

Com o surgimento da Indústria 4.0, a importância das redes industriais tornou-se ainda mais evidente. O relatório da National Academy of Science and Engineering – Acatech (2013), afirma que redes de comunicação confiáveis e de alta qualidade são um requisito fundamental para a Indústria 4.0. Para Kagermann (2013), no mesmo relatório, a Indústria 4.0 possibilita fabricar itens únicos com critérios individuais para um único cliente e mesmo assim, obter lucro, tudo isso, proporcionando eficiência de recursos, gastando menos matéria-prima e energia.

Nesse contexto, proporcionar o domínio dessas tecnologias aos engenheiros de automação, representa não somente uma evolução industrial, mas também uma solução ambiental, promovendo eficiência energética e o uso racional de recursos naturais.

2.2 O protocolo Modbus

Criado pela Modicon em 1979 que posteriormente foi adquirida pela Schneider Electric, teve os direitos liberados pela Modbus Organization, o Modbus destaca-se entre os protocolos de comunicação industrial devido a sua simplicidade e facilidade de implementação (FREITAS, 2014).

Segundo a Modbus Organization, o Modbus é ideal para fins educacionais, por se tratar de um protocolo de comunicação aberto, possuir padronização e transparência, sendo propício para uso em pesquisas e desenvolvimento de protótipos de automação, permitindo compreender princípios de endereçamento, leitura e escrita de registradores. Essas características possibilitam sua implementação em plataformas de baixo custo como o Arduino.

2.3 Sistemas SCADA

Os sistemas SCADA (Supervisory Control and Data Acquisition) são ferramentas voltadas à supervisão e controle de processos industriais. Permitem a obtenção de dados em tempo real, visualização gráfica, geração de relatórios e o acionamento remoto de equipamentos (BOYER, 2010).

Sistemas SCADA proprietários como iFIX, Wonderware e Elipse E3, este último líder no mercado brasileiro, são amplamente utilizados, porém possuem licenças de custo elevado, o que dificulta a adoção em ambientes educacionais (ELIPSE SOFTWARE, 2025). A falta de acesso a estes softwares reduz a possibilidade de vivenciar situações reais de supervisão pelos alunos.

Por outro lado, o ScadaBR é um software supervisor de código aberto e compatível com os principais protocolos industriais, incluindo o Modbus (SCADABR Manual, 2025). Uma ferramenta multiplataforma, extensível e gratuito, que facilita sua aplicação em projetos acadêmicos.

2.4 ScadaBR como ferramenta de ensino

Desenvolvido com recursos públicos com o apoio da FINEP, SEBRAE e CNPq, o ScadaBR é um Software livre, gratuito e de código aberto, para desenvolvimento de sistemas supervisórios, automação de equipamentos, aquisição de dados e controle remoto (SCADABR, 2025).

Conforme destaca Silva et al. (2013), o Software livre é algo extremamente viável para o ensino, permitindo que todos tenham acesso ao conhecimento científico e tecnológico, assim as instituições de ensino utilizam estas ferramentas para descentralizar o conteúdo das disciplinas ofertadas.

O ScadaBR é uma das ferramentas livres com funcionalidades semelhantes às plataformas proprietárias, podendo ser utilizada por docentes para ensinar seus alunos sobre sistemas supervisórios de controle e aquisição de dados (SILVA et al. 2013). Estudos anteriores demonstram a eficácia do ScadaBR como ferramenta didática no desenvolvimento de sistemas SCADA, como o demonstrado por Carneiro (2021).

2.5 Ensino e aprendizagem baseados em prática

Ao longo dos anos a literatura dedicada à formação em engenharia tem mostrado que a aprendizagem é significativamente reforçada por meio da prática experimental. Andrews no livro (Hands-On, Engineering, 2019), pontua que expor os alunos a atividades hands-on, ou seja, atividades “mão à obra”, em tradução direta, proporciona engajamento, senso de autoria e retenção dos conceitos.

Na visão de Steele no livro (Experiential Learning in Engineering Education, 2023), os benefícios da aprendizagem experiencial são muitos, desde o aumento da aprendizagem ao desenvolvimento do indivíduo que pode se tornar um profissional eficaz e autodidata. Gibbs (2013), aponta que não basta apenas fazer, também não basta apenas pensar. Aprender com a experiência deve envolver conexões entre o fazer e o pensar. Teoria e prática devem caminhar juntas e ordenadas, para construção de um conhecimento sólido e duradouro.

Na literatura Brasileira, a obra “Engenharia na Prática: Ensino, pesquisa e aplicações” (ATENA EDITORA, 2020) corrobora com essa visão, destacando que a prática experimental é essencial para a formação do engenheiro e para o desenvolvimento da capacidade crítica e criativa na solução de problemas reais.

