

Universidade Federal de Minas Gerais  
Escola de Engenharia  
Curso de Graduação em Engenharia de Controle e Automação

**Planta acadêmica de baixo custo para facilitar prática remota dos laboratório de controle e automação.**

Lucas Neves Lima

Orientador: Prof. Vítor Costa da Silva Campos

Belo Horizonte, Dezembro 2022



## **Monografia**

**Planta acadêmica de baixo custo para facilitar prática remota dos laboratório de controle e automação.**

Monografia submetida à banca examinadora designada pelo Colegiado Didático do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Minas Gerais, como parte dos requisitos para aprovação na disciplina Projeto Final de Curso II.

Belo Horizonte, Dezembro de 2022



# Resumo

Procurando soluções para problemas que se destacaram ainda mais durante o período do ensino remoto, o presente trabalho propõe uma alternativa para os módulos existentes em algumas disciplinas práticas de controle e automação. A solução apresentada é o desenvolvimento de uma planta acadêmica para diminuir o custo de se realizar as atividades que saem da teoria. O módulo proposto simula uma situação encontrada no dia a dia do Engenheiro de Controle e Automação, que é o controle de umidade e temperatura em uma sala, que pode ser visto em museus, salas limpas e cabines de pintura automotiva.

Para cumprir esse objetivo, foi estipulado um valor máximo, sendo usados hardwares e softwares de baixo ou nenhum custo, por exemplo, uso do Codesys e Arduino, usado como “interface” de I/O de um SoftPLC por meio da comunicação MODBUS. O resultado é uma alternativa relativamente portátil e com um custo de desenvolvimento realizável em escala.



# Agradecimentos

Gostaria de agradecer ao apoio de todos que me influenciaram na escolha da carreira de engenheiro e aos amigos e professores que tive durante toda a etapa de formação até a graduação.





# Sumário

<b>Resumo</b>	<b>i</b>
<b>Agradecimentos</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>viii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Justificativa . . . . .	1
1.2 Ensino remoto e o aprendizado de Engenharia . . . . .	2
1.3 Os problema do ensino remoto nas disciplinas práticas . . . . .	2
1.4 Objetivos do Projeto . . . . .	3
1.5 Estrutura da Monografia . . . . .	3
<b>2 Apresentação da proposta e impacto da prática no aprendizado</b>	<b>5</b>
2.1 Apresentando o protótipo da planta acadêmica . . . . .	5
2.2 O impacto das práticas no aprendizado. . . . .	6
2.3 Propondo melhorias para o ensino . . . . .	6
<b>3 Especificações</b>	<b>7</b>
3.1 O uso plataforma Arduino . . . . .	7
3.2 SoftPLCs . . . . .	9
3.3 Codesys como ferramenta de programação de PLCs . . . . .	11
3.4 Protocolo Modbus no uso do Arduino como interface de I/O . . . . .	11
3.4.1 O protocolo Modbus . . . . .	12
3.5 Salas com controle de umidade e temperatura . . . . .	12
3.6 Sensor de temperatura e umidade . . . . .	14
3.7 Atuadores no Controle de umidade e temperatura . . . . .	15
3.7.1 Placas de Peltier . . . . .	15
3.8 O projeto da planta acadêmica . . . . .	15
<b>4 Desenvolvimento</b>	<b>17</b>
4.1 Etapas do Desenvolvimento . . . . .	17
4.2 Desenvolvimento da Estrutura da Estufa . . . . .	18
4.3 Etapa de configuração de software . . . . .	18
4.3.1 Arduino . . . . .	18

4.3.2	Codesys . . . . .	23
4.4	Montagem da planta . . . . .	25
<b>5</b>	<b>Resultados</b>	<b>27</b>
5.1	A Estufa . . . . .	27
5.1.1	Preço . . . . .	29
5.2	Proposta módulos de ensino . . . . .	29
5.2.1	Prática de laboratório informática industrial . . . . .	29
5.2.2	Prática de laboratório Controle . . . . .	31
<b>6</b>	<b>Conclusões</b>	<b>37</b>
6.1	Propostas de Continuidade . . . . .	37
	<b>Referências Bibliográficas</b>	<b>39</b>

# Lista de Figuras

2.1	Modelo final da planta acadêmica . . . . .	5
3.1	Placa Arduino Uno R3 Usada no desenvolvimento do projeto. . . . .	7
3.2	Imagem da IDE para programação do Arduino . . . . .	9
3.3	Dilema da inovação no contexto dos SoftPLCs Fonte: Soft PLCs: The industrial innovator's dilemma, Sharmila Annaswamy 11	
3.4	Frame do protocolo Modbus RTU Fonte: Modicon Modbus Protocol Reference Guide, 1996 . . . . .	13
3.5	Cabine de Pintura Automotiva . . . . .	13
3.6	Vitrine com ambiente controlado em museus. . . . .	14
3.7	Sensor DHT11 . . . . .	14
3.8	Condensador feito para retirada de umidade do ambiente. . . . .	16
4.1	Vista em corte da estufa . . . . .	19
4.2	Vista lateral 1 da estufa. . . . .	20
4.3	Vista lateral 2 da estufa em corte . . . . .	21
4.4	Código que roda no arduino com Métodos de Leitura e Escrita dos dados no Modbus. . . . .	22
4.5	Função de leitura das entradas. . . . .	24
4.6	Função de escrita das saídas. . . . .	24
4.7	Esquemático de ligações no Arduino. . . . .	25
5.1	Planta acadêmica - Estufa com controle de Temperatura e Umidade. . . . .	28
5.2	Vista Interna - Planta acadêmica - Estufa com controle de Temperatura e Umidade . . . . .	28
5.3	Configuração das TAGs como variáveis globais. . . . .	30
5.4	IHM para controlar a partir da interface a planta acadêmica. . . . .	30
5.5	Mapeando as entradas e saídas configuradas via MODBUS no Arduino para as variáveis globais criadas. . . . .	31
5.6	Servidor OPC para manipular as variáveis manipuladas e monitor as variáveis de processo. . . . .	31
5.7	Dados da reposta a um pulso do sistema de aumento da temperatura - Experimento 1 . . . . .	32
5.8	Comparação entre modelo e dados coletados para o sistema de aumento de temperatura. . . . .	33
5.9	Dados da reposta a um pulso do sistema de redução de umidade . . . . .	34

5.10	Comparação entre modelo e dados coletados para o sistema de redução de umidade. . . . .	34
5.11	Tela do PID Tunner com o resultado do controlador projetado para a malha de aquecimento. . . . .	35
5.12	Interface interativa para manipular valores das constantes do PID. . . . .	35

# Lista de Tabelas

3.1	Especificações Arduino Uno R3. . . . .	8
3.2	Tabela Funções protocolo Modbus. . . . .	12
5.1	Tabela de custo planta acadêmica. . . . .	29



# Capítulo 1

## Introdução

### 1.1 Motivação e Justificativa

No ensino de Engenharia de Controle e Automação, a dificuldade de assimilação das disciplinas teóricas é conhecida, dificuldade esta que foi agravada devido a uma série de mudanças e consequências geradas pelo ensino remoto emergencial durante a pandemia de COVID-19. Dentre os problemas relacionados ao ensino, agravados pelo afastamento forçado das atividades presenciais, destaca-se a viabilidade e acessibilidade a práticas laboratoriais de engenharia. Disciplinas que contam com uma infraestrutura *in loco*, as plantas e módulos didáticos foram prejudicadas tendo sua oferta cancelada ou foram remodeladas usando simulações virtuais, impossibilitando que os futuros engenheiros aplicassem seu conhecimento teórico em cenários realistas com consequências sensoriais mais claras. Existe uma diferença no processo de construção do conhecimento entre a abordagem da práxis em detrimento de simulações e conhecimentos teóricos, como pode ser visto em Amorim e Aegeli (1996), ficando claro, assim, o prejuízo acarretado aos alunos que perdem essa experiência.

Pensando nessa problemática, esse trabalho tenta trazer uma alternativa que viabilize o processo de aprendizado completo do futuro engenheiro de controle e automação, mesmo em situações adversas. Para construir a solução desse trabalho, é preciso trazer a tona os pontos que podem ser melhorados em relação aos projetos didáticos já existentes até o presente momento. Por meio da experiência cursando disciplinas no modelo tradicional e algumas no modelo remoto e conversando com professores, as seguintes características das práticas impeditivas e ou indesejadas foram listadas: alto custo, baixa portabilidade, complexidade de montagem, irreprodutibilidade de casos reais e obsolescência.

Atualmente, as plantas acadêmicas do curso de controle e automação da Universidade Federal de Minas Gerais (UFMG) fazem uso de CLPs, sensores e atuadores de alto custo, que se justificam por serem os mesmos equipamentos com que o engenheiro formado vai se deparar na indústria, mas têm um alto custo, como os CLPs CompactLogix 1769-L32ER contidos no MICA módulo de aprendizado (PASSOS e BRAGA, 2008) usados na disciplina de informática industrial R\$10,851.77 e instrumentação. Limitações de software também existem, como no caso do uso de ferramentas que não são gratuitas ligadas a equipamentos que só funcionam com elas, deixando, muitas vezes, o aluno em contato com um software muito antigo. Outra característica é o uso do Matlab em várias práticas das disciplinas de controle, mesmo sendo uma ferramenta poderosa e importante para aprendizado, grande parte das práticas nela são simuladas ou com uma interface difícil com os periféricos da

planta. A solução proposta neste trabalho busca fornecer um ambiente mais próximo ao encontrado no mercado de trabalho, onde o engenheiro tem que resolver os problemas usando as linguagens de programação definidos pela norma IEC 61131-3, como: Function Block Diagram(FBD), Sequential Function Chart(SFC), Structured Text (ST) e até mesmo lógica Ladder.

Em contrapartida ao cenário apresentado anteriormente, temos um número cada vez maior de computadores e microcontroladores de baixo custo que estão disponíveis no mercado. Podemos destacar, além do preço, a comunidade dos microcontroladores de baixo custo, que é cada vez maior em torno dessas tecnologias, alguns exemplos são: Arduino, NodeMCU e Raspberry Pi. Já no software para programação de CLPs podemos destacar o Codesys que é gratuito, tem amplo suporte a diversas plataformas e que os alunos já tem contato em algumas disciplinas do curso.

Neste contexto, foi observada uma excelente oportunidade para propor e montar uma planta acadêmica de baixo custo, onde os alunos poderão ser avaliados e trabalhar em cima dos conceitos abordados nas disciplinas teóricas. Ao final do trabalho, é desejado chegar em uma solução que tenha um custo baixo, seja portátil, use softwares gratuitos e incorpore um processo próximo ao que pode ser encontrado no dia a dia do engenheiro de Controle e Automação. Existe a intenção que, futuramente, disciplinas como informática industrial, e laboratório de controle possam servir de validação para o protótipo construído.

## **1.2 Ensino remoto e o aprendizado de Engenharia**

O ensino remoto trouxe a tona para o aprendizado de engenharia uma série de novas dificuldades, bem como evidenciou problemas já conhecidos. Pensando nessas questões, o seguinte trabalho apresentará uma solução que buscou atender alguns pontos de principal impacto. Foram apresentadas referências que demonstram: a dificuldade gerada pelo afastamento dos alunos, a importância da práxis no contexto de ensino e consolidação do conhecimento, a diferença entre abordagem prática e simulada no andamento da aprendizagem e também a influência do uso de um processo que se aproxima do contexto real encontrado pelo profissional no dia a dia de trabalho durante as atividade de ensino. Ao final do capítulo o leitor encontrará algumas propostas de solução e a discussão sobre a viabilidade das mesmas.

