

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* SABARÁ  
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Arthur Henrique Pimenta Silva

**AUTOMAÇÃO HOSPITALAR: SUPERVISÃO E CONTROLE PARA  
GESTÃO EFICIENTE**

Sabará  
2024

ARTHUR HENRIQUE PIMENTA SILVA

## **AUTOMAÇÃO HOSPITALAR: SUPERVISÃO E CONTROLE PARA GESTÃO EFICIENTE**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus Sabará*, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

**Orientador:** Prof. Dr. Rodrigo Hiroshi Murofushi

**Coorientador:** Prof. Me. Luiz Guilherme Hilel Drumond  
Silveira

Sabará  
2024

Silva, Arthur Henrique Pimenta

S586a

Automação hospitalar: supervisão e controle para gestão eficiente [manuscrito]. / Arthur Henrique Pimenta Silva. - 2024.

69 f. : il.

Orientação: Prof. Dr. Rodrigo Hiroshi Murofushi.

Coorientador: Prof. Me. Luiz Guilherme Hilel Drumond Silveira.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Minas Gerais, *Campus* Sabará.

1. Hospitais – Automação. – Monografia. 2. Hospitais – Administração. – Monografia. 3. Sistemas de controle supervisão. – Monografia. 4. Software (Eclipse E3). – Monografia. I. Murofushi, Rodrigo Hiroshi. II. Silveira, Luiz Guilherme Hilel Drumond. III. Instituto Federal de Minas Gerais, *Campus* Sabará. IV. Bacharelado em Engenharia de Controle e Automação. V. Título.

CDU 681.3.06

César dos Santos Moreira / CRB6-2229  
Biblioteca do IFMG *Campus* Sabará



**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS**  
**Campus Sabará**  
**Diretoria de Ensino, Pesquisa e Extensão**  
**Conselho de Área - Controle e Processos Industriais**  
Rodovia MGC 262, Km 10 - Bairro Sobradinho - CEP 34590-390 - Sabará - MG  
- www.ifmg.edu.br

## **ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO**

Ao dia 09 do mês de outubro do ano de 2024, às 21:00 horas, sob a presidência de Rodrigo Hiroshi Murofushi, o discente **Arthur Henrique Pimenta Silva** do Curso de Engenharia de Controle e Automação, R.A nº 0050368 do IFMG *campus* Sabará, defendeu o Trabalho de Conclusão de Curso (TCC) intitulado “**Automação Hospitalar: Supervisão e Controle para Gestão Eficiente**” e foi avaliado com a nota final média de 91 pontos, que está condicionada ao cumprimento dos procedimentos pós-defesa do TCC.

Compuseram a Banca Examinadora:

Membro 1: Dr. Rodrigo Hiroshi Murofushi -IFMG *campus* Sabará (orientador),

Membro 2: Dra. Mariella Maia Quadros - IFMG *campus* Sabará,

Membro 3: Me. Paulo Aparecido Amaral Junior - IFMG *campus* Sabará.

O discente deverá apresentar o trabalho com as devidas modificações em formato pdf, até 30/10/2024 à Coordenação de TCC e fazer o depósito no repositório institucional de TCC do IFMG. O não cumprimento dos procedimentos pós-defesa de TCC até a data estipulada implica no não cumprimento das horas referentes aos componentes curriculares de TCC I e II.

Sabará, 09 de outubro de 2024.



Documento assinado eletronicamente por **Rodrigo Hiroshi Murofushi, Professor**, em 10/10/2024, às 22:22, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Mariella Maia Quadros, Professora EBTT**, em 30/10/2024, às 18:20, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Paulo Aparecido Amaral Junior, Professor EBTT**, em 31/10/2024, às 09:31, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2068596** e o código CRC **BBD73173**.

## RESUMO

A automação hospitalar tem se mostrado uma área em rápida expansão, oferecendo soluções que otimizam a gestão de recursos e a qualidade das atividades. Com o avanço da tecnologia e o desenvolvimento de sistemas de supervisão, é possível integrar diversos processos, desde o controle de aplicações específicas até a gestão completa de um estabelecimento, resultando em maior eficiência e segurança para o sistema. Nesse contexto, esse trabalho apresenta o processo de desenvolvimento de um protótipo em escala reduzida para simular a gestão automatizada de um hospital. O projeto utiliza um arduino e a integração de sensores por meio de uma comunicação em *Modbus*, possibilitando a implementação de um sistema mais abrangente. O processo pode ser monitorado a partir de um sistema supervisório que se comunica com o *software* Elipse E3. Essa integração permite ao hospital uma melhor gestão sob o que está acontecendo em toda a estrutura, diminuindo os obstáculos enfrentados por toda a rede, bem como oferece melhor conforto para os hospitalizados. Para tomada de decisões no trabalho foram utilizadas duas metodologias principais, sendo elas o método do Duplo Diamante e a Matriz GUT, dessa forma pode-se delimitar melhor os pontos focais que serão trabalhados no projeto.