Contudo, integrando o hardware (Arduino), protocolos industriais (como o Modbus) e ferramentas de supervisão (ScadaBR) neste trabalho, representa não apenas uma alternativa tecnológica, mas também uma estratégia pedagógica coerente de educação em engenharia.

3 METODOLOGIA

O desenvolvimento deste projeto foi baseado em pesquisa bibliográfica, incluindo artigos científicos, dissertações, normas e manuais, com o objetivo de demonstrar a viabilidade de implementar uma experiência prática de Redes Industriais para alunos do curso de Engenharia de Controle e Automação, com a integração de plataformas abertas e acessíveis e componentes eletrônicos simples, desta forma desenvolver um sistema de monitoramento e controle de LEDs utilizando um microcontrolador Arduino UNO, com comunicação via protocolo Modbus e toda interface gráfica no ScadaBR.

3.1 Funcionamento do sistema

O sistema criado inclui um Arduino que funciona como escravo Modbus, encarregado de gerenciar três LEDs: um no pino 12, outro no pino 13 e o LED da porta 13 do próprio Arduino. Além disso, ele monitora o estado de um botão que ativará um dos LEDs e lê um potenciômetro que ajustará a intensidade de iluminação do segundo LED, utilizando uma conexão por meio do “Point link”. Como interface gráfica para interação com o usuário, foi empregado o ScadaBR, possibilitando o controle do LED no pino 13 e a supervisão do estado de todos os componentes. A seguir são descritos os principais objetivos do sistema.

- Implementar a comunicação Modbus entre o Arduino e o ScadaBR.
- Controlar o estado de um LED (pino 13) diretamente pela interface do ScadaBR.
- Ler um potenciômetro e permitir que faça o controle de um dos LEDs.
- Monitorar o estado de um LED (pino 12) e exibi-lo na interface do ScadaBR.
- Implementar um botão físico para alternar o estado do LED no pino 12, com debounce para evitar leituras incorretas.

3.2 Materiais utilizados

O projeto contou com o mínimo de componentes necessários, contudo, que proporcione uma experiência significativa de um sistema de controle e supervisão, envolvendo entradas e saídas tanto analógicas como digitais, podendo ser ampliado conforme necessidade. A tabela 1 mostra a lista de materiais utilizados.

Tabela 1 – Hardware utilizado.

HARDWARE	QUANTIDADE
Arduino Uno R3	1
LED Vermelho	1
LED Verde	1
Potenciômetro 10K Ω	1
Push Boton	1
Resistor 10K Ω	1
Resistor 330 Ω	1
Resistor 150 Ω	1
Cabo USB - B	1

Fonte: Autores.

O ScadaBR foi executado dentro do ambiente Eclipse Temurin. A tabela 2 apresenta os softwares utilizado no projeto.

Tabela 2 – Software utilizado.

SOFTWARE
ScadaBR / Eclipse Temurin
Biblioteca “SimpleModbusSlave” para Arduino

Fonte: Autores.

3.3 Desenvolvimento do código Arduino

O código foi desenvolvido utilizando a biblioteca SimpleModbusSlave, que facilita a implementação do protocolo Modbus no Arduino. O Arduino foi configurado como um dispositivo escravo Modbus, com registradores para armazenar o valor do potenciômetro, o estado do botão e os estados

dos LEDs. Na figura 2 podemos observar a inclusão da biblioteca mencionada.

Figura 2 – Inclusão da biblioteca SimpleModbusSlave.

```
#include <SimpleModbusSlave.h>
```

Fonte: Autores.

Na sequência, foram definidos os registradores Holding Registers, responsáveis por armazenar os dados trocados entre o Arduíno e o ScadaBR. Na figura 3 estão listados todos os Holding Registers usados no projeto.

Figura 3 – Definição dos Holding Registers.

```
enum
{
    POTENCIOMETRO_VAL,    // Valor do potenciômetro (entrada analógica)
    PWM_VAL,              // Valor do PWM do LED 1
    PUSH_BUTTON_STATE,   // Estado do push button
    LED2_STATE,           // Estado do LED 2
    LED13_STATE,         // Estado do LED 13
    HOLDING_REGS_SIZE     // Tamanho total dos registradores
};

unsigned int holdingRegs[HOLDING_REGS_SIZE]; // Array de registradores Modbus
```

Fonte: Autores.

Dentro do setup (), foi configurado o protocolo Modbus Serial, conforme demonstrado na figura 4.

Figura 4 – Configuração protocolo Modbus.

```
void setup()
{
  modbus_configure(&Serial, 9600, SERIAL_8N1, 1, 2, HOLDING_REGS_SIZE, holdingRegs);
  modbus_update_comms(9600, SERIAL_8N1, 1);
}
```

Fonte: Autores.