## **1.3 Os problema do ensino remoto nas disciplinas práticas**

Durante o período de Ensino Remoto Emergencial (ERE), o ensino foi prejudicado, como aponta o relatório da Unesco (MARINONI; VAN'T LAND; JENSEN, 2020) sobre os impactos no ensino superior causados pela pandemia de COVID-19, em que uma das causas foi devido à característica dessa modalidade que a difere do ensino a distância concebido desde o início com o distanciamento dos alunos da sala de aula. Levando em conta o aspecto intrínseco do ERE, que é o curto tempo para planejamento, e mesmo com o relevante esforço do corpo docente, foi possível elencar problemas casuais e também antigos percebidos no contexto de um aluno de Engenharia de Controle e Automação da UFMG, são eles:

- A condição temporária de acesso restrito aos laboratórios devido a medidas sanitárias;
- O limite de recursos para disponibilização das práticas, módulos de ensino;



- A falta de manuseabilidade e portabilidade dos módulos de ensino existentes;
- O número limitado de módulos de ensino;
- Simulações com exigência de hardwares computacional alto para serem executadas pelos alunos;
- O custo elevados das plantas acadêmicas existentes;
- A dificuldade de acesso no mercado aos equipamentos usados nos módulos de ensino, que impede a reprodução por parte do aluno.

## 1.4 Objetivos do Projeto

Como já foi citado anteriormente, o principal objetivo é construir uma plataforma de ensino que seja acessível e contemple as áreas de práticas dentro do ensino de Engenharia de Controle e Automação. A planta acadêmica deve seguir algumas especificações, como ter um custo total inferior a R\$ 400,00, usar software gratuito para programação de CLPs que segue a norma IEC 61131-3 e fazer uso de hardware acessível, outros objetivos desse PFC são:

- Pesquisa e Teste de alternativas de baixo custo às tecnologias de hardware e software usadas atualmente nos laboratórios;
- Projeto de uma planta acadêmica com processo similar ao encontrado no dia a dia do engenheiro de controle e automação;
- Implementação e montagem do módulo de estudo projetado.
- Implementação de prática de controle e informática industrial que podem ser feitas em cima do módulo.

## 1.5 Estrutura da Monografia

O trabalho está dividido em seis capítulos. Este apresenta uma introdução ao projeto, os motivadores com referências para justificar a realização do trabalho e o segundo fala sobre a proposta para melhoria do ensino remoto e os impactos desejados com a implementação da planta. O capítulo 3 mostra os requisitos e especificações do projeto a ser desenvolvido. O capítulo 4 apresenta como foi o desenvolvimento, a montagem da planta acadêmica e desenvolvimento dos módulos necessários e o quinto apresenta os resultados, como custos, objetivos atingidos e dificuldades encontradas. Finalizando, o capítulo 6 compila a conclusão do trabalho, dando, ainda, sugestões para continuações ao trabalho.



## Capítulo 2

# Apresentação da proposta e impacto da prática no aprendizado

Com os problemas e motivadores apresentados, o objetivo desse capítulo é apresentar em termos gerais o projeto que será desenvolvido e discutir os impactos que o mesmo pode gerar no aprendizado dos alunos.

### 2.1 Apresentando o protótipo da planta acadêmica

A solução pensada para resolver as questões trazidas pelo capítulo 1 consiste em um protótipo de uma planta acadêmica, mesmo que o uso da palavra planta não se encaixe dentro do que o projeto é, pois o mesmo, diferente das plantas industriais, não tem objetivo de produzir um produto por fim.

A planta é um caixa feita com materiais de baixo custo e usando sensores e atuadores de fácil acesso, de forma com que seja rápido e barata a sua construção. Parte da solução também é a escolha dos softwares sem custo e que seguem os padrões adotados na indústria para programação de CLPs. Com esse conjunto tenta-se se aproximar de um ambiente que simula situações reais de sala controlada como estufas de pintura.

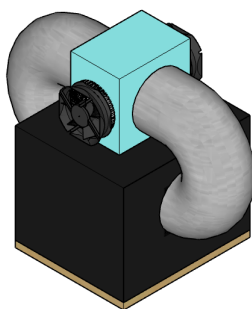


Figura 2.1: Modelo final da planta acadêmica

## **2.2 O impacto das práticas no aprendizado.**

O ensino de engenharia vem ganhando cada vez mais novas ferramentas com a evolução tecnológica e também das metodologias de ensino. Uma característica que pode ser lembrada, pois sempre se destacou no campo de conhecimento da engenharia devido à natureza inovadora e exigência do sistema produtivo, é aplicação de conceitos construídos durante o ciclo básico de formação na solução de problemas multidisciplinares.

## **2.3 Propondo melhorias para o ensino**

Enfim, para finalizar o esclarecimento da justificativa do respectivo trabalho, vale citar os problemas de ensino na Engenharia sempre foram pauta e estão cada vez mais sendo discutidos em congressos específicos para o tema, como, por exemplo, o Congresso Brasileiro de Educação em Engenharia (COBENGE). A proposta deste trabalho é consoante com outros que apresentam também módulos de aprendizado e busca ser uma opção usando ferramentas populares e de baixo custo para resolver os problemas elencados.

# Capítulo 3

## Especificações

O presente capítulo apresenta detalhes da solução proposta para o desenvolvimento da planta didática. Será exposto nessa sessão um contexto geral, especificação e justificativa para escolha de cada um dos componentes presentes no módulo, passando pelas bibliotecas de linguagens de programação, protocolos de comunicação, características de hardware e do processo utilizado. Ao final, foi detalhada a maneira como todas as peças descritas se encaixam, especificando, assim, o protótipo da planta didática.

### 3.1 O uso plataforma Arduino

O Arduino é uma plataforma composta por um framework e um conjunto de hardwares e foi concebida com o intuito de facilitar a programação de microcontroladores para um processo de prototipagem rápida. O projeto nasceu herdando bastante do projeto WIRING, usando o microcontrolador ATmega8. As primeiras versões de hardware do Arduino tinham porta serial e só depois evoluíram para portas USB com conversor serial, a partir do Arduino Uno, lançado em 2010 o conversor USB-Serial foi trocado para um microcontrolador ATmega16U2, na versão de revisão 3, o microcontrolador usado é o ATMEGA328P. Uma imagem do modelo usado e a tabela de especificações do componente podem ser encontradas respectivamente na figura 3.1 e tabela 3.1.

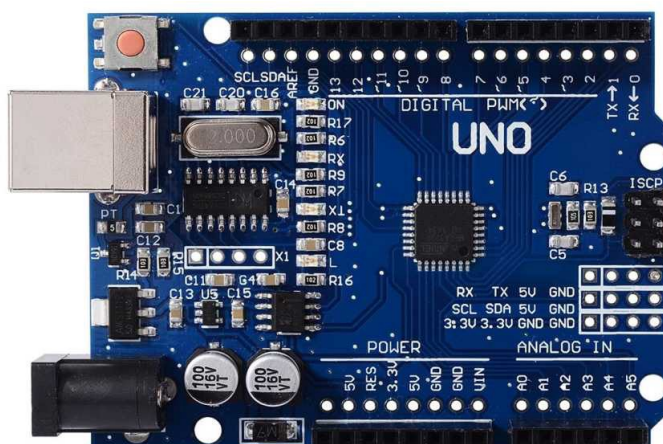


Figura 3.1: Placa Arduino Unu R3 Usada no desenvolvimento do projeto.

<b>Board</b>	<b>Name</b>	<b>Arduino UNO R3</b>
	<b>SKU</b>	A000066
<b>Microcontroller</b>	ATmega328P	
<b>USB connector</b>	USB-B	
<b>Pins</b>	<b>Built-in LED Pin</b>	13
	<b>Digital I/O Pins</b>	14
	<b>Analog input pins</b>	6
	<b>PWM pins</b>	6
<b>Communication</b>	<b>UART</b>	Yes
	<b>I2C</b>	Yes
	<b>SPI</b>	Yes
<b>Power</b>	<b>I/O Voltage</b>	5V
	<b>Input voltage (nominal)</b>	7-12V
	<b>DC Current per I/O Pin</b>	20 mA
	<b>Power Supply Connector</b>	Barrel Plug
<b>Clock speed</b>	<b>Main Processor</b>	ATmega328P 16 MHz
	<b>USB-Serial Processor</b>	ATmega16U2 16 MHz
<b>Memory</b>	<b>ATmega328P</b>	2KB SRAM, 32KB FLASH, 1KB EEPROM
<b>Dimensions</b>	<b>Weight</b>	25 g
	<b>Width</b>	53.4 mm
	<b>Length</b>	68.6 mm

Tabela 3.1: Especificações Arduino Uno R3.

O software para programação do Arduino está sob a licença General Public License (GPL), que garante ao usuário da plataforma permissão para usar, compartilhar e modificar o software, já o hardware está coberto pela Creative Commons (CC) license, uma licença de distribuição gratuita que habilita que outras pessoas desenvolvam seus projetos nas placas e tenham o direito de distribuí-los. Dentre os diversos modelos o escolhido foi um dos mais populares, o Arduino Uno R3. É possível ver a interface da IDE de programação do Arduino na imagem 3.2, nessa imagem temos o código desenvolvido pronto para ser carregado no dispositivo.

Os principais pontos para a escolha desse dispositivo para ser usado no projeto foi devido a sua grande popularidade como ferramenta de prototipagem, baixo custo, valores variando de R\$40,00 a R\$120,00, e destaque no meio open-source como uma das comunidades mais ativas em fóruns e sites. Esse fato contribuiu para este trabalho no momento de encontrar bibliotecas de comunicação Modbus e para leitura de dados de sensores, que foram de grande valor no desenvolvimento desse trabalho, a disponibilização de uma Interface de Desenvolvimento Integrado (IDE) bem amigável também foi um ponto forte no uso da plataforma.

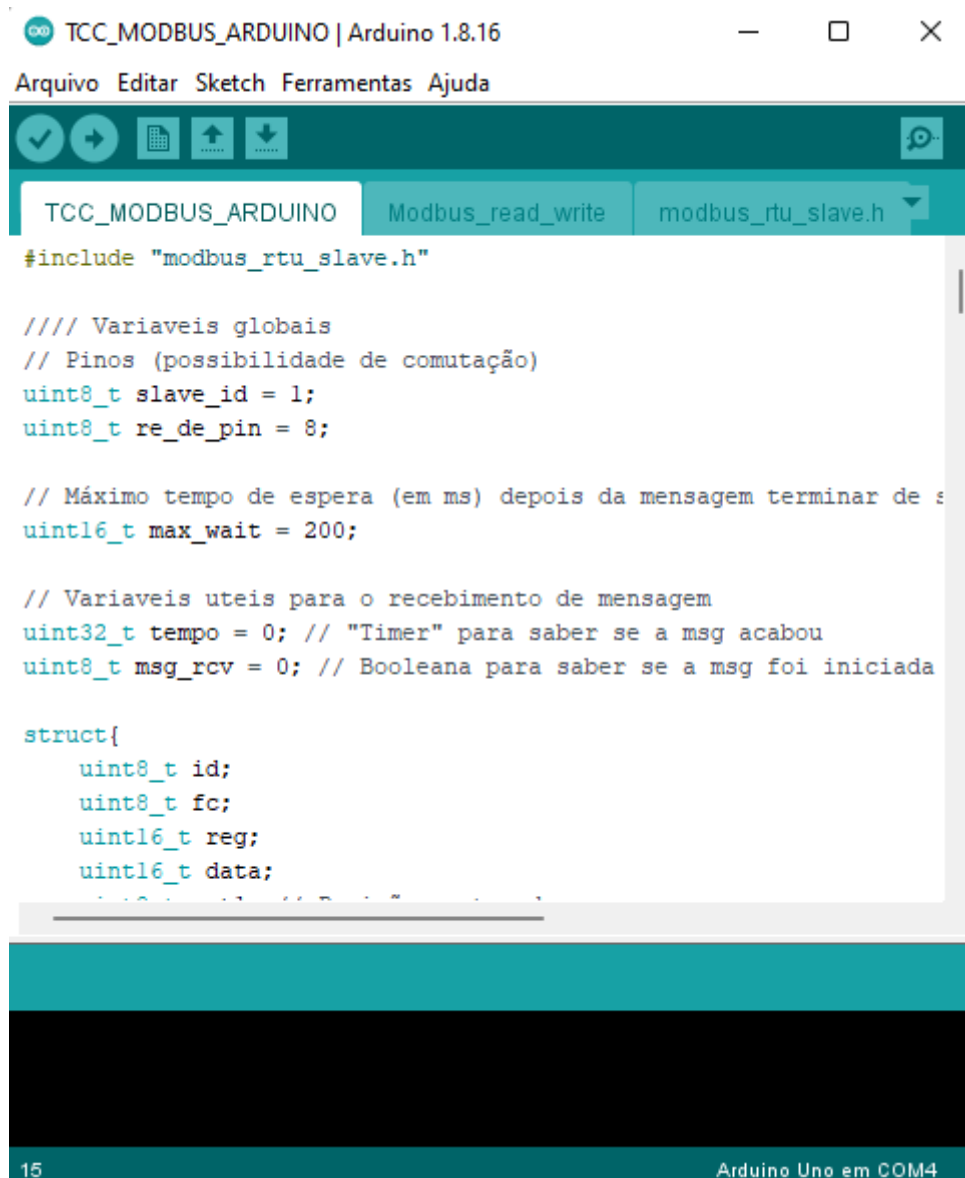


Figura 3.2: Imagem da IDE para programação do Arduino

## 3.2 SoftPLCs

Programmable Logic Controller(PLC) ou Controladores Lógicos Programáveis(CLP), surgiram na indústria automotiva como um alternativa à lógica de relés e são computadores com propósito de operar programas que controlam os diversos processos industriais existentes. Os PLCs se diferenciam de computadores tradicionais por características como a robustez de construção necessária por se tratar de um equipamento responsável por operações críticas e geralmente alocado em ambiente com ruídos de diversos tipos. Os sistemas que esses hardwares rodam também são outra características que distinguem os CLPs dos computadores do dia a dia que rodam Sistemas Operacionais(SOs) de propósito geral, já os PLCs rodam SOs de tempo real.