**Palavras-chave:** Automação hospitalar; Gestão hospitalar; Elipse E3; Sistema Supervisório.

## ABSTRACT

Hospital automation has shown itself to be a rapidly expanding area, offering solutions that optimize resource management and the quality of activities. With the advancement of technology and the development of supervision systems, it is possible to integrate several processes, from the control of specific applications to the complete management of an establishment, resulting in greater efficiency and safety for the system. In this context, this work presents the process of developing a prototype on a reduced scale to simulate the automated management of a hospital. The project uses an Arduino and the integration of sensors through Modbus communication, enabling the implementation of a more comprehensive system. The process can be monitored from a supervisory system that communicates with the software Elipse E3. This integration allows the hospital to better manage what is happening in the entire structure, reducing the obstacles faced by the entire network, as well as offering greater comfort for hospitalized patients. Two main methodologies were used to make decisions at work, namely the Double Diamond method and the GUT Matrix, in this way it is possible to better define the focal points that will be worked on in the project.

**Keywords:** Hospital automation; Hospital management; Elipse E3; Supervisory System.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Uso de computador e internet pelos estabelecimentos de saúde em 2021. . . . .	9
Figura 2 – Total de estabelecimentos de saúde que utilizaram Internet em 2021. . . . .	10
Figura 3 – Arquitetura da placa Arduino. . . . .	16
Figura 4 – Estrutura de <i>Hardware</i> do CLP. . . . .	17
Figura 5 – Estrutura CLP. . . . .	18
Figura 6 – Terminais do sensor LM35. . . . .	20
Figura 7 – Diagrama do circuito de conexão do sensor LM35. . . . .	20
Figura 8 – Sensor Infravermelho Reflexivo de Obstáculo. . . . .	21
Figura 9 – Sensor de Gás MQ-2 e seus terminais de conexão. . . . .	22
Figura 10 – Sensor RFID. . . . .	23
Figura 11 – Conexão dos pinos ICSP do Arduino. . . . .	23
Figura 12 – Servo Motor MG996R. . . . .	24
Figura 13 – Fases do Método do Duplo Diamante. . . . .	27
Figura 14 – Dimensões das partes do protótipo do hospital. . . . .	30
Figura 15 – Vista isométrica do quarto do hospital. . . . .	31
Figura 16 – Protótipo na versão final. . . . .	31
Figura 17 – Vista superior do protótipo. . . . .	32
Figura 18 – Dimensões das partes do protótipo da cama. . . . .	33
Figura 19 – Parafuso de rosca sem fim, porca e rolamento. . . . .	33
Figura 20 – Vista geral da montagem final da cama. . . . .	34
Figura 21 – Protótipo da cama na versão final. . . . .	34
Figura 22 – Inserindo o <i>Driver</i> Modbus.dll. . . . .	36
Figura 23 – Configurando o <i>Driver</i> Modbus.dll. . . . .	37
Figura 24 – Selecionando a camada física trabalhada. . . . .	38
Figura 25 – Estabelecendo a porta serial e a taxa de baud. . . . .	39
Figura 26 – Adicionando Tags à aplicação. . . . .	40
Figura 27 – Tags internas e demo. . . . .	41
Figura 28 – Alarmes configurados. . . . .	42
Figura 29 – Tela de login. . . . .	43
Figura 30 – Tela de monitoramento principal. . . . .	44
Figura 31 – Indicação do acionamento dos alarmes no sistema supervisório . . . . .	44
Figura 32 – Tela da lista de médicos. . . . .	45
Figura 33 – Tela dos quartos. . . . .	45
Figura 34 – Tela de cirurgia e configuração da cama. . . . .	46
Figura 35 – Tela da farmácia. . . . .	47
Figura 36 – Tela dos quartos. . . . .	47

## LISTA DE TABELAS

Tabela 1 – Blocos de modelos de dados <i>Modbus</i> . . . . .	15
Tabela 2 – Funções sobre elementos do protocolo <i>Modbus</i> . . . . .	15
Tabela 3 – Comparação entre Arduino e CLP. . . . .	19
Tabela 4 – Heurísticas de Nielsen. . . . .	26
Tabela 5 – Desenvolvimento da Matriz GUT. . . . .	29

## LISTA DE ABREVIATURAS E SIGLAS

CLP	Controlador Lógico Programável
IoT	Internet das Coisas
UBS	Unidades Básicas de Saúde
NIC.br	Núcleo de Informação e Coordenação do Ponto BR
CGI.br	Comitê Gestor da Internet no Brasil
Cetic.br	Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação
RFID	<i>Radio Frequency Identification</i>
PEP	Prontuário Eletrônico do Paciente
IA	Inteligência Artificial
GM	General Motors
SCADA	<i>Supervisory Control and Data Acquisition</i>
OMS	Organização Mundial da Saúde