O potenciômetro é lido pelo Arduíno e seu valor é armazenado no registrador Holding Register 0, para que o ScadaBR possa monitorá-lo. Na figura 5 podemos ver esse comando.

Figura 5 – Leitura do potenciômetro.

```
holdingRegs[POTENCIOMETRO_VAL] = analogRead(potPin);
```

Fonte: Autores.

No Holding Register 2 é lido e armazenado no estado do Push Button, permitindo que seja monitorado pelo ScadaBR. A figura 6 apresenta essa função.

Figura 6 – Leitura do push button.

```
holdingRegs[PUSH_BUTTON_STATE] = digitalRead(buttonPin) == HIGH ? 1 : 0;
```

Fonte: Autores

O botão funciona como um interruptor (toggle switch) para o LED2, alternando entre ligado e desligado a cada aberto. Na figura 7 podemos visualizar a parte do código que atualiza o valor do LED2.

Figura 7 – Atualiza o valor do LED2.

```

if (digitalRead(buttonPin) == HIGH) {
  if (!buttonPressed && (millis() - lastDebounceTime) > debounceDelay) {
    led2Status = !led2Status; // Alterna o estado do LED 2
    holdingRegs[LED2_STATE] = led2Status; // Atualiza registrador Modbus
    buttonPressed = true;
    lastDebounceTime = millis();
  }
} else {
  buttonPressed = false;
}

digitalWrite(led2Pin, led2Status);

```

Fonte: Autores.

Por fim, o código atualiza constantemente os Holding Registers, permitindo que o ScadaBR leia escreva e mostre dados na interface gráfica em tempo real. As linhas do código responsáveis por essa função podem ser analisadas na figura 8.

Figura 8 – Atualização dos valores nos Holding Registers.

```

holdingRegs[POTENCIOMETRO_VAL] = analogRead(potPin);
analogWrite(led1Pin, holdingRegs[PWM_VAL]);
holdingRegs[PUSH_BUTTON_STATE] = digitalRead(buttonPin) == HIGH ? 1 : 0;
holdingRegs[LED2_STATE] = led2Status;
digitalWrite(led13Pin, holdingRegs[LED13_STATE] == 1 ? HIGH : LOW);

```

Fonte: Autores.

3.4 Principais funcionalidades implementadas:

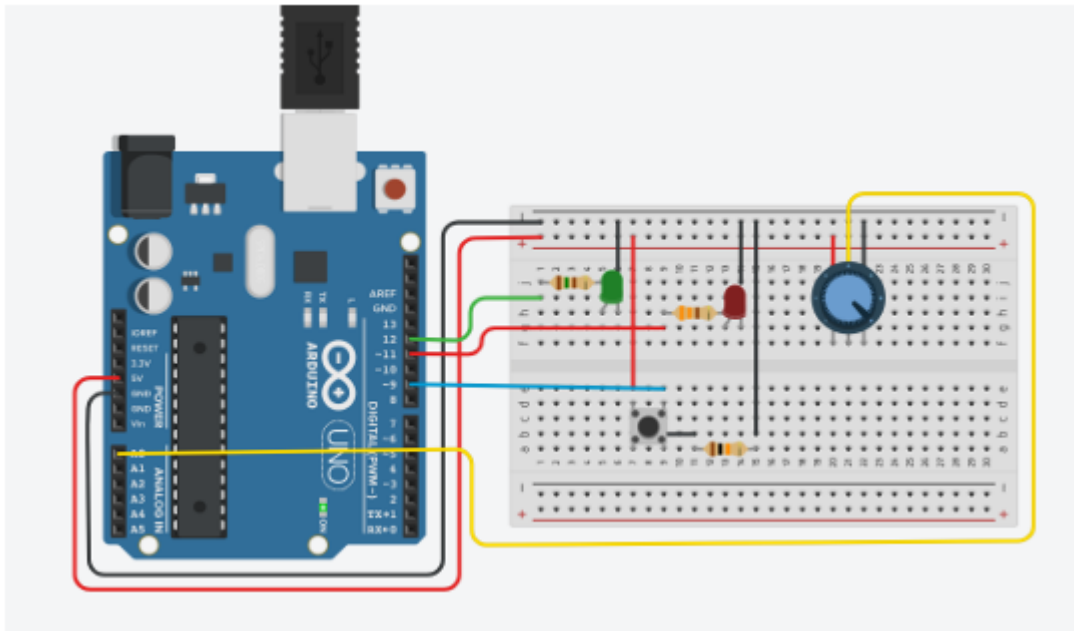
- Leitura do estado do botão com debounce para evitar ruídos.
- Leitura do valor do potenciômetro.
- Controle do LED no pino 11 pelo Potenciômetro via (Point link).