Com a evolução tecnológica, surgiram os Software Based Programmable Logic Controllers (SoftPLCs), que consistem na configuração de um sistema de controle, em que a lógica de controle é parcial ou complementemente desacoplada do hardware. A característica de desacoplamento permite a virtualização de um software executar as lógicas de controle em diferentes tipos de hardware. Os primeiros SoftPLCs surgiram junto com o Windows em 1985, porém eram muito problemáticos devido à instabilidade do sistema em que eles rodavam. Atualmente, com a evolução dos sistemas operacionais e hardwares os SoftPLCs são muito mais confiáveis e resilientes do que antigamente, chegando a ser alternativa para implementação de controle em processos não críticos.

No presente trabalho, os programas contendo a lógica de controle foram executados usando um SoftPLC. A escolha da solução virtualizada apresentada no parágrafo anterior foi uma alternativa ao HardPLC, com os três principais fatores de escolha sendo o custo, a característica do cunho didático e do processo controlado ser não crítico. O software usado foi o Codesys Control Win V3, um runtime system (aplicação que permite rodar um determinado código compilado para esse runtime) que transforma um computador de propósito geral em um CLP padrão IEC 61131-3, rodando no Windows com propriedades soft real-time. O Codesys Control Win V3 possui as seguintes propriedades:

- Licença gratuita de uso;
- Comunicação com Codesys Development System;
- Suporte à sistema de entrada e saída e fieldbus I/O;
- Retenção dados em disco, memória flash e SD Card;
- Interface com OPC UA server;
- Suporte a field BUS:
  - Codesys CANopen Manager / Device;
  - Codesys EtherCAT Master;
  - Codesys EtherNet/IP Scanner / Adapter;
  - Codesys J1939;
  - Codesys Modbus TCP Master / Slave;
  - Codesys Modbus Serial Master / Slave;
  - Codesys PROFIBUS Master;
  - Codesys PROFINET Controller / Device.

No gráfico mostrado na figura 3.3 é possível ver a popularização do uso de SoftPLCs pela indústria.



### Innovator's dilemma facing industrial automation

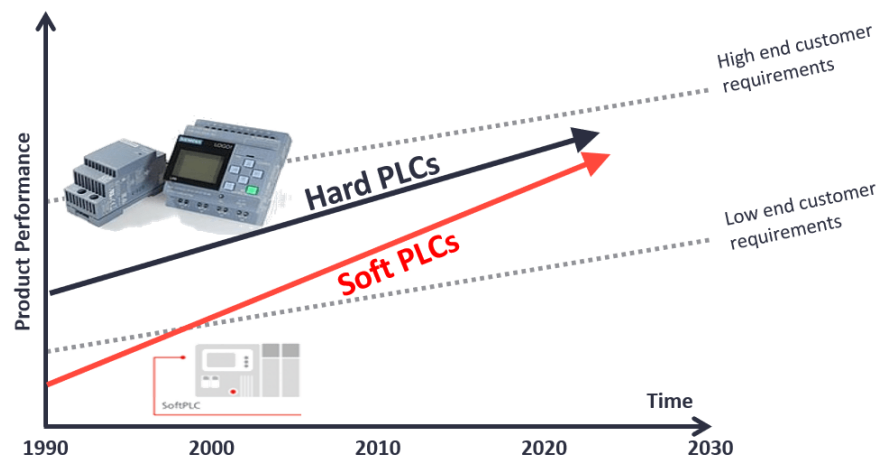


Figura 3.3: Dilema da inovação no contexto dos SoftPLCs

Fonte: Soft PLCs: The industrial innovator's dilemma, Sharmila Annaswamy

## 3.3 Codesys como ferramenta de programação de PLCs

Codesys é uma ferramenta para programação de aplicações de controle que segue o padrão internacional IEC 61131-3. Além de fornecer o ambiente para programação, a ferramenta também oferece uma série de outras funcionalidades como a característica multi dispositivo, por funcionar com CLPs que trabalham no padrão IEC 61131-3. O que é uma grande vantagem em antítese às várias ferramentas proprietárias que apenas funcionam para gerar programas para hardwares do mesmo fabricante, além disso possui, também, ferramentas para construção de Interfaces Homem Máquina(IHM). Uma das principais vantagens do Codesys é sua licença gratuita, o que vai de encontro com o objetivo de baixo custo do trabalho.

## 3.4 Protocolo Modbus no uso do Arduino como interface de I/O

Ao rodar a lógica de controle virtualizada em um SoftPLC, foi preciso uma interface de comunicação com os dispositivos de entrada e saída para interagir com o mundo real. Analisando trabalhos que propõem módulos de aprendizado, foi possível observar como o Arduino sempre é uma ferramenta utilizada, porém não foi encontrado nenhum caso em que foi utilizada linguagem de programação para CLPs interagindo com as entradas e saídas do Arduino. Buscando unir a vantagem de uma ferramenta como o Codesys e do Arduino, foi elaborado uma sistema para o uso do Arduino como se fosse um cartão de I/O do PLC. A fim de realizar a integração citada anteriormente, chegou-se na ideia de configurar o Arduino como um dispositivo Modbus e configurar o mesmo como um dispositivo usado na aplicação construída. Usando um dispositivo Modbus, foi possível fazer o mapeamento das entradas e saídas do Arduino para variáveis do programa. Na tabela 3.2 são apresentadas as diversas funções existentes no protocolo Modbus, onde as funções 2 e 6 foram as principais usadas no trabalho.

Código da função	Descrição
1	Leitura de bloco de bits do tipo coil(saída discreta).
2	Leitura de bloco de bits do tipo entradas discretas.
3	Leitura de bloco de registradores do tipo holding.
4	Leitura de bloco de registradores do tipo input.
5	Escrita em um único bit do tipo coil(saída discreta).
6	Escrita em um único registrador do tipo holding.
7	Ler o conteúdo de 8 estados de exceção.
8	Prover uma série de testes para verificação da comunicação e erro internos.
11	Modbus: Obter o contador de eventos.
12	Modbus: Obter um relatório de eventos.
15	Escrita em bloco de bits do tipo coil(saída discreta).
16	Escrita em bloco de registradores do tipo holding.
17	Ler algumas informações do dispositivo.
20	Ler informações de um arquivo.
21	Escrever informações em um arquivo.
22	Modificar o conteúdo de registradores de espera através de operações lógicas.
23	Combina ler e escrever em registradores numa única transação.
24	Ler o conteúdo da fila FIFO de registradores.
43	Identificação do modelo do dispositivo.

Tabela 3.2: Tabela Funções protocolo Modbus.

### 3.4.1 O protocolo Modbus

O protocolo Modbus é um protocolo de comunicação serial desenvolvido em 1979 que trabalha na camada 7 do modelo OSI, a camada de aplicação. Funciona no modelo Mestre-Escravo (atualizado recentemente para Servidor-Cliente), em que o Mestre sempre inicia a comunicação mandando as requisições de dados para o Escravo, i.e quando é preciso ler um valor, o nó mestre manda um quadro para o nó escravo pedindo a informação. O frame do protocolo é apresentado na figura 3.4 tem os seguinte campos:

- Adress: Endereço do registrador em que se deseja ler ou escrever;
- Function: O código da função que será utilizada, as mais comuns são as da tabela 3.2;
- Data: O dado que está sendo enviado no pacote, podendo ser um dado de escrita, leitura ou informação sobre a mensagem trocada;
- CRC: É um código de detecção de erros no quadro, para garantir que a mensagem não teve alteração durante o envio e continua íntegra.

## 3.5 Salas com controle de umidade e temperatura

O processo real escolhido para inspirar a planta acadêmica foi o de salas controladas, que exigem o controle de várias características do microclima existente. Alguns exemplos de

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4

Figura 3.4: Frame do protocolo Modbus RTU

Fonte: Modicon Modbus Protocol Reference Guide, 1996

salas controladas são: as salas limpas de indústrias de microchips, salas hospitalares, cabines de pintura e armazém de museus. Como visto em Pavlogeorgatos (2003), a umidade é o principal fator de deterioração e a temperatura o terceiro fator de deterioração das peças dentro dos museus. Já no caso da pintura automotiva, cada fabricante tem uma indicação da umidade relativa e temperatura ideal para aplicação do produto, isso ocorre devido às características químicas de cada tipo de tinta.



Figura 3.5: Cabine de Pintura Automotiva

Uma característica comum de todos esses ambientes citados é o controle de umidade e temperatura, essas duas variáveis foram as escolhidas para o módulo de ensino, pois têm característica de variação lenta, o que permite uma taxa de amostragem menor, compatível com as restrições do projeto. As grandezas que foram escolhidas são definidas da seguinte forma:

“A temperatura é uma grandeza física usada para indicar se um corpo está mais ‘quente’ ou mais ‘frio’ do que outros tomados como referência.” - Máximo e Alvarenga Alvarenga (1997)

"A umidade relativa do ar é a relação entre a quantidade de água existente no ar (umidade absoluta) e a quantidade máxima que poderia haver na mesma temperatura (ponto de saturação)."

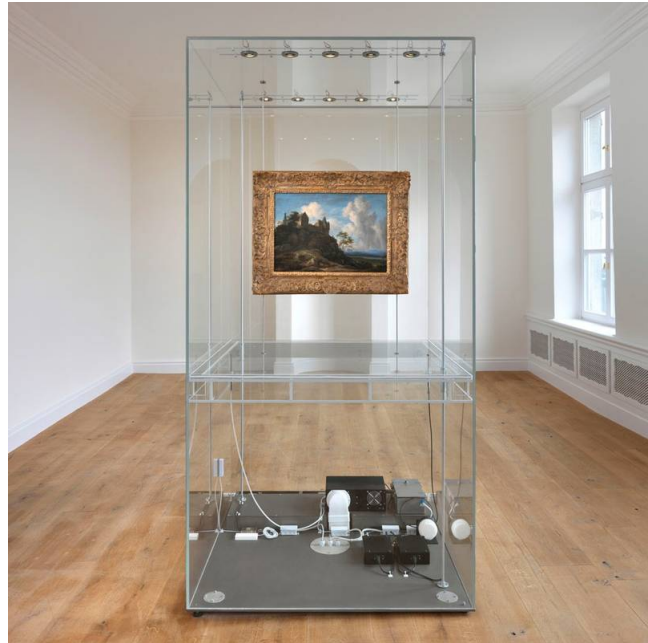


Figura 3.6: Vitrine com ambiente controlado em museus.

### 3.6 Sensor de temperatura e umidade

Para realizar a leitura das variáveis envolvidas no processo, foi escolhido o sensor DHT11, que é capaz de medir umidade relativa e temperatura. O sensor de umidade encontrado dentro do DHT11 capacitivo enquanto o sensor de temperatura é um termistor do tipo NTC (Negative Temperature Coefficient) de temperatura. Os valores dos sensores são lidos e por um microcontrolador dentro do DHT11 e transmitidos de forma digital.