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
<b>1.1</b>	<b>Objetivos</b>	<b>7</b>
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>7</i>
<i>1.1.2</i>	<i>Objetivos específicos</i>	<i>8</i>
<b>1.2</b>	<b>Justificativa</b>	<b>8</b>
<b>1.3</b>	<b>Organização do Texto</b>	<b>11</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>12</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
<b>3.1</b>	<b>Banco de Dados</b>	<b>14</b>
<b>3.2</b>	<i>Modbus</i>	<i>14</i>
<b>3.3</b>	<b>Dispositivos</b>	<b>15</b>
<i>3.3.1</i>	<i>Arduino</i>	<i>15</i>
<i>3.3.2</i>	<i>Controladores Lógicos Programáveis</i>	<i>17</i>
<i>3.3.3</i>	<i>Sensor de temperatura LM35</i>	<i>19</i>
<i>3.3.4</i>	<i>Sensor Infravermelho Reflexivo de Obstáculo</i>	<i>20</i>
<i>3.3.5</i>	<i>Sensor de gás MQ-2</i>	<i>21</i>
<i>3.3.6</i>	<i>Sensor de Radio Frequency Identification (RFID)</i>	<i>22</i>
<i>3.3.7</i>	<i>Servo Motor MG996R</i>	<i>24</i>
<b>3.4</b>	<b>Internet das Coisas</b>	<b>25</b>
<b>3.5</b>	<b>Heurísticas de Usabilidade de Nielsen</b>	<b>25</b>
<b>3.6</b>	<b>Sistemas Supervisórios</b>	<b>26</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>27</b>
<b>4.1</b>	<b>Método do Duplo Diamante</b>	<b>27</b>
<b>4.2</b>	<b>Matriz GUT</b>	<b>28</b>
<b>5</b>	<b>DESENVOLVIMENTO E RESULTADOS</b>	<b>30</b>
<b>5.1</b>	<b>Maquete do hospital</b>	<b>30</b>
<b>5.2</b>	<b>Projeto da cama hospitalar</b>	<b>32</b>
<b>5.3</b>	<b>Construção do Sistema Supervisório</b>	<b>35</b>
<i>5.3.1</i>	<i>Protocolo e Driver de comunicação</i>	<i>35</i>

5.3.2	<i>Configuração das Tags de comunicação</i>	39
5.3.3	<i>Tags de comunicação internas e tags demo</i>	41
5.3.4	<i>Configuração dos alarmes</i>	41
5.3.5	<i>Desenvolvimento das telas</i>	42
[black]5.3.5.1	<i>Tela de Login</i>	42
[black]5.3.5.2	<i>Tela de Monitoramento</i>	43
[black]5.3.5.3	<i>Tela dos Médicos</i>	44
[black]5.3.5.4	<i>Tela dos Quartos</i>	45
[black]5.3.5.5	<i>Tela de Cirurgia</i>	46
[black]5.3.5.6	<i>Tela da Farmácia</i>	46
[black]5.3.5.7	<i>Tela de Alarmes</i>	47
6	<b>CONCLUSÃO</b>	<b>49</b>
6.1	<b>Trabalhos futuros</b>	49
	<b>REFERÊNCIAS</b>	<b>51</b>
7	<b>APÊNDICE A - CÓDIGO ARDUINO PARA A COMUNICAÇÃO COM O SISTEMA SUPERVISÓRIO</b>	<b>54</b>

# 1 INTRODUÇÃO

A automação no ambiente hospitalar é uma área que tem apresentado diversas mudanças para o cenário, a partir do momento em que a tecnologia tem se desenvolvido de forma exponencial ao longo dos anos, trazendo vantagens e facilidades para a área da saúde. Juntamente com a evolução da tecnologia e inovação, a automação tem revolucionado setores da saúde, tornando-os cada vez mais inteligentes, eficientes e interconectados. Atualmente, vários processos automatizados são adaptados e aplicados no setor hospitalar, por exemplo, no hospital Sírio Libanês, em São Paulo, que utiliza um sistema robotizado para operar a farmácia (FALCAO, 2015). Já o Hospital Marina Salud de Dénia, na Espanha, atingiu um estágio de reconhecimento como *paperless*, pela conquista eficaz na utilização de sistemas aptos para fazer a gestão de documentos de processos sem o uso do papel, como apresentado no trabalho de Salomi e Maciel (2016). Bárbara e Gabriel (2017) apresentam a automação voltada para a conferência de itens dos carros de emergência, fazendo com que o trabalho seja ainda mais produtivo, demonstrando o alto desempenho da automação no campo da saúde. Essas soluções são capazes de gerar um ganho positivo e otimizam, desde a gestão hospitalar até o atendimento aos pacientes, permitindo ter um controle e qualidade sobre as situações em diversos processos da saúde através de dispositivos que sejam capazes de se conectarem entre si (NITZAN; ROSEN, 1976).

O avanço e a progressão do uso da tecnologia nos processos hospitalares tem explorado as principais áreas do setor, como o ramo do diagnóstico avançado, que possibilitou o desenvolvimento de equipamentos médicos extremamente precisos com o uso da impressora 3D que já é uma realidade para produção de modelos e próteses. O desenvolvimento auxilia, também, no monitoramento e suporte de sinais vitais que obtém resultados positivos, possibilitando o acompanhamento em tempo real.