- Controle do LED no pino 12 via botão físico.
- Controle do LED no pino 13 via ScadaBR.
- Atualização dos registradores Modbus para comunicação com o ScadaBR.
- Monitoramento em tempo real via interface gráfica no ScadaBR.

3.5 Circuito eletrônico

O circuito eletrônico foi testado inicialmente via plataforma online TinkerCAD, onde foi possível testar virtualmente todo o sistema antes de montá-lo fisicamente. Na Figura 9 temos o circuito eletrônico detalhado no TinkerCAD.

Figura 9 – Circuito no TinkerCAD.



Fonte: Autores.

3.6 Configuração ScadaBR

Após o Circuito montado e código carregado no Arduíno, iniciou-se a configuração do ScadaBR.

a) Data Sources: Criação de data source usando o protocolo ModBus Serial, onde são criadas e armazenados os data points. A figura 10 mostra o data source nomeado como “Arduíno”, criado dentro da plataforma ScadaBR.

Figura 10 – Criação do data source Arduíno.



Fonte: Autores.

b) Data points: Tags Modbus para leitura e escrita dos registradores do Arduíno, usando os Holding Register, conforme ordem de declaração no código Arduíno. Na figura 11 podem ser analisados todos os data points criados neste projeto.

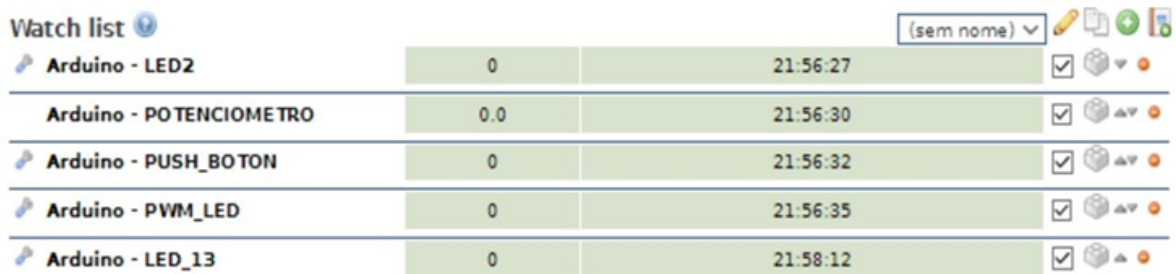
Figura 11 – Data Points criados.

Data points					
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
LED2	Binário		1	Registrador holding	3/0
LED_13	Binário		1	Registrador holding	4/0
POTENCIOMETRO	Numérico		1	Registrador holding	0
PUSH_BOTON	Binário		1	Registrador holding	2/0
PWM_LED	Numérico		1	Registrador holding	1

Fonte: Autores.

c) Watch list: Conferência de conexão, atualização e escrita de dados dentro da watch list conforme cada tags e holding register correspondente a cada sensor e atuador.

Figura 12 – Watch list.



Component	Value	Timestamp	Actions
Arduino - LED2	0	21:56:27	Check, Refresh, Stop
Arduino - POTENCIOMETRO	0.0	21:56:30	Check, Refresh, Stop
Arduino - PUSH_BOTON	0	21:56:32	Check, Refresh, Stop
Arduino - PWM_LED	0	21:56:35	Check, Refresh, Stop
Arduino - LED_13	0	21:58:12	Check, Refresh, Stop

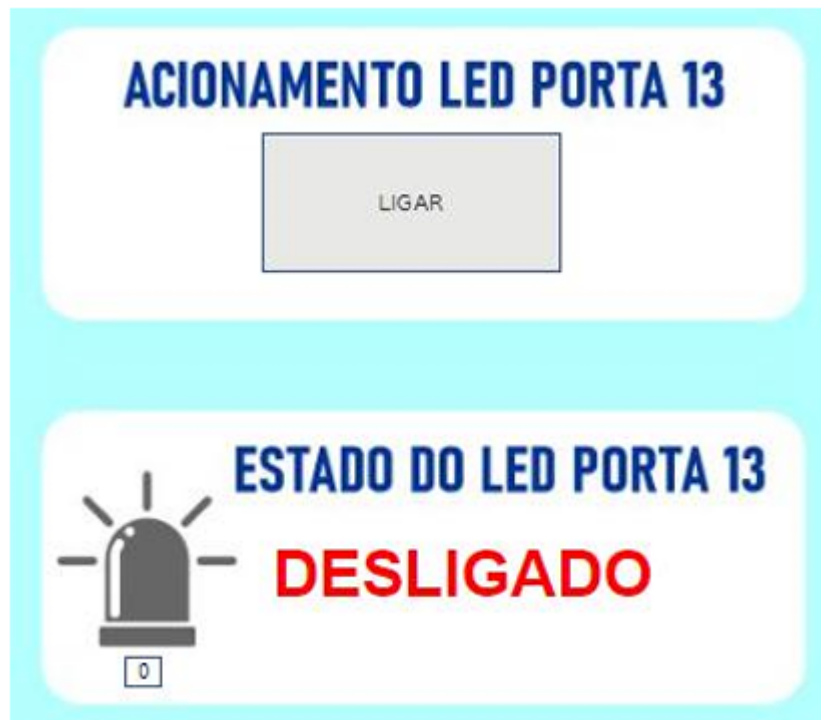
Fonte: Autores.