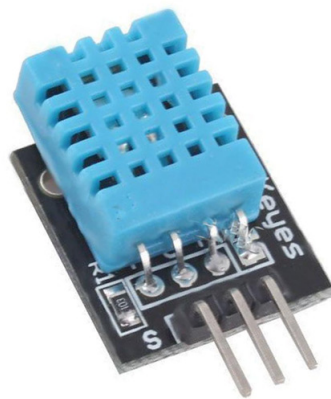


Figura 3.7: Sensor DHT11

Características do DHT11:

- Faixa de medição de umidade: 20 a 90% da Umidade Relativa;
- Faixa de medição de temperatura: 0° a 50°C;
- Alimentação: 3-5VDC (5,5VDC máximo);

- Corrente: 200 $\mu$ A a 500mA, em stand by de 100 $\mu$ A a 150 $\mu$ A;
- Precisão de umidade de medição:  $\pm 5,0\%$  da Umidade Relativa;
- Precisão de medição de temperatura:  $\pm 2.0$  °C;
- Tempo de resposta: 2s;
- Dimensões: 23 x 12 x 5mm (incluindo terminais);

A escolha desse sensor foi baseada no baixo custo, função de leitura das duas grandezas em um único componente e capacidade de transmitir, de maneira digital, os dados para o Arduino.

## 3.7 Atuadores no Controle de umidade e temperatura

Dois tipos de atuadores foram escolhidos para o projeto, um para cada grandeza controlada. Ambos fazem conversão da energia elétrica para térmica, em que a variação da tensão entregue para os mesmos varia a temperatura de ambos.

O dispositivo que vai atuar na temperatura é uma PCB Heatbed, que consiste em uma placa de circuito impresso com trilhas de condutores por onde a corrente elétrica passa gerando calor. Essa mesa de aquecimento resistiva foi desenvolvida para o uso em aplicações de impressora 3D e sua tensão de alimentação é de 12V, resistência entre 1.0 e 1.2 ohms, dimensão de 214 mm x 214 mm e corrente recomendada da fonte para alimentar 14 Ampères. Para atuar na umidade relativa, foi montado um elemento condensador de água a partir do resfriamentos de aletas de cobre. Ao resfriar as aletas em contato com o ar, o vapor d'água encontrado no ambiente passa para o estado líquido, reduzindo, assim, a umidade relativa do ambiente. Para isso, foi usada como fonte fria placas de Peltier e como aletas dissipadores de calor de microcomputadores. O projeto de desumidificador foi inspirado no de Amir Hossein Shourideh (2018), que consiste em condensador de água que tinha como objetivo gerar água do efeito Peltier.

### 3.7.1 Placas de Peltier

A pastilha Peltier é um resfriador termoelétrico que tem como princípio de funcionamento um efeito físico de mesmo nome. A placa é usada em diversas aplicações, como refrigeração de água, desumidificadores e condensadores. O efeito Peltier consiste na produção de um gradiente de temperatura na junção de dois condutores quando os mesmos são submetidos a uma tensão. A variação na energia térmica da placa de Peltier é proporcional à corrente elétrica que passa pelo mesmo.

## 3.8 O projeto da planta acadêmica

A partir das especificações levantadas, obtém-se a seguinte especificação do projeto que será construído: uma estufa com controle de umidade e temperatura, onde os sensores e atuadores estarão ligados ao Arduino, que será a interface I/O que se comunicará, via Modbus, com o SoftPLC. O PLC rodará um programa feito no Codesys, com objetivo de controlar as

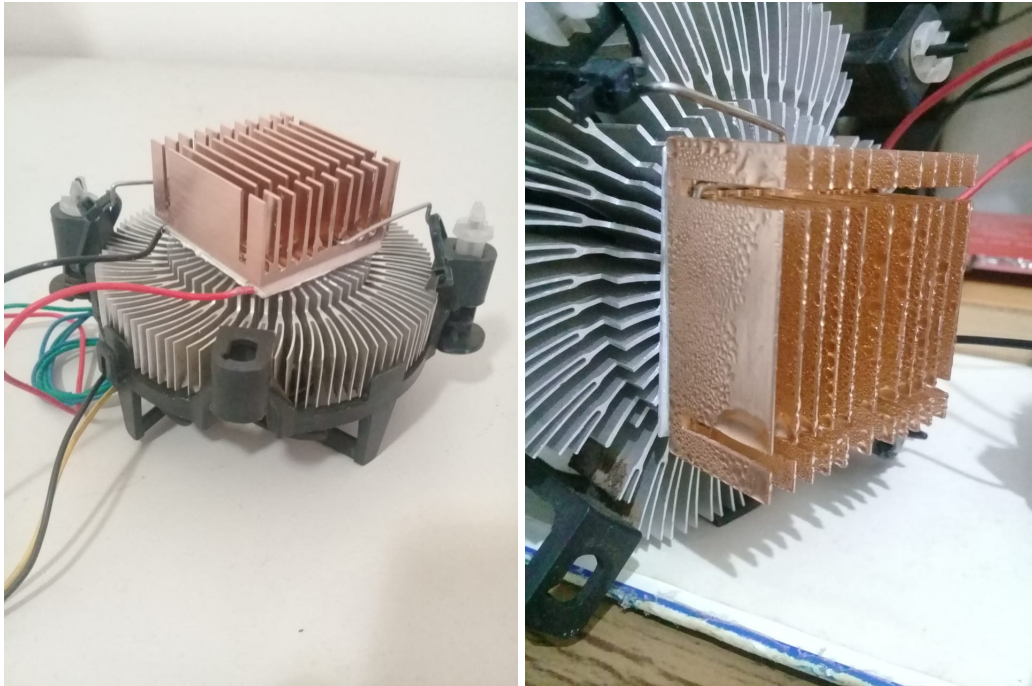


Figura 3.8: Condensador feito para retirada de umidade do ambiente.

variáveis do processo a partir a implementação de um bloco PID. O projeto também terá uma interface desenvolvida dentro do Codesys para visualização e manipulação das variáveis.

# Capítulo 4

## Desenvolvimento

Neste capítulo, será detalhada a montagem da planta, configuração dos dispositivos e desenvolvimento dos códigos usados na programação do Arduino e SoftPLC. No presente capítulo, também serão mostrados os problemas encontrados em cada etapa e as estratégias de solução que foram tomadas.

### 4.1 Etapas do Desenvolvimento

O projeto seguiu a metodologia *double diamond* para desenvolvimento, que consiste em ter um escopo grande, o objetivo desse método é mapear os estágios divergentes e convergentes dentro de um processo de design, com a intenção de encontrar uma solução real e assertiva em relação ao problema. O cronograma adotado foi o seguinte:

- Pesquisa inicial;
- Metodologia Double Diamond;
- Escolhas;
- Testes do Arduino usando Modbus;
- Teste do SoftPLC;
- Adicionando um dispositivo Modbus no projeto do Codesys;
- Teste comunicação Arduino - Codesys via Modbus;
- Desenvolvimento código usado no projeto parte Arduino;
- configuração I/O;
- mapeamento I/O no Codesys;
- Montagem do processo.;
- Teste;
- Coleta de dados;

- Modelagem caixa preta;
- Implementação de controle;
- Desenvolvimento de prática de ensino;
- Conclusão.

## 4.2 Desenvolvimento da Estrutura da Estufa

A estrutura física desse módulo acadêmico consiste em um ambiente fechado composto por: um primeiro volume maior em formato de paralelepípedo conectado a um segundo volume menor por meio um duto cilíndrico. O objetivo do volume maior é simular uma variedade de ambientes com temperatura e umidade controladas, como estufas e cabines de pintura, por exemplo. Baseado no modelo mostrado na figura 2.1, foi construída uma caixa sobre uma base de madeira de dimensões 24.0cm x 28.0cm, sob essa foi fixado com um parafuso a *heat bed* figura 3.5, evitando, assim, o contato direto da fonte de calor com a madeira deixando o contato totalmente com o ar.

As paredes do volume foram feitas de PVC, material com baixa condutividade térmica de dupla face com dutos entre as faces, além de isopor e EVA nas faces, isso com intuito de isolar o ar que está dentro da caixa com o ar do ambiente. Dentro da caixa, duas ventoinhas são responsáveis por forçar o fluxo pelos dutos, passando pelo desumidificador, uma delas faz a função de empurrar e a outra faz função de puxar o ar. O duto é composto de alumínio sanfonado e, por meio dele, existe o fluxo de ar. No volume superior, encontram-se os desumidificadores, que são compostos por dissipadores de calor conectados à face fria de uma placa de Peltier que tem resfriadores mecânicos conectados à face quente. A parte quente fica isolada do lado de fora do volume de desumidificação e a parte fria é responsável por condensar água e retirar a umidade do ar. A seguir nas figuras 4.1, 4.2 e 4.3 um modelo digital 3D do projeto é mostrado.

## 4.3 Etapa de configuração de software

Para o funcionamento do projeto, foi preciso o desenvolvimento de códigos diversos. Primeiramente, foi preciso criar uma lógica no Arduino, em que o microcontrolador lesse os valores do sensor DHT11 e escrevesse nos buffers Modbus e também os valores recebidos via protocolo MODEBUS nas saídas PWM do Arduino, que servem de sinal para os reguladores de tensão usados no projeto.

### 4.3.1 Arduino

No Arduino, o código feito tem a função de ler os sinais do sensor e escrever nas saídas PWM *Anlog 5* e *Analog 6*, de forma síncrona e periódica. Existem duas saídas que correspondem às saídas que vão nos reguladores de tensão dos atuadores e existe a leitura dos dois valores do sensor temperatura e umidade. A saída de dados do sensor está conectada à porta A0 do Arduino. Para realizar esse processo, duas bibliotecas principais são usadas, DHT\_U, para