Nesse contexto, esse trabalho de conclusão de curso tem como objetivo introduzir soluções automatizadas no ambiente hospitalar, através do uso da tecnologia moderna (GONSALVES; SISCOOTTO, 2020). Pretende-se, portanto, integrar por meio do uso do Arduino, vários dispositivos eletrônicos que possam comunicar entre si, como sensores de segurança, de presença e diversos outros.

Além disso, o trabalho inclui um sistema supervisor que permite a identificação de falhas, sugestão de melhorias e, principalmente, o monitoramento do ambiente hospitalar, possibilitando o ajuste e controle dos leitos.

## 1.1 Objetivos

### 1.1.1 *Objetivo geral*

O objetivo geral do trabalho é tornar o ambiente hospitalar mais inteligente e interconectado, sendo capaz de identificar e melhorar os gargalos gerados dentro de um hospital e

automatizar situações relevantes para o conforto, a eficiência e segurança dos pacientes, em particular, quando relacionado à proteção de dados, conhecida também como segurança cibernética. Outro aspecto importante é fazer com que o espaço hospitalar esteja unificado em uma só interface pelo sistema supervisorio, propiciando a gestão dos ambientes hospitalares.

### 1.1.2 *Objetivos específicos*

- Projetar e desenvolver um protótipo do hospital;
- Especificar os melhores dispositivos compatíveis com Arduino para automatizar o sistema;
- Definir o protocolo de comunicação que será utilizado entre os dispositivos;
- Integrar e comunicar os dispositivos com o sistema supervisorio;
- Desenvolver um sistema supervisorio capaz de monitorar o ambiente hospitalar.

## 1.2 Justificativa

Segundo estudos conduzidos pela Organização Mundial da Saúde (OMS), a presença de acompanhantes durante a internação hospitalar é de suma importância e pode contribuir para a recuperação do paciente, uma vez que é possível promover apoio emocional e a sensação de mais segurança nos pacientes. Essa influência tem resultado nos melhores desfechos clínicos e na redução do tempo de internação, conforme afirma Ana Amélia Junior (2017) durante a discussão do projeto no Senado<sup>1</sup>. Entretanto, por se tratar de diretrizes próprias, a instalação dessas políticas depende da instituição hospitalar, que não há a garantia da atenção durante esse período de hospitalização.

Dessa forma, com a integração de um sistema capaz de supervisionar e gerenciar um hospital é possível garantir mais segurança para os internados, evidenciando uma maneira de dar atenção para os hospitalizados a partir do monitoramento contínuo dos quartos, dos parâmetros vitais do paciente e é possível também saber a localização do internado através de telas integradas.

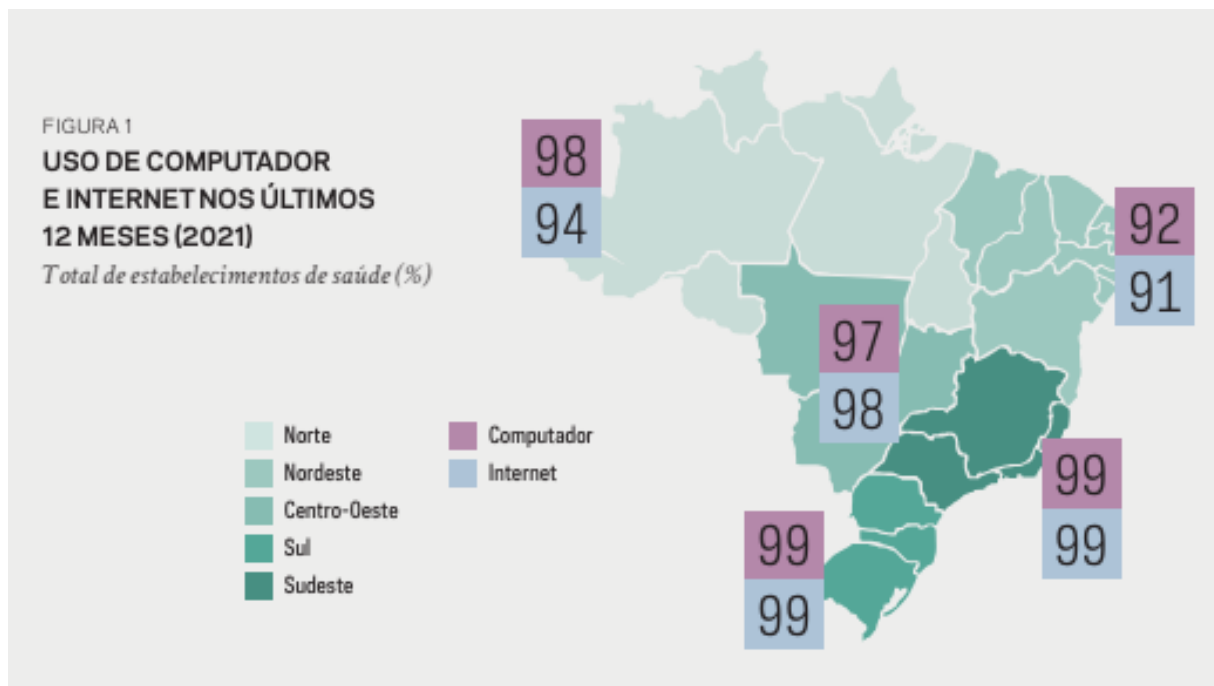
Além disso, destaca-se também que apenas determinadas categorias possuem o direito a acompanhante durante a internação<sup>2</sup>. Sem acompanhantes para auxiliar no cuidado dos pacientes, a equipe médica precisar atender um número maior de solicitações. Com a presença do supervisorio é possível automatizar tarefas que possam reduzir a carga de trabalho da equipe, priorizando cuidados mais críticos.