3.7 Interface gráfica no ScadaBR

A criação da interface gráfica de monitoramento e acionamento do sistema, obteve resultados satisfatórios, todavia não tenha seguido as normas relacionadas, contudo mostrou-se intuitiva e usual.

O botão LIGA/DESLIGA permite o controle do LED do pino 13 diretamente na interface, exibindo o estado “LIGADO” quando o valor é 1 e “DESLIGADO” quando o valor é 0. A figura 13 mostra esta parte da interface criada.

Figura 13 – Botão e monitoramento do LED porta 13.



Fonte: Autores.

O ScadaBR também monitora o valor do potenciômetro (1 a 1023), ajustando proporcionalmente a intensidade luminosa do LED vermelho (pino 11), conforme indicado nas barras de preenchimento. A figura 14 mostra esse monitoramento das barras referente aos dados lidos pelo ScadaBR.

Figura 14 – Monitoramento da leitura do potenciômetro e intensidade do LED.



Fonte: Autores.

O estado do botão push button e do LED verde conectado ao pino 12 é atualizado em tempo real, exibindo “FECHADO” e o valor “1” enquanto o botão está pressionado. O LED alterna entre ligado e desligado a cada acionamento, com retorno visual no supervisor. Essa parte da interface pode ser observada na figura 15.

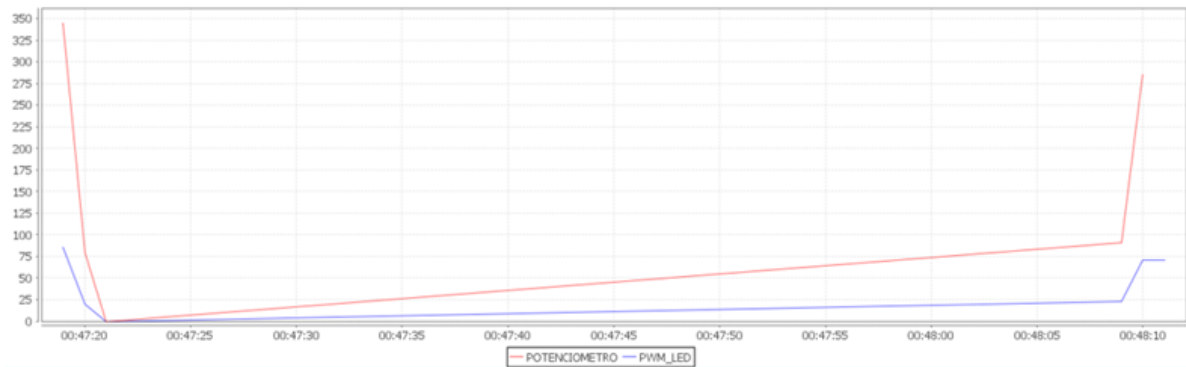
Figura 15 – Monitoramento do push button e LED verde.



Fonte: Autores.

Foi implementado um gráfico comparativo em tempo real entre os valores lidos do potenciômetro e a intensidade PWM do LED vermelho. Essa comparação pode ser vista na figura 16.

Figura 16 – Gráfico comparativo valores LED pwm e Potenciômetro.



Fonte: Autores.