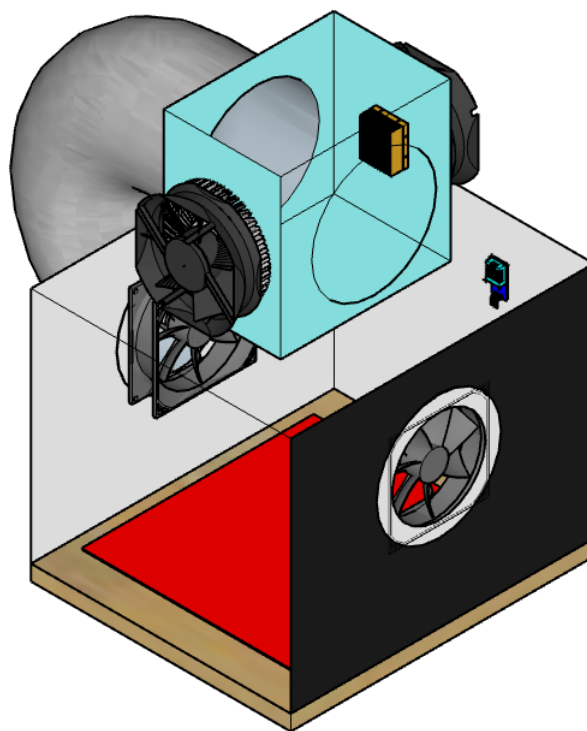


Figura 4.1: Vista em corte da estufa

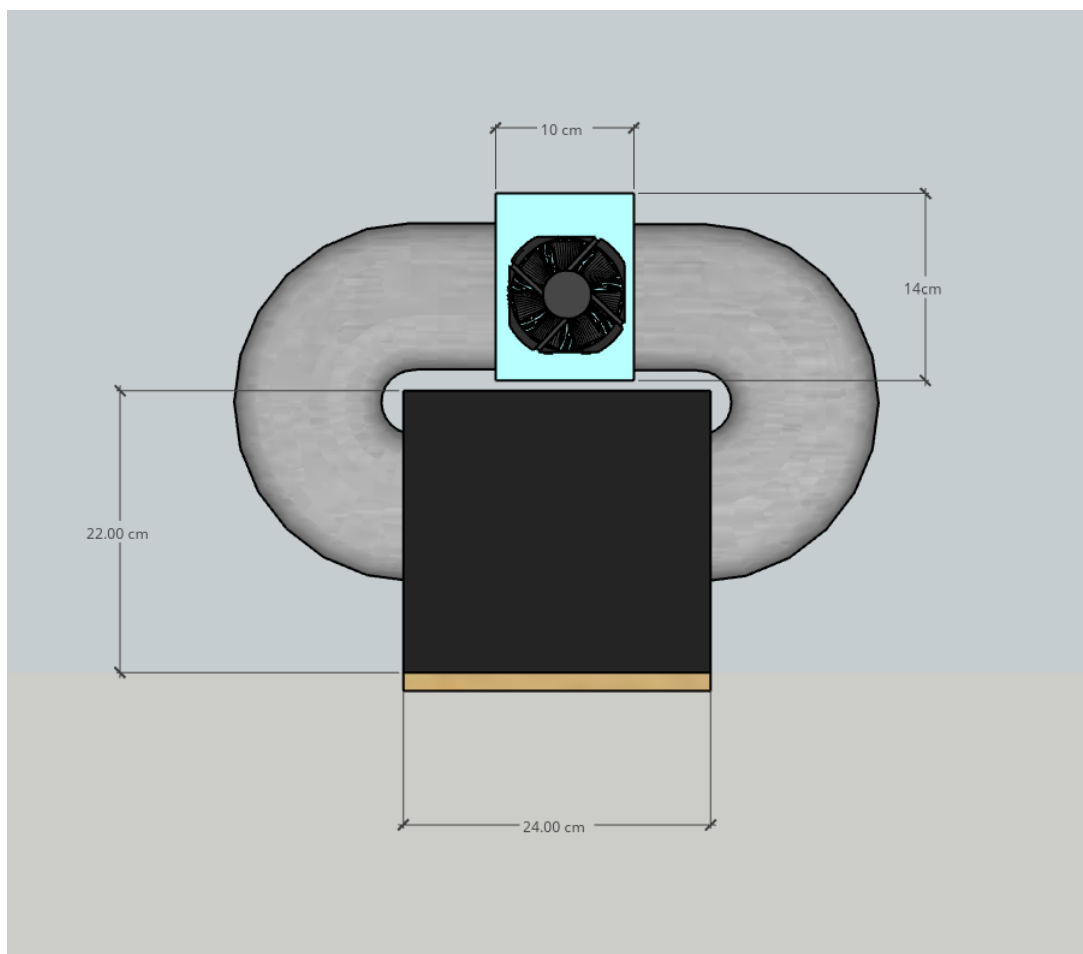


Figura 4.2: Vista lateral 1 da estufa.

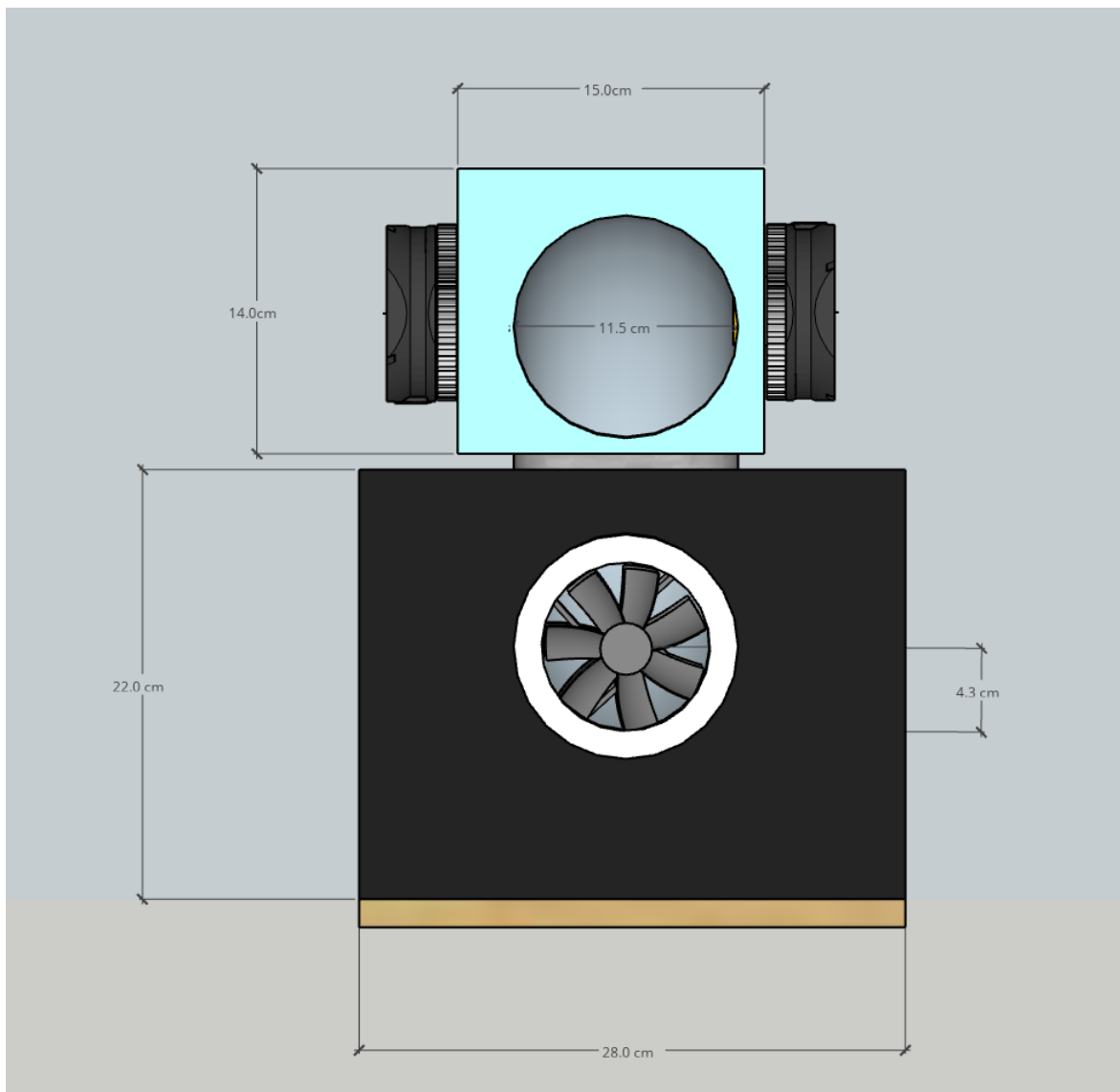


Figura 4.3: Vista lateral 2 da estufa em corte

uso do sensor, e `modbus_rtu_slave`, para configuração do Arduino como um slave Modbus e operação do mesmo.

A seguir, o código feito é apresentado:

```
#include "modbus_rtu_slave.h"
#include <DHT_U.h>

#define RE_DE 8
#define SLAVE_ADDRS 1
#define PWM_PIN1 6
#define PWM_PIN2 5
#define DHTPIN A0
#define DHTTYPE DHT11
#define SAMPLE_INTERVAL 200
DHT dht(DHTPIN, DHTTYPE);

unsigned long timer=0;

// -- Inicialização das entradas e saídas.
int in_01=0;
int out_01=0;
int out_02=0;

void setup()
{
  // Inicializa o modbus
  modbus_init(9600, RE_DE, SLAVE_ADDRS);

  // Inicializa o pino 6 e 5 PWM como saída
  pinMode(PWM_PIN1, OUTPUT);
  pinMode(PWM_PIN2, OUTPUT);

  // Inicializa a leitura do sensor de temperatura e umidade
  dht.begin();
}

// --Loop Principal do programa
void loop()
{
  modbus_check();
  ReadInput();
  WriteOutput();
}
```

Figura 4.4: Código que roda no arduino com Métodos de Leitura e Escrita dos dados no Modbus.

Inicialmente, fazemos a inclusão das bibliotecas necessárias citadas anteriormente. Em seguida, usa-se a série de constantes que são definidas. `RE_DE_8 SLAVE_ADDRS` define o endereço de slave que o Arduino terá. `PWM_PIN 1 e 2` definem quais serão os pinos de saída analógica do Arduino usados. `DHTPIN` define qual pino será usado como entrada de dados do sensor DHT e `DHTYPE` define o tipo de sensor dentro da família DHT, está sendo usado `SAMPLE_INTERVAL` para definir o intervalo de amostragem de checagem do sensor e escrita nos registradores do Modbus.

Em seguida, é preciso configurar o Arduino para usar a lib `DHT_U` e inicializar as variáveis que guardam os valores de entrada e saída. Na parte de setup, são feitas as inicializações do Modbus, configuração dos pinos como entrada ou saída e inicialização da leitura do sensor de umidade e temperatura.

No laço principal, temos a função `Modbus_check`, que faz a checagem de mensagem Modbus vindo da master Modbus. Em seguida, é chamada a função de `ReadInput`, que faz a leitura dos inputs no caso dos valores vindos do DHT11 e depois a função de escrita `WriteOutput`, que escreve a saída definida pelo master do Modbus nas saídas digitais do Arduino.

Temos aqui a função de leitura de valores, que faz uma leitura periódica das entradas, a cada período, com tempo igual a `SAMPLE_INTERVAL`, é feito o processo de leitura da umidade do sensor por meio da função `readHumiditu` da lib `DHT_U`, o valor é multiplicado por 100 devido a uma característica de transmissão de valores inteiros por via do Modbus. Assim, é possível conseguir uma precisão maior das variáveis que são enviadas para o Soft-PLC, onde o valor será convertido em real e dividido por 100, recuperando, assim, o valor inicial com a precisão de leitura de duas casas decimais. Ao final desse processo, é feito o *update* do registrador de número um com o valor. O mesmo é realizado para a temperatura com a diferença que o valor é armazenado no registrador de número 1.

Ao final da função, a variável auxiliar timer é *resetada* para que um novo ciclo de leitura e atualização de registradores só ocorra dentro do período de amostragem. A função de escrita tem o simples papel de ler os registradores onde o master escreve os valores das variáveis de controle (as variáveis que vão ter o valor da ação de controle) e escrever na saída analógica do Arduino.

## **Biblioteca para implementar comunicação Modbus**

### **4.3.2 Codesys**

Foi criado no Codesys um projeto que será gravado e rodará no SoftPLC. Nesse programa, temos a configuração do dispositivo Modbus, onde existe um master rodando que se conecta ao slave, que é o Arduino por meio da porta COM virtual que o Arduino cria ao se conectar via USB ao computador. Nesse master, o Arduino é configurado como um dispositivo slave, também duas leituras de registradores usando a função 3 do protocolo Modbus, para que os valores sejam gravados nos registradores 0 e 1 são configuradas. Usando a função de código 6 a escrita dos registradores 2 e 3 é configurada. Com todos os registradores devidamente guardando e escrevendo os valores, é feito um mapeamento desses valores para variáveis globais criadas na GVL (Global Variable List) do código. Também é configurado o tempo de timeout das mensagens do protocolo Modbus.

Existem 3 tarefas configuradas nos Codesys:

```
// -- Função de leitura das entradas.
void ReadInput() {
  // -- Condição para leitura no intervalo de amostragem definido
  if((millis() - timer)>SAMPLE_INTERVAL){

    //-----Chamadas de leitura de temperatura e umidade.
    float h = dht.readHumidity();
    int hum = (int)(h*100);
    modbus_update_holding(0,hum);

    float t = dht.readTemperature();
    int temp = (int)(t*100);
    modbus_update_holding(1,temp);

    timer = millis();
  }
}
```

Figura 4.5: Função de leitura das entradas.

```
// -- Função de escrita nas saídas.
void WriteOutput() {
  //---- Chamadas de escrita na saídas de PWM.
  out_01=modbus_read_holding(3);
  out_02=modbus_read_holding(4);
  analogWrite(PWM_PIN1,out_01);
  analogWrite(PWM_PIN2,out_02);
}
```

Figura 4.6: Função de escrita das saídas.