<sup>1</sup> Projeto de lei (PL 4996/16) que dá direito a acompanhante para todos os usuários de serviços de saúde públicos ou privados, como hospitais e clínicas, pelo tempo da internação ou atendimento. O acompanhante será pessoa de livre escolha, havendo a possibilidade de revezamento. Fonte: Agência Câmara de Notícias

<sup>2</sup> Gestantes, idosos, crianças e adolescentes, pessoas com deficiência, todos regidos por uma Lei específica para cada um

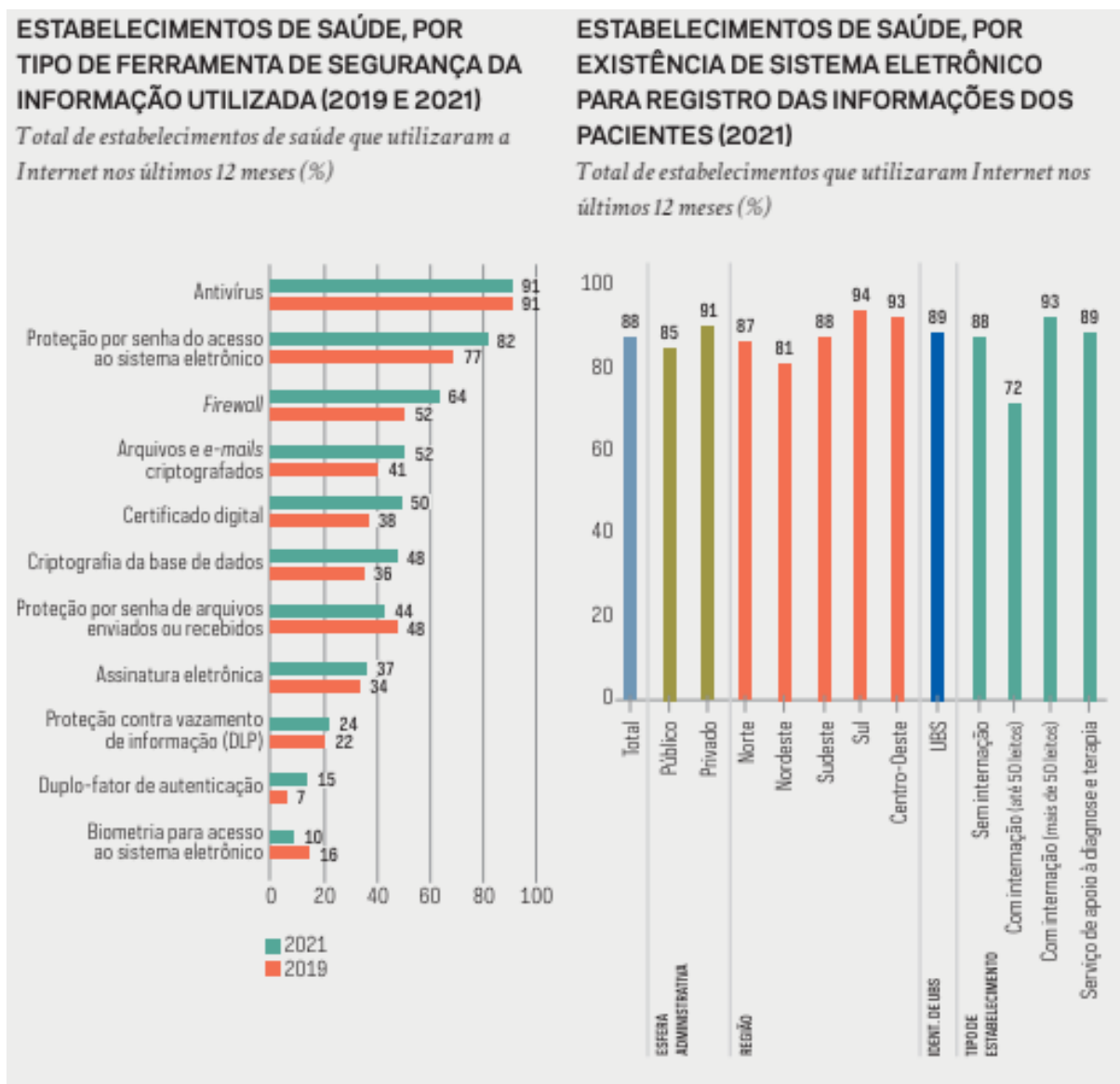
Um estudo concluído da Pesquisa TIC Saúde 2021, do Comitê Gestor da Internet no Brasil (CGI.br), conduzida pelo Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br) do Núcleo de Informação e Coordenação do Ponto BR (NIC.br) revelou que os centros de saúde no Brasil estão cada vez mais digitalizados, embora, a adoção de medidas protetoras para reforçar a segurança de dados pessoais que são armazenados ainda seja um obstáculo e tanto para o setor. Vale ressaltar que o mesmo estudo levantou dados que indicam que 94% dos estabelecimentos possuem computador e Internet, o que equivale a 9% a mais em relação a 2019. No caso de Unidades Básicas de Saúde (UBS), observa-se um crescimento de 3% do uso de computadores e 10% do uso da Internet quando comparado os anos de 2019 e 2021, destacado nas Figura 1 e 2 abaixo. Isso mostra que além da digitalização estar mais presente, ela permite que processos de acompanhamento ao paciente possam ser implementados para aprimorar o sistema de monitoramento do internado através dessa transição digital.