3.8 Testes e validação

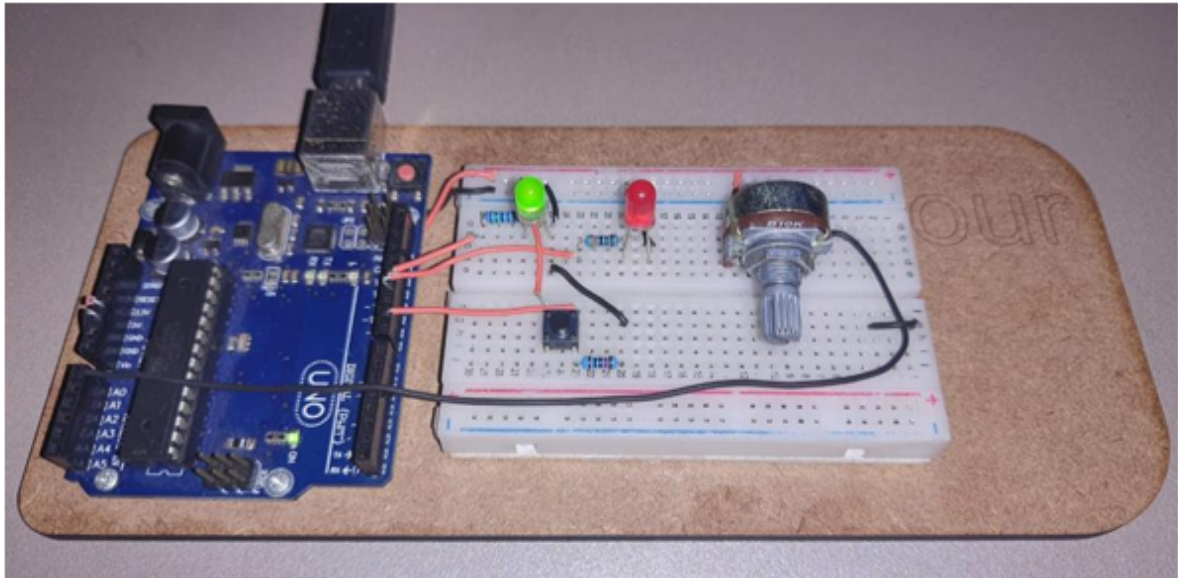
O sistema foi testado em várias etapas:

1. Verificação da comunicação Modbus entre o Arduino e o ScadaBR.
2. Teste do controle do LED no pino 13 via ScadaBR.
3. Teste do botão físico para alternar o estado do LED no pino 12.
4. Validação da exibição do estado do LED no pino 12 na interface do ScadaBR.
5. Leitura e exibição do valor do potenciômetro na interface do ScadaBR.

4 RESULTADOS

O circuito físico foi montado em uma base cortada a laser, para facilitar a organização dos componentes. Todas as funções desenvolvidas foram executadas conforme o esperado. A figura 17 ilustra o circuito montado.

Figura 17 – Circuito Físico montado



Fonte: Autores

4.1 Comunicação Modbus

A comunicação Modbus foi estabelecida com sucesso, permitindo a leitura e escrita dos registradores do Arduino a partir do ScadaBR. Foi constatado estabilidade nos dados transmitidos, sem falhas ou ruídos durante os testes.

4.2 Controle do LED no pino 13

O LED no pino 13 foi controlado com sucesso pela interface do ScadaBR. Ao escrever “1” no registrador correspondente, o LED foi ligado, e ao escrever “0”, o LED foi desligado, confirmando a comunicação entre o ScadaBR e o Arduino.

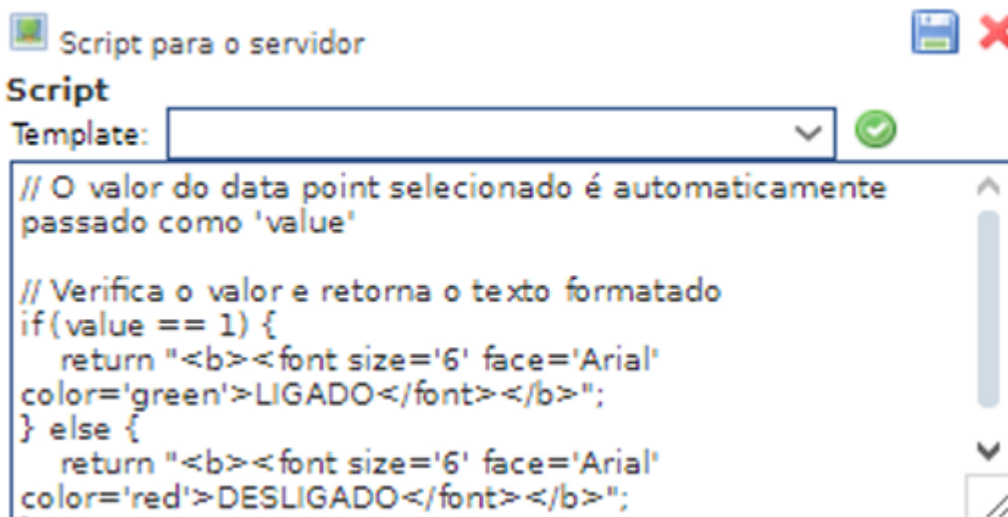
4.3 Controle do LED no pino 12

O botão físico controlou o LED no pino 12, com algoritmo de debounce para prevenir leituras erradas. O estado do LED foi atualizado no ScadaBR em tempo real.

4.4 Exibição do estado dos LED nos pinos 12 e 13 (Arduino)

O estado dos LED no pino 12 foi exibido na interface do ScadaBR como “LIGADO” ou “DESLIGADO”, utilizando um script JavaScript para formatar a exibição, que pode ser conferido na figura 18.