- *Error Check Task*: nessa tarefa, roda um programa feito para lidar com um problema que surgiu durante o desenvolvimento do projeto que era o paralisação da comunicação entre o softPLC e o Arduino depois que o primeiro erro ocorria entra a comunicação do Modbus. Para solucionar isso, foi criado um código que faz a leitura do do status do slave para checar se está em estado de erro. Ao detectar um erro, as seguintes variáveis são setadas para *true*: Slave.xReset, Slave.xAcknowledge, Slave.xTrigger, Master.xStop. Com isso, toda vez que um erro é encontrado um *reset* da comunicação é feito e as tarefas de leitura e escrita voltam a funcionar normalmente;
- *Main Task*: nessa tarefa, fica a lógica de controle criada para o programa. Nela que os valores lidos são divididos por 100 para voltarem à grandeza original com a precisão desejada;
- *Visu Task*: tarefa que é responsável por executar a IHM criada para controle manual e visualização de alarmes e valores das variáveis controladas.







# Capítulo 5

## Resultados

Após o desenvolvimento, montagem da estufa, configuração de softwares e dispositivos, foi possível fazer uma série de ensaios e definir os escopos para os quais a planta acadêmica pode ser utilizada. Durante a montagem e teste, também foram constatados problemas que vão ser apresentados a seguir e servirão como uma base para sugestão de trabalhos futuros. Com o módulo funcionando, duas ideias de práticas foram apresentadas e estão detalhadas ao final desse capítulo.

### 5.1 A Estufa

A montagem da estufa ocorreu bem e foi possível controlar as variáveis de processo planejadas. O resultado da montagem pode ser visto nas imagens 5.1 e 5.2 que se encontram nesse capítulo. As escolhas de alternativas de baixo custo em relação ao hardware e software utilizados atualmente nas plantas acadêmicas mostrou que são factíveis montagens mais baratas para o ensino prático de engenharia, podendo, assim, se pensar expansões dos laboratórios para uso de bancadas individualizadas ou até mesmo montagem de kits para o ensino remoto. Com a planta acadêmica pronta, foi possível levantar as seguintes características:

- A temperatura pode ser controlada para valores entre 0° C e 50°C;
- A variação máxima de umidade relativa que se pode obter é de 20% utilizando os condensadores;
- O controle simultâneo da umidade relativa e temperatura é afetado pelo valor da umidade relativa variar com a temperatura.

Durante o processo de configuração do módulo acadêmico, foi encontrado um erro na comunicação MODBUS de instabilidade, em que, ao ocorrer um erro de comunicação entre o nó Master configurado no SoftPLC e o nó Slave configurado no Arduino, as mensagens recebidas começavam a sinalizar erro 100% das vezes. Foi feita uma pesquisa de solução para o problema e a indicação encontrada era habilitar a reconexão automática nas configurações do dispositivo MODBUS dentro do Codesys, porém essa alteração não surtiu efeito no problema. A solução encontrada foi criar uma nova tarefa de verificação de erro para rodar no SoftPLC, que reinicia manualmente a comunicação entre os nós MODBUS e limpa o código de erro, iniciando, assim, novamente a comunicação.

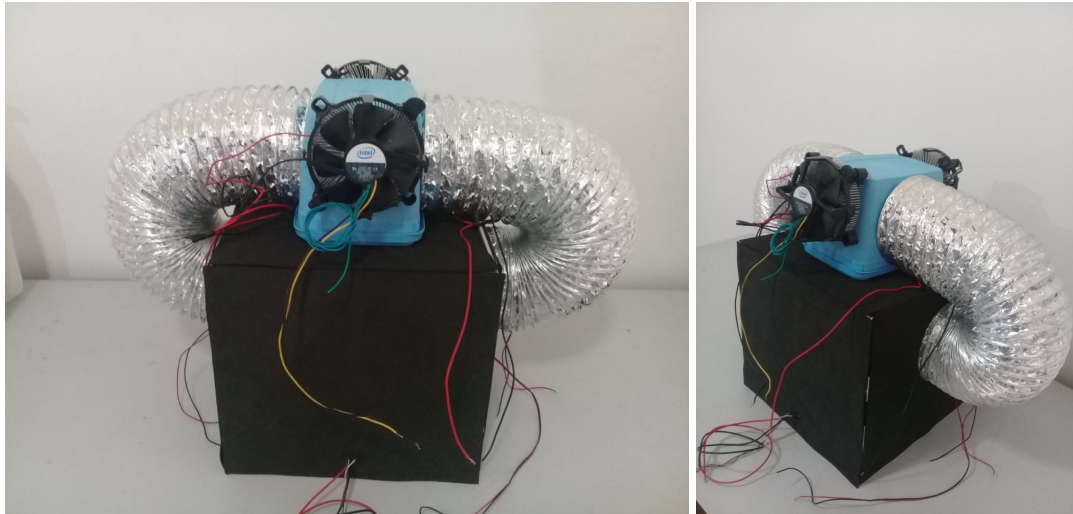


Figura 5.1: Planta acadêmica - Estufa com controle de Temperatura e Umidade.



Figura 5.2: Vista Interna - Planta acadêmica - Estufa com controle de Temperatura e Umidade

Outro aprendizado foi sobre o processo escolhido para que a dinâmica fosse mais rápida, ao final da montagem, a taxa de amostragem se mostrou rápida o suficiente para permitir o funcionamento de processos mais rápidos. A saturação do sinal controle da umidade acontece de forma muito rápida, pois o ganho desse processo é baixo, fazendo, assim, que não seja possível escolher especificações de controle muito robustas em que, o tempo de acomodação seja satisfatório para uma prática com duração de 60 minutos.

### 5.1.1 Preço

Um dos objetivos do trabalho era construir o módulo de ensino dentro do orçamento de até R\$ 400,00. Para se alcançar esse alvo de valor, os módulos e equipamentos escolhidos tiveram como um dos critérios de escolha o preço, além do uso de vários componentes que podem ser reaproveitados de sucata eletrônica. Componentes como ventiladores de circulação de ar, *cooler* para processador e dissipadores de alumínio foram reaproveitador de computadores velhos e não tiveram custo na montagem feita para o trabalho. A tabela 5.1 mostra a precificação dos componentes principais utilizados na montagem da estufa.

Componente	Preço	Quantidade
Arduino Uno	R\$ 40,00	1
Mesa Aquecida Heatbed Mk2b 12/24V	R\$ 65,00	1
Placa Peltier TEC1 12706	R\$ 38,00	2
Módulo Sensor DHT11	R\$ 11,50	1
Módulo PWM MOSFET D4184	R\$ 7,50	3
Fonte 12 V 30 A	R\$ 50,00	1
Tubo Flexível Alumínio - 1,5 metros	R\$ 36,00	1
Folha Forro PVC dupla - 200mm x 4m	R\$ 25,00	1
Ventiladores 12V - 8mm x 8mm	R\$ 7,00	2
Dissipador de alumínio - 25mm x 25mm	R\$ 10,00	2
Cooler para processador 12V - 12mm x 12mm	R\$ 30,00	2
<b>Total</b>	<b>R\$ 390,00</b>	<b>-</b>

Tabela 5.1: Tabela de custo planta acadêmica.

## 5.2 Proposta módulos de ensino

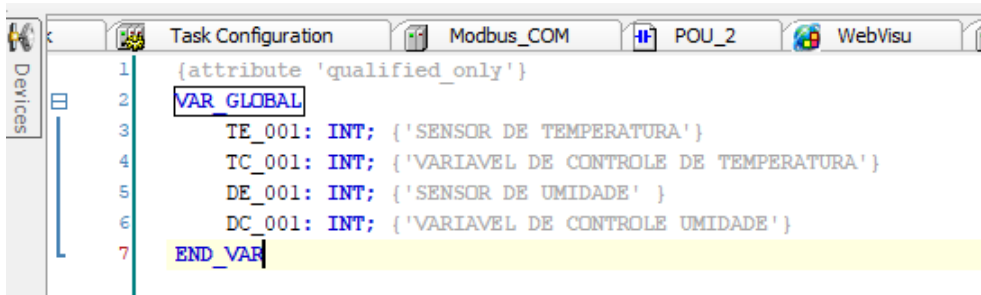
Após a montagem e levantamento de característica da planta acadêmica, foi possível experimentar e, desta maneira, sugerir o desenvolvimento de práticas para utilização da mesma. A partir da leitura das ementas de disciplinas do curso de Engenharia de Controle e Automação, as disciplinas de Informática Industrial e Laboratório de Controle I foram escolhidas e foi elaborada uma sugestões de roteiros parada cada uma usando a estufa.

### 5.2.1 Prática de laboratório informática industrial

A primeira abordagem didática para o módulo foi pensada como uma prática da disciplina de informática industrial. A ementa da disciplina traz como temas: Controladores Lógicos Programáveis, Linguagens de programação de CLPs, Sistemas SCADA, Projeto de automação utilizando CLPs e sistemas supervisórios. A ementa e análise os resultados apresentados anteriormente que demonstraram que é possível utilizar a planta acadêmica como consolidação e aplicação do conceitos da disciplina.

O objetivo da prática é a configuração das TAGs no CLP para representar os atuadores e sensores da planta seguindo a nomenclatura padrão, montagem de uma interface IHM simples para interagir com as variáveis criadas, e ao final, configuração da lista de símbolos e de um cliente OPC para receber e escrever dados na planta.

Os materiais necessários para a prática são apenas o módulo e um computador com Co-desys instalado.



```
1 {attribute 'qualified_only'}
2 VAR_GLOBAL
3   TE_001: INT; {'SENSOR DE TEMPERATURA'}
4   IC_001: INT; {'VARIAVEL DE CONTROLE DE TEMPERATURA'}
5   DE_001: INT; {'SENSOR DE UMIDADE'}
6   DC_001: INT; {'VARIAVEL DE CONTROLE UMIDADE'}
7 END_VAR
```

Figura 5.3: Configuração das TAGs como variáveis globais.

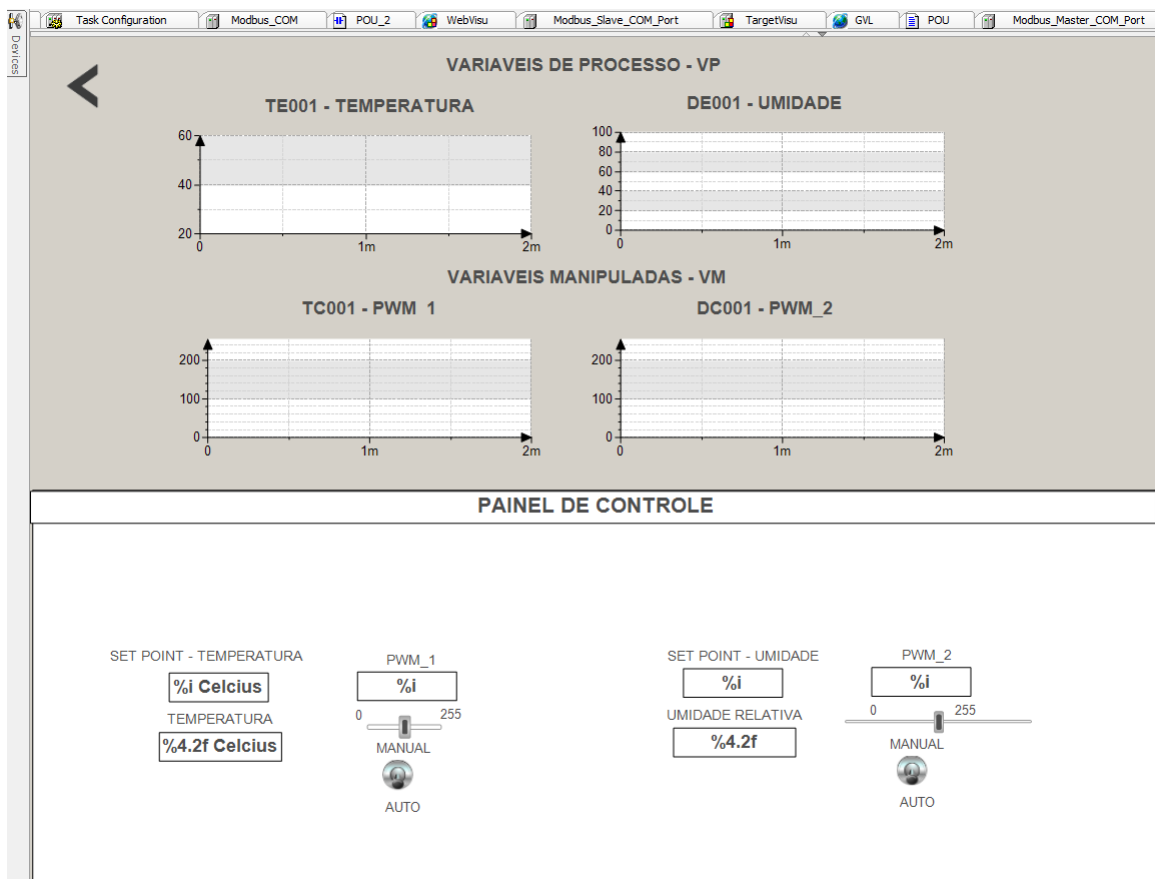


Figura 5.4: IHM para controlar a partir da interface a planta acadêmica.