Figura 1 – Uso de computador e internet pelos estabelecimentos de saúde em 2021.



Fonte: [https://cetic.br/media/docs/publicacoes/2/20211124124231/resumo\\_executivo\\_tic\\_saude\\_2021.pdf](https://cetic.br/media/docs/publicacoes/2/20211124124231/resumo_executivo_tic_saude_2021.pdf).

Figura 2 – Total de estabelecimentos de saúde que utilizaram Internet em 2021.



Fonte: [https://cetic.br/media/docs/publicacoes/2/20211124124231/resumo\\_executivo\\_tic\\_saude\\_2021.pdf](https://cetic.br/media/docs/publicacoes/2/20211124124231/resumo_executivo_tic_saude_2021.pdf).

Em uma quantidade aproximada de 40.600 UBS, cerca de 38.100 possuem equipamentos necessários, enquanto 37.200 possuem conexão com a rede, o que corresponde à aproximadamente 93,84% e 91,62%, respectivamente, conforme pesquisa realizada pela Medicina SA, em 2021. Isso evidencia como os sistemas de saúde, mesmo os mais básicos, podem utilizar tecnologias suficientes para que o gerenciamento das unidades deixe de ser apenas uma ideia e se torne, de fato, algo presente e capaz de ser aplicado à essas unidades.

De forma geral, um sistema supervisor fornece monitoramento contínuo, automação, e controle centralizado, o que contribui para uma operação mais segura, econômica e eficiente, desde a entrada dos pacientes em um leito até mesmo o controle de estoque dos remédios e a configuração de alarmes automáticos para alertar condições adversas presentes no ambiente hospitalar, proporcionando um gerenciamento mais equilibrado e funcional.

### **1.3 Organização do Texto**

Este trabalho está organizado da seguinte forma: No primeiro capítulo há uma introdução sobre o tema do projeto. No capítulo dois, encontra-se a revisão bibliográfica, enquanto no terceiro capítulo, há a explicação dos conceitos e dispositivos relevantes que serão usados no projeto. Já no capítulo quatro, está descrita a metodologia usada para desenvolver o projeto, a fim de chegar em um resultado factível. No capítulo cinco, apresenta-se o desenvolvimento e resultados do trabalho, evidenciando como o projeto foi elaborado e produzido e no último capítulo, há a conclusão do trabalho, destacando quais os aprendizados adquiridos durante a fase de desenvolvimento do projeto e também a satisfação por parte do criador.

## 2 REVISÃO BIBLIOGRÁFICA

A automação e inovação em hospitais tem possibilitado grandes resultados com o alto avanço da tecnologia. Leite *et al.* (2015) cita a produção de um sistema capaz de monitorar em tempo real os níveis de glicose dos pacientes, usado inclusive em Unidades de Terapia Intensiva (UTI), enaltecendo os excelentes resultados obtidos no processo. O mesmo trabalho também mostra a alta aplicação da tecnologia por rádio-frequência (RFID) no ambiente hospitalar. Existente desde a época da Segunda Guerra Mundial, a implementação do RFID permite automatizar objetos por meio do reconhecimento de *Tags* configuradas e associadas a esses objetos, evidenciando o grande impacto que essa tecnologia tem sobre o sistema hospitalar, principalmente, na logística de remédios. O RFID usa de ondas de rádio frequência para identificar, rastrear, gerenciar e compartilhar informações sobre objetos e pessoas. O sistema conta com três principais componentes, sendo eles: Tags RFID, leitores RFID e um software para processar os dados. Essa tecnologia possibilita um ganho significativo em qualquer área. Nesse caso, como citado no trabalho de Balbi e Ribeiro (2018), com o uso do RFID, poderia ter sido evitado o desvio de remédios de elevado custo em hospitais de São Paulo que ocasionou um prejuízo significativo.