Figura 18 – Código JavaScript para monitoramento.



```
Script para o servidor
Script
Template:
// O valor do data point selecionado é automaticamente
passado como 'value'

// Verifica o valor e retorna o texto formatado
if (value == 1) {
    return "<b><font size='6' face='Arial'
color='green'>LIGADO</font></b>";
} else {
    return "<b><font size='6' face='Arial'
color='red'>DESLIGADO</font></b>";
}
```

Fonte: Autores.

4.5 Leitura do potenciômetro e controle do LED vermelho

A interface do ScadaBR exibiu corretamente o valor lido do potenciômetro. O valor foi ajustado em tempo real, demonstrando as alterações na posição do potenciômetro, e os valores foram vinculados por meio do “Point Link” para regular e controlar a intensidade do LED vermelho conectado ao pino 11.

5 CONCLUSÃO

Ao colocar o sistema em prática, ficou claro que o Arduino e o ScadaBR conseguem se comunicar de forma eficiente usando o protocolo Modbus. Isso demonstra que é possível controlar e acompanhar aparelhos à distância e em tempo real. Juntar esses programas gratuitos foi uma ótima maneira de tornar o aprendizado sobre redes industriais mais fácil e barato. O sistema criado simula como uma fábrica funciona, mostrando a viabilidade de aprender sobre comunicação mestre-escravo, registradores Modbus e como usar a interface SCADA com pouco recurso.

O projeto alcançou todos os objetivos, provando que é possível ensinar redes industriais na prática usando tecnologias com acesso gratuito. Este estudo mostra a importância de aprender fazendo e oferece uma alternativa para instituições de ensino com recursos limitados.

Como perspectivas futuras, sugere-se a expansão do sistema, permitindo o controle de múltiplos dispositivos Modbus e a integração com sensores mais complexos, além da implementação de rotinas de segurança e otimização de comunicação serial.

Em síntese, usar o Arduino e o ScadaBR representa uma solução pedagógica que aproxima teoria com a prática, contribuindo significativamente para o ensino de automação e redes industriais.

REFERÊNCIAS

KAGERMANN, H et al. Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Acatech – National Academy of Science and Engineering, 8 abr. 2013. Disponível em: https://www.acatech.de/wpcontent/uploads/2018/03/Final_report__Industrie_4.0_accessible.pdf . Acesso em: 9 out. 2025.

LUGLI, Alexandre Baratella; SANTOS, Max Mauro Dias. Redes industriais para automação industrial: AS-I, PROFIBUS e PROFINET. 2. ed. São Paulo: Érica, 2019.

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL (SENAI). Redes industriais: atualização tecnológica em mecatrônica. Porto Alegre: SENAI-RS, 2014. 39 p. (Atualização Tecnológica em Mecatrônica).

MOROMISATO, G. D. Y. et al. A utilização de um software livre no ensino de sistemas de controle. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION – ICECE, 2007, São Paulo. Anais... São Paulo: ICECE, 2007. p. 597–601. Disponível em: https://www.researchgate.net/publication/310452323_A_Utilizacao_de_um_Software_Livre_no_Ensino_de_Sistemas_de_Controle. Acesso em: 14 out. 2025.

ARTERO, Óscar Torrente. Arduino: curso práctico de formación. 1. ed. México: Alfaomega Grupo Editor, 2013. 588 p. ISBN 978-607-707-648-3.

NASCIMENTO, João Maria Araújo do; LUCENA, Pedro Berretta de. Protocolo Modbus. Natal: Universidade Federal do Rio Grande do Norte (UFRN), Departamento de Engenharia de Computação e Automação (DCA), 2003. 12 p.

SCADABR. ScadaBR – Software livre de supervisão e controle. Disponível em: <https://scadabr.org>. Acesso em: 14 out. 2025.

CLEVELAND CLINIC. Nervous system. Cleveland, 2023. Disponível em: <https://my.clevelandclinic.org/health/body/21202-nervous-system>. Acesso em: 14 out. 2025.

ACATECH – NATIONAL ACADEMY OF SCIENCE AND ENGINEERING. Recommendations for implementing the strategic initiative INDUSTRIE 4.0: securing the future of German manufacturing industry. Final report of the Industrie 4.0 Working Group. Munich: Acatech, Apr. 2013. Disponível em: <https://en.acatech.de/publication/recommendations-for-implementing-the-strategic-initiativeindustrie-4-0/>. Acesso em: 14 out. 2025.

FREITAS, Carlos Márcio. Protocolo Modbus. Embarcados, 07 abr. 2014. Disponível em: <https://embarcados.com.br/protocolo-modbus>. Acesso em: 07 out. 2025.