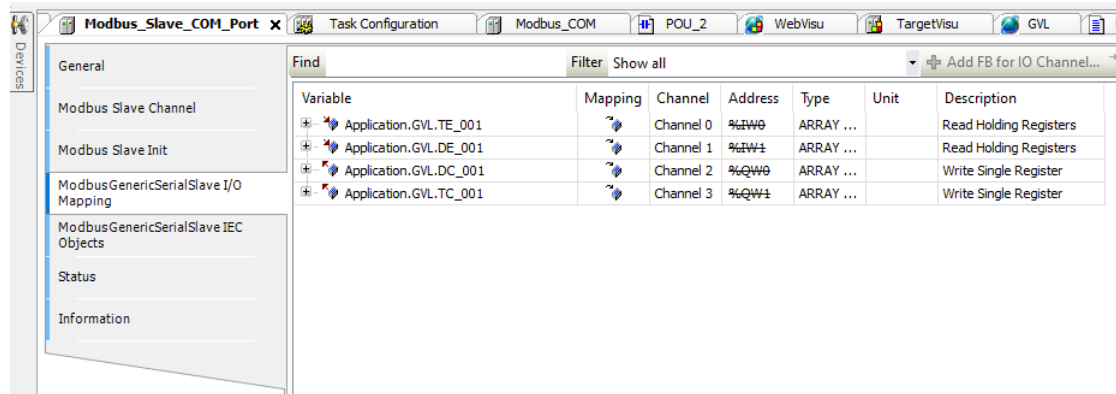


Figura 5.5: Mapeando as entradas e saídas configuradas via MODBUS no Arduino para as variáveis globais criadas.

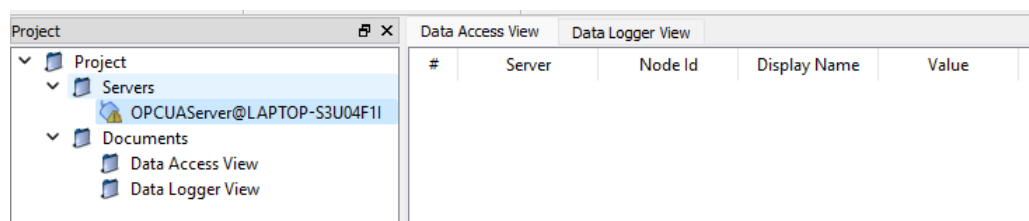


Figura 5.6: Servidor OPC para manipular as variáveis manipuladas e monitor as variáveis de processo.

## 5.2.2 Prática de laboratório Controle

Para a disciplina de Laboratório de Controle, a sugestão feita é uma prática em que o objetivo final do aluno é implementar diferentes estratégias de controle para as variáveis de processo da planta. Os conhecimentos necessários para se executar um possível roteiro seriam, modelagem de sistemas, projeto de controladores PID e conhecimento em informática industrial para uso de programação de Function Blocks (FB). A primeira parte da prática consistiria em caracterização do sistema, em que a sugestão é utilizar a caracterização via resposta ao degrau em malha aberta, obtendo, assim, a função de transferência relativa as duas malhas isoladas.

Para fazer essa caracterização, é preciso coletar os dados ao se aplicar um degrau. Após isso, seria feita a etapa da modelagem do sistema, a sugestão é que os alunos usem o MATLAB e as ferramentas de identificação de sistemas, além do conhecimento prévio, para obter a melhor função de transferência que represente a planta. Com o modelo do sistema pronto, entra a parte de projeto de controle, o objetivo é que cada aluno projete dois controladores, um para cada sistema, com características diferentes, por exemplo: um controlador sem *overshoot* com dinâmica mais lenta e outro com dinâmica mais rápida que não precisa evitar *overshooting*.

Modelo encontrado no sistema de aquecimento da estufa:

$$\frac{0.0001857s - 3.53 \cdot 10^{-6}}{s^2 + 0.01153s + 3.872 \cdot 10^{-5}}$$

Modelo para o sistema de retirada de umidade da estufa:

$$\frac{2.481 \cdot 10^5 s - 5.206 \cdot 10^{-6}}{s^2 + 0.03312s + 0.000173}$$

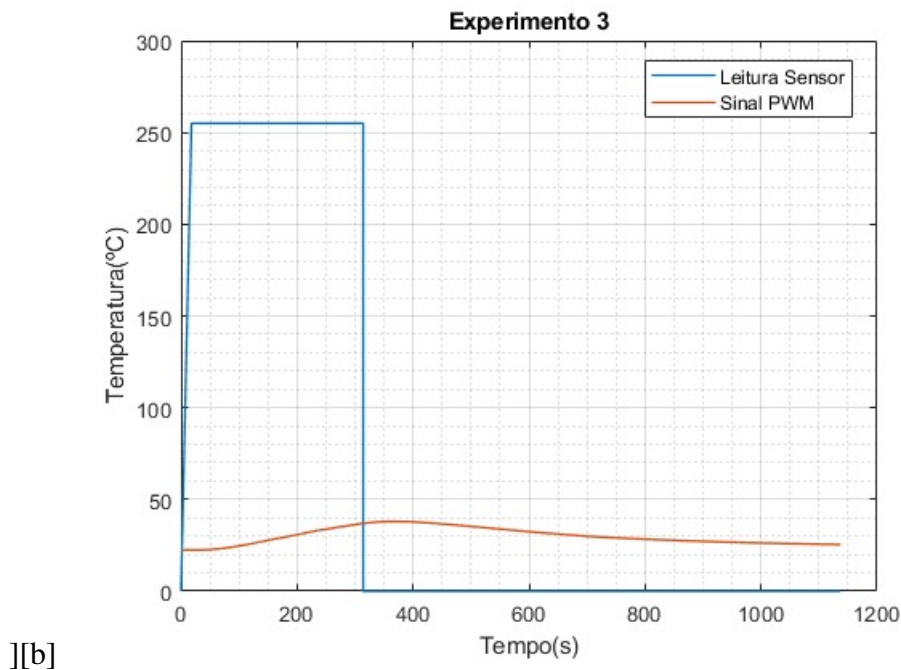


Figura 5.7: Dados da reposta a um pulso do sistema de aumento da temperatura - Experimento 1

O primeiro controlador é mais robusto, porém com a condição de não ter sobressinal da variável do processo (VP), o segundo um controle mais agressivo, onde a VP pode oscilar. É importante, nessa etapa, o aluno observar as limitações dos atuadores que podem saturar e não conseguir implementar o controlador projetado. Ao final da prática, os alunos implementariam os controladores projetados e fariam um teste para ver se os parâmetros e requisitos estabelecidos foram atendidos.

Na figura 5.7 é possível ver os dados coletados via OPC plotados em um gráfico do Matlab, as figuras seguintes mostram o resultado do levantamento das funções de transferências que correspondem ao sistema da planta academia, tanto para o laço de controle de umidade quanto o laço de controle de temperatura.

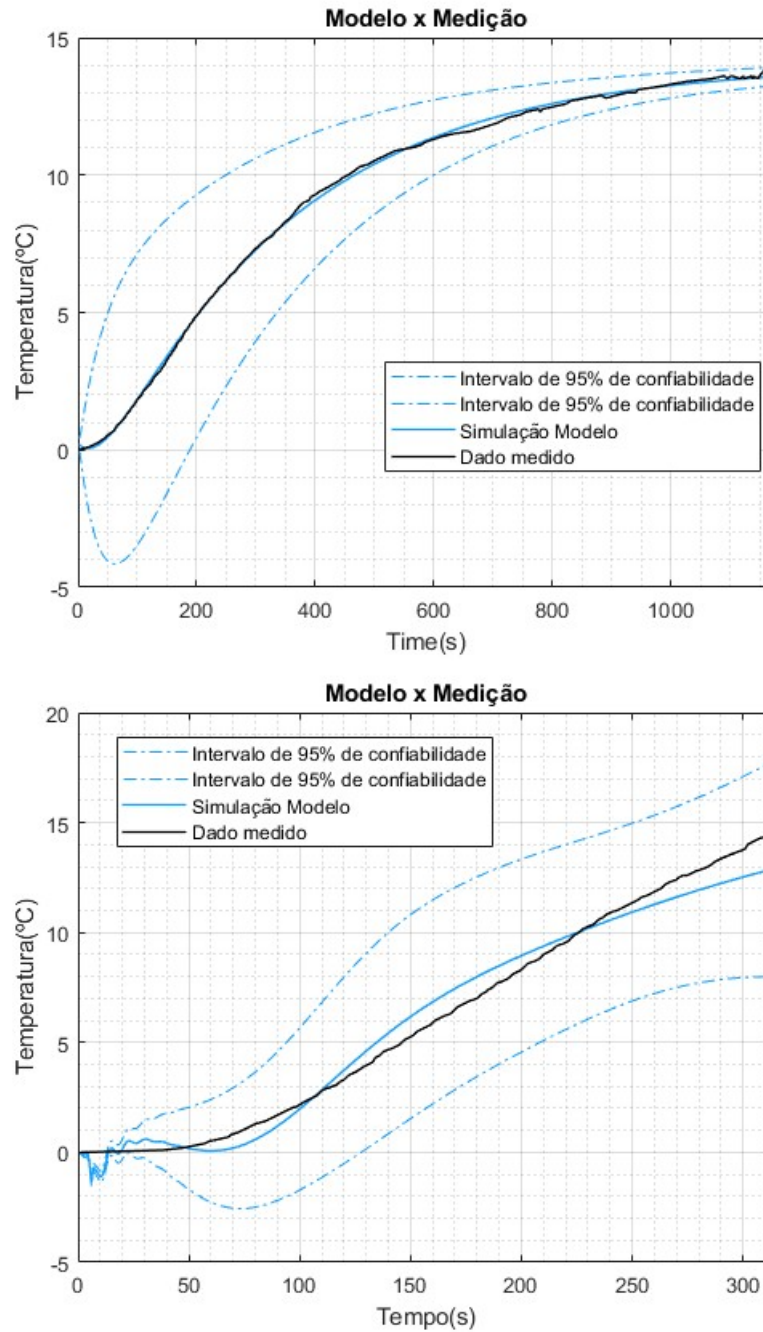


Figura 5.8: Comparação entre modelo e dados coletados para o sistema de aumento de temperatura.