Ainda em Balbi e Ribeiro (2018), cita-se a eficiência operacional no hospital *Northern Westchester Hospital (NWH)* implementando essa solução para localizar os equipamentos médicos dentro do estabelecimento, enquanto (SWEDBERG, 2018) também apresenta o uso desse sistema, mas para controlar o estoque nas farmácias e facilitar o acesso aos medicamentos para os funcionários, sem que haja demora durante o processo, todos utilizando o sistema RFID.

Os processos de automação tem sido fundamentais na melhoria no atendimento dos pacientes na área da saúde. Modernamente, a substituição do papel para o formato eletrônico do Prontuário Eletrônico do Paciente (PEP) transformou a gestão médica por conta do rápido acesso às informações dos pacientes e por ser mais seguro, incluindo histórico e resultados de procedimentos aplicados aos próprios pacientes (PRONTPED, 2022).

Correia *et al.* (2021) mostra uma proposta de monitoramento de pontos críticos em hospitais através de um sistema supervisorio e a implementação da Internet das Coisas (IoT) com o intuito de garantir maior eficiência e segurança aos hospitalizados.

Ainda no cenário da tecnologia, houve a expansão da telemedicina<sup>1</sup>, que em épocas passadas, não se pensava na possibilidade de progressão e impacto que poderia causar no futuro. Atualmente, além de ser possível, a telemedicina permite que cirurgiões realizem procedimentos com mais precisão e que os pacientes possuam uma recuperação mais eficaz e com menos transtornos pós-operatórios (TELEMEDICINA, 2024).

<sup>1</sup> A telemedicina foi impulsionada durante a pandemia da COVID-19, pois havia a necessidade de manter o distanciamento social e reduzir a pressão gerada sobre os sistemas de saúde. Com as sobrecargas dos hospitais e a população confinada, a telemedicina se tornou uma solução eficiente para garantir o máximo de cuidados médicos possíveis. Fonte: <https://bvsm.s.saude.gov.br/telemedicina-chegou-com-a-pandemia-e-veio-para-ficar-indica-estudo/>

Em Andrade (2019) há o desenvolvimento da automação de leitos hospitalares focado ao paciente por meio do uso de aplicativos móveis e o código QR com um monitoramento feito a partir da utilização de um sistema supervisorio integrado ao banco de dados, sendo possível controlar o ambiente em que o paciente está, possibilitando ainda mais o conforto, tranquilidade e estrutura dos leitos para as pessoas.

A evolução causada pelo alto dinamismo da automação e tecnologia em hospitais também tem como foco o crescimento da Inteligência Artificial (IA) e o aprendizado de máquina. O uso dessas ferramentas envolvem análises de imagens médicas até diagnósticos assistidos por computador (GAMBUS; SHAFER, 2018) e, juntamente com o aprendizado de máquina, os algoritmos podem auxiliar a controlar dados personalizados dos pacientes.

A inteligência artificial permitirá em um futuro próximo um auxílio ainda mais eficiente na saúde, porém, ainda precisa ser estudada para que possa ser usada com ética pelos profissionais, assim como afirma Han, Kim e os demais contribuintes:

“Inteligência artificial reduzirá os esforços exigidos pelos médicos para interpretar dados digitais e melhorar sua capacidade de estabelecer um diagnóstico e prognóstico. Portanto, o aspecto humanístico não analítico da medicina passará a ser mais enfatizado, porque é difícil substituí-lo por tecnologia. (...) A educação médica futura deve ser reestruturada para se alinhar a essas mudanças inexoráveis, considerando os alunos que trabalharão em sistemas de saúde digitalizados (...). Existem algumas considerações sobre o aprendizado digital com tecnologia avançada, embora tenha muitas vantagens técnicas. Devemos considerar cuidadosamente os desafios éticos e morais, porque algoritmos de aprendizado baseado em computador e inteligência artificial podem ser programados para serem tendenciosos contra certos grupos ou desviados para quaisquer interesses. Acima de tudo, uma abordagem humanística deve ser priorizada para que futuros médicos lidem com a complexidade biopsicossocial de pacientes que não é facilmente acessível às máquinas” (HAN *et al.*, 2019).

A afirmação acima, possibilita concluir que a tecnologia industrial tem avançado significativamente na área da medicina, tornando possível desenvolver diversas aplicações que sejam suficientes para aprimorar as atividades nas quais estamos submetidos diariamente, principalmente no ramo da medicina, mesmo apesar de ser um grande desafio.

## 3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão explorados conceitos e tecnologias essenciais para o desenvolvimento de sistemas de automação e controle. Primeiramente, discute-se a importância dos bancos de dados, seguido pelo protocolo de comunicação optado na construção do trabalho. A seção também explora os principais dispositivos empregados no projeto.