MODBUS ORGANIZATION, INC. Modbus Organization. Westford, MA: Modbus Organization, Inc., [s.d.]. Disponível em: <https://www.modbus.org>. Acesso em: 03 out. 2025.

BOYER, Stuart A. SCADA: supervisory control and data acquisition. 3. ed. Research Triangle Park, NC: ISA – The Instrumentation, Systems and Automation Society, 2004. ISBN 1-55617-877-8.

ELIPSE SOFTWARE. Elipse E3– plataforma HMI/SCADA para aplicações avançadas e distribuídas. Disponível em: <https://www.elipse.com.br/produto/elipse-e3>. Acesso em: 13 out. 2025.

SCADABR. Manual ScadaBR 0.7: sistema open-source para supervisão e controle. Florianópolis, out. 2010. Disponível em: <https://www.scadabr.org>. Acesso em: 01 out. 2025.

SILVA, Marlon Ramos; OLIVEIRA, Ângelo Rocha de; CARMO, Marlon José do; ARAÚJO JÚNIOR, Lindolpho Oliveira de. Importância da ferramenta ScadaBR para o ensino em Engenharia. Leopoldina: Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG), 2013.

SILVA, Marlon Ramos et al. Importância da ferramenta ScadaBR para o ensino em Engenharia. Leopoldina: CEFET-MG, 2013.

CARNEIRO, José Airton Borges. Aplicação da metodologia de aprendizagem baseada em projeto utilizando o ScadaBR. 2021. 100 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal do Ceará, Fortaleza, 2021.

ANDREWS, Beth L. Hands-on engineering: grades 4–6. 2nd ed. New York: Routledge, 2021. 203 p. ISBN 978-1-0321-4376-7. DOI

10.4324/9781003235453.

STEELE, Alan L. *Experiential learning in engineering education*. 1. ed. Boca Raton, FL: CRC Press, 2023. 261 p. ISBN 978-0-367-43962-0. DOI 10.1201/9781003007159.

GIBBS, Graham. *Learning by doing: a guide to teaching and learning methods*. Oxford: Oxford Centre for Staff and Learning Development, Oxford Brookes University, 1988. Reimpressão online, 2013. ISBN 978-1-873576-86-1. Disponível em: <https://thoughtsmostlyaboutlearning.wordpress.com/wp-content/uploads/2015/12/learning-bydoing-graham-gibbs.pdf> . Acesso em: 10 out. 2025.

TULLIO, Franciele Braga Machado (org.). *Engenharia na prática: importância teórica e tecnológica*. Ponta Grossa: Atena Editora, 2020. 1. ed. 428 p. ISBN 978-65-5706-308-8. DOI 10.22533/at.ed.088202408.

¹ Discente do Curso Superior de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia Campus Porto Velho Calama. e-mail: alfredo.justiniano@gmail.com

² Discente do Curso Superior de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia Campus Porto Velho Calama. e-mail: thiagopvh@live.com

³ Docente do Curso Superior de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia Campus Porto Velho Calama. Mestre em Sistemas Mecatrônicos (UNB). e-mail: kariston.alves@ifro.edu.br

⁴ Docente do Curso Superior de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia Campus Porto Velho Calama. Mestre em Sistemas Mecatrônicos (UNB). e-mail: rafael.pissinati@ifro.edu.br

[← Post anterior](#)

[Post seguinte →](#)

RevistaFT

A RevistaFT têm 29 anos. É uma **Revista Científica Eletrônica Multidisciplinar Indexada de Alto Impacto e Qualis “B2”**.

Periodicidade mensal e de acesso livre. Leia gratuitamente todos os artigos e publique o seu também [clikando aqui](#),



Contato

Queremos te ouvir.

WhatsApp: (21) 99451-7530

WhatsApp: (21) 99217-2623

e-Mail:
contato@revistaf
t.com.br

ISSN: 1678-0817

CNPJ:
48.728.404/0001-
22

Fator de impacto FI=
5.397 (muito alto)

Turismo
Acadêmico

Agência **ft**

Conselho Editorial

Editores

Fundadores:

Dr. Oston de
Lacerda Mendes.
Dr. João Marcelo
Gigliotti.

Editor

Científico:

Dr. Oston de
Lacerda Mendes

Jornalista

Responsável:
Marcos Antônio
Alves MTB

6036DRT-MG

Orientadoras:

Dra. Hevellyn
Andrade
Monteiro

Dra. Chimene
Kuhn Nobre

Revisores:

Lista atualizada
periodicamente
em

revistaft.com.br/expandente Venha
fazer parte de
nosso time de
revisores
também!

Copyright © Revista ft Ltda. 1996 -
2025

Rua José Linhares, 134 - Leblon | Rio
de Janeiro-RJ | Brasil