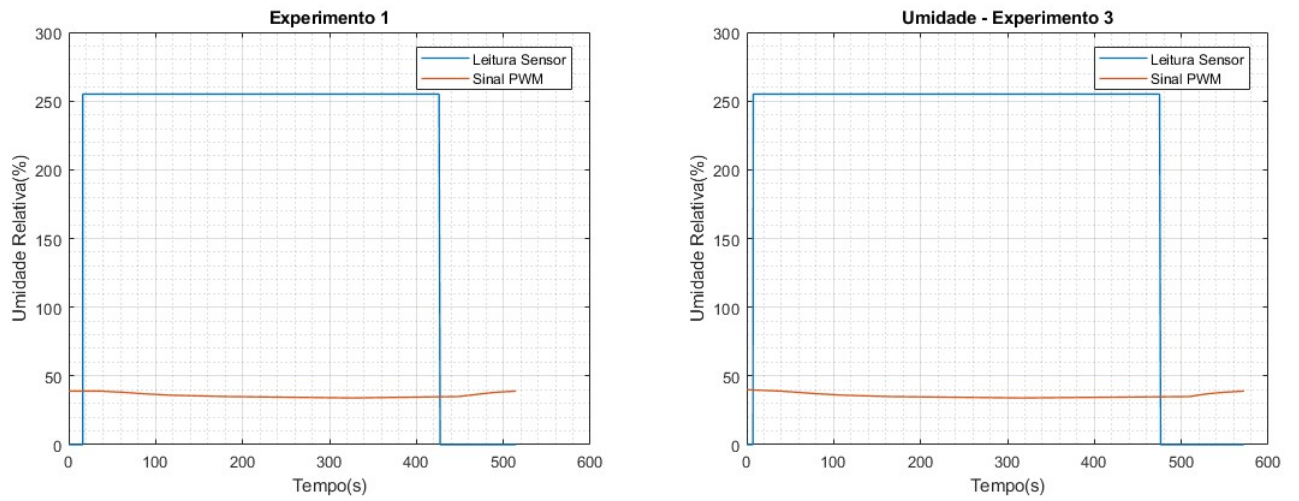


Figura 5.9: Dados da resposta a um pulso do sistema de redução de umidade

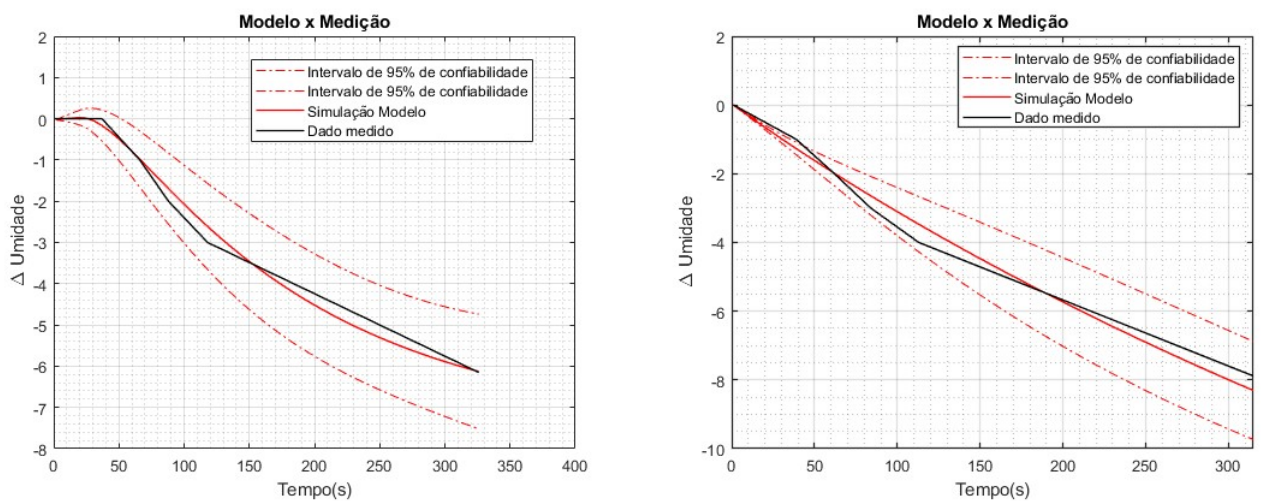


Figura 5.10: Comparação entre modelo e dados coletados para o sistema de redução de umidade.



Com os modelos que satisfazem um grau de confiabilidade pre estabelecido é possível projetar um controlador PID par as malhas como é mostrado na figura 5.11. Após obter os parâmetros do controlador é possível editar os parâmetros do controlador desejado na interface criada para que o aluno possa acompanhar os resultados de esforço de controle no laço. Na figura 5.12 é possível ver a interface onde o aluno irá inserir os valores das constantes do PID e acompanhar os valores de cada nó da malha.

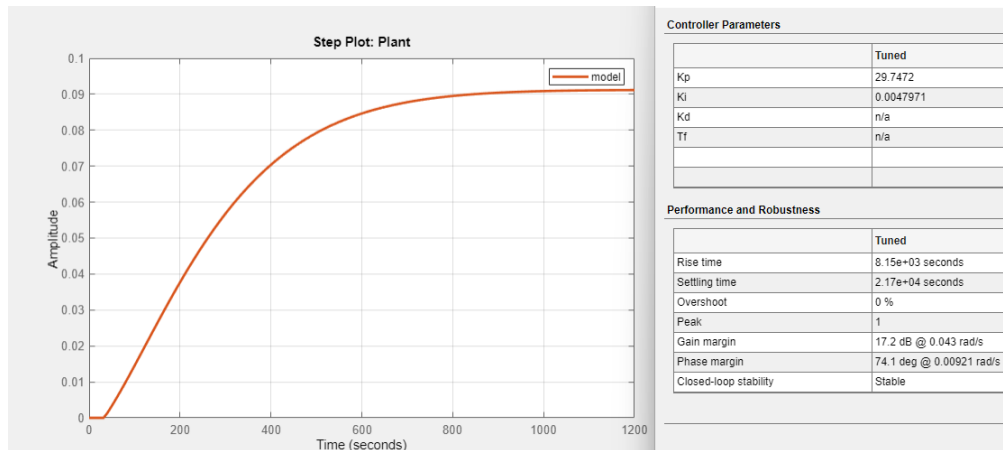


Figura 5.11: Tela do PID Tuner com o resultado do controlador projetado para a malha de aquecimento.

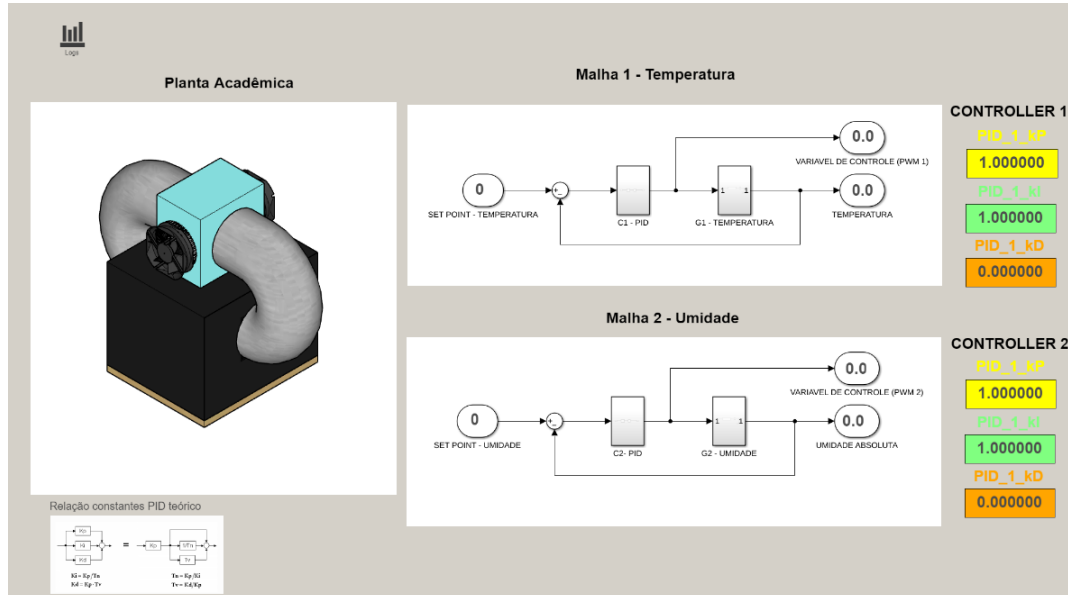


Figura 5.12: Interface interativa para manipular valores das constantes do PID.



# Capítulo 6

## Conclusões

A proposta deste trabalho era elaborar uma alternativa de baixo custo para ensino prático de controle e automação, construir esse módulo proposto e propor roteiros de estudo usando essa planta acadêmica. Ao final da construção do módulo, a elaboração dos roteiros se mostrou possível e duas sugestões de práticas foram adicionadas aos resultados, mostrando que, dentro do contexto de restrição de acesso aos espaços físicos da universidade e buscando alternativas a simulações, a planta elaborada seria uma boa alternativa. Por isso, ao analisar os resultados obtidos por este trabalho, foi possível concluir que existem alternativas mais baratas para a realização de aulas práticas sem perder muita qualidade.

Também se destacou um ponto percebido durante o desenvolvimento do trabalho, que o maior valor no protótipo desenvolvido é a característica do uso do Arduino como interface de I/O se comunicando com o Codesys. A partir do mapeamento das entradas e saídas da placa do microcontrolador, é possível implementar uma grande diversidade de processos, dependendo apenas dos sensores e atuadores disponíveis, essa possibilidade abriu um leque grande para futuros trabalhos utilizando esta mesma base. A planta acadêmica construída simula um tipo de processo encontrado na indústria, com uma dinâmica lenta devido a restrições impostas pelos atuadores usados e características físicas do próprio processo proposto.

### 6.1 Propostas de Continuidade

Como sugestão para continuidade desse projeto, pode-se selecionar duas vias. A primeira seria a de estudo do impacto do uso do módulo desenvolvido no ensino, podendo ser elaborado um trabalho como metodologia de pesquisa com teste A B, experimento e levantamento de dados com pessoas que tiveram contato com o projeto e proposta de roteiros acadêmicos embasados na literatura a partir do que o projeto se mostrar capaz. A segunda já é voltada a melhorias do módulo desenvolvido, tanto na solução de problemas que foram citados durante o texto como na performance, visando tornar o módulo mais próximo do ideal para o uso no ensino dentro de um laboratório ou remotamente. Dentre essas correções e melhorias estão:

- Elaboração de uma placa de circuito impresso para agregar todos os componentes eletrônicos do módulo;
- Pesquisa e desenvolvimento de um atuador para o controle de umidade mais eficiente;

- Melhoria nos códigos desenvolvidos para diminuir a taxa de erro e conseguir fazer a base SoftPLC + Arduino ser usada para processos de dinâmica mais rápida.

# Referências Bibliográficas

- [1] Escribens Mauricio Sabzalieva Emma Vieira do Nascimento Daniele Yerovi Clarisa Abdrasheva, Dana. *The impact of COVID-19 on global higher education. International Higher Education*. 2022.
- [2] C H Bamorim, F S Naegeli. Integração teoria e prática no ensino de engenharia: a construção de um novo modelo didático. *II Encontro de Ensino de Engenharia - UFRJ - Rio de Janeiro*, 1996.
- [3] A.R. Braga, Lúcio Passos, C. Jr, and Carmela Polito Braga. Projeto e implementação de um módulo didático para abordagem integrada de instrumentação, controle e automação. 01 2008.
- [4] André Luiz Nunes Do Nascimento and Sandey Basso. Câmara climática automatizada, desenvolvimento de uma câmara para controle de temperatura e umidade. *Curso de engenharia de controle e automação, UTFP(UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)*, 2018.
- [5] Charles Hodges, Stephanie Moore, Barb Lockee, Aaron Bond, et al. As diferenças entre o aprendizado online e o ensino remoto de emergência. *Revista da escola, professor, educação e tecnologia*, 2, 2020.
- [6] Paulo Alexandre Martin and Fuad Kassab Junior. Uso de trocador de calor como ferramenta didática para o ensino de modelagem e sistemas de controle. *Revista de Ensino de Engenharia*, 25(2), 2008.
- [7] G Pavlogeorgatos. Environmental parameters in museums. *Building and Environment*, 38(12):1457–1462, 2003.
- [8] JA Rossiter, SA Pope, B LI Jones, and JD Hedengren. Evaluation and demonstration of take home laboratory kit. *IFAC-PapersOnLine*, 52(9):56–61, 2019.
- [9] Amir Hossein Shourideh, Wael Bou Ajram, Jalal Al Lami, Salem Haggag, and Abraham Mansouri. A comprehensive study of an atmospheric water generator using peltier effect. *Thermal Science and Engineering Progress*, 6:14–26, 2018.
- [10] Jacobo Sáenz, Jesús Chacón, Luis De La Torre, Antonio Visioli, and Sebastián Dormido. Open and low-cost virtual and remote labs on control engineering. *IEEE Access*, 3:805–814, 2015.
- [11] Ismail Uyanik and Bahadir Catalbas. A low-cost feedback control systems laboratory setup via arduino–simulink interface. *Computer Applications in Engineering Education*, 26(3):718–726, 2018.