### 3.1 Banco de Dados

Banco de dados, criado no início dos anos 1960, é um sistema organizado de informações estruturadas e interligadas (KORTH; SILBERSCHATZ, 1994) que tem como objetivo armazenar de forma eletrônica, todas as informações passadas pelo ambiente em que está integrado (DATE, 1985). O sistema é desenvolvido para ser acessado, gerenciado e atualizado de acordo com os critérios estabelecidos pelo usuário. O intuito de sua utilização é tornar o conjunto de dados o mais seguro e eficiente possível. Nele também é possível recuperar e administrar um alto volume de informações contidas em sua base.

Um banco de dados possui em sua essência os próprios dados, que são as informações armazenadas no sistema e as tabelas, que são organizadas em linhas e colunas, onde geralmente as linhas representam itens e as colunas referenciam atributos desses itens. Ainda, possui Sistemas de Gerenciamento de Banco de Dados (SGBD) que é o software responsável por possibilitar o acesso, entre outras funções no banco de dados.

Diversas áreas utilizam o sistema de banco de dados, como educação, ciência e principalmente o setor de saúde, especialmente em hospitais.

### 3.2 Modbus

O *Modbus* é um protocolo industrial, desenvolvido em 1979 pela Modicon para proporcionar a comunicação entre sistemas de automação industrial e controladores programáveis. Inicialmente, o *Modbus* foi criado como um protocolo de nível de aplicação simples para transferir dados, mas ao longo do tempo, o *Modbus* se expandiu no mercado, se tornando um protocolo de comunicação padrão para dispositivos eletrônicos (MCCROHAN, 2011).

O protocolo *Modbus* utiliza uma arquitetura cliente/servidor (mestre/escravo), podendo enviar e solicitar dados do servidor. O processo é iniciado pelo cliente que envia um código de função especificando o tipo de operação que se deseja utilizar. A partir dessa comunicação, ele define como um controlador acessa outro dispositivo, padronizando o formato e conteúdo dos campos de mensagem.

Os dados acessíveis através do protocolo, geralmente, são armazenados em um dos quatro tipos de dados no dispositivo escravo (SIMOES; FERNANDES; SEABRA, 2016). Esses dados

podem ser de registros, registros de entrada, entradas discretas e bobinas. A Tabela 1 mostra como funciona o comportamento de cada tipo de banco de dados.

Tabela 1 – Blocos de modelos de dados *Modbus*.

Elemento	Endereços		Modo	Tamanho
Bobina (Coil)	1	9999	Leitura / Escrita (Read / Write)	1 bit
Contato (Discrete Input)	10001	19999	Somente leitura (Read only)	1 bit
Registrador (Input Register)	30001	39999	Somente leitura (Read only)	16 bits
Registrador Retentivo (Holding Register)	40001	49999	Leitura / Escrita (Read / Write)	16 bits

Fonte: <https://automacaoecartoons.com/2018/11/23/protocolo-modbus/>.

Além dos blocos de dados do protocolo *Modbus*, também existem as funções que são comandos específicos usados para realizar operações em dispositivos de automação. Cada função possui um código único que determina a ação a ser realizada, podendo ser de leitura ou escrita de dados. As funções, bem como suas definições e uso estão representadas na Tabela 2.

Tabela 2 – Funções sobre elementos do protocolo *Modbus*.

Código	Elemento	Função	Uso / Exemplo
01	Read Coil Status	Leitura de estado de bobinas	Relé / LEDs
02	Read Input Status	Leitura de estado de um contato	Botões / Sensores
03	Read Holding Registers	Leitura do valor de um registrador	Temperatura / Controladores
04	Read Input Registers	Leitura do valor de um registrador	Tensão / Corrente / Parâmetros analógicos
05	Force Single Coil	Escrita de estado de 1 bobina	Ligar / Desligar motores
06	Preset Single Register	Escrita do valor de um registrador	Setpoints / Limites / Definição de parâmetros
15	Force Multiple Coils	Escrita de estado de várias bobinas	Alteração de várias saídas digitais simultaneamente
16	Preset Multiple Registers	Escrita dos valores de vários registradores	Definição de vários parâmetros de operação em uma única transação

Fonte: <https://embarcados.com.br/protocolo-modbus/>.

Para o trabalho em questão, foi utilizado apenas o elemento de Leitura de estado de bobinas, para ler sensores digitais e o elemento cuja função é a Leitura do valor de um registrador para se comunicar com sensores analógicos, como de temperatura e o controle de velocidade do motor.

### 3.3 Dispositivos

#### 3.3.1 *Arduino*

O *Arduino* é uma plataforma de prototipagem que permite o usuário desenvolver projetos eletrônicos, especialmente relacionados ao controle. O *Arduino* tem como seu principal componente o microcontrolador, que é um processador para executar os algoritmos inseridos. Em resumo, pode-se entender o *Arduino* como o cérebro que comanda todo o sistema integrado a ele. É uma plataforma *open-source*, ou seja, é de livre acesso e gratuito (SOUZA, 2011).

Segundo Dong, Wu e Zhang (2018), plataformas abertas tem ganhado cada vez mais espaço nos últimos anos, mostrando que as pessoas tem se interessado mais por esse tipo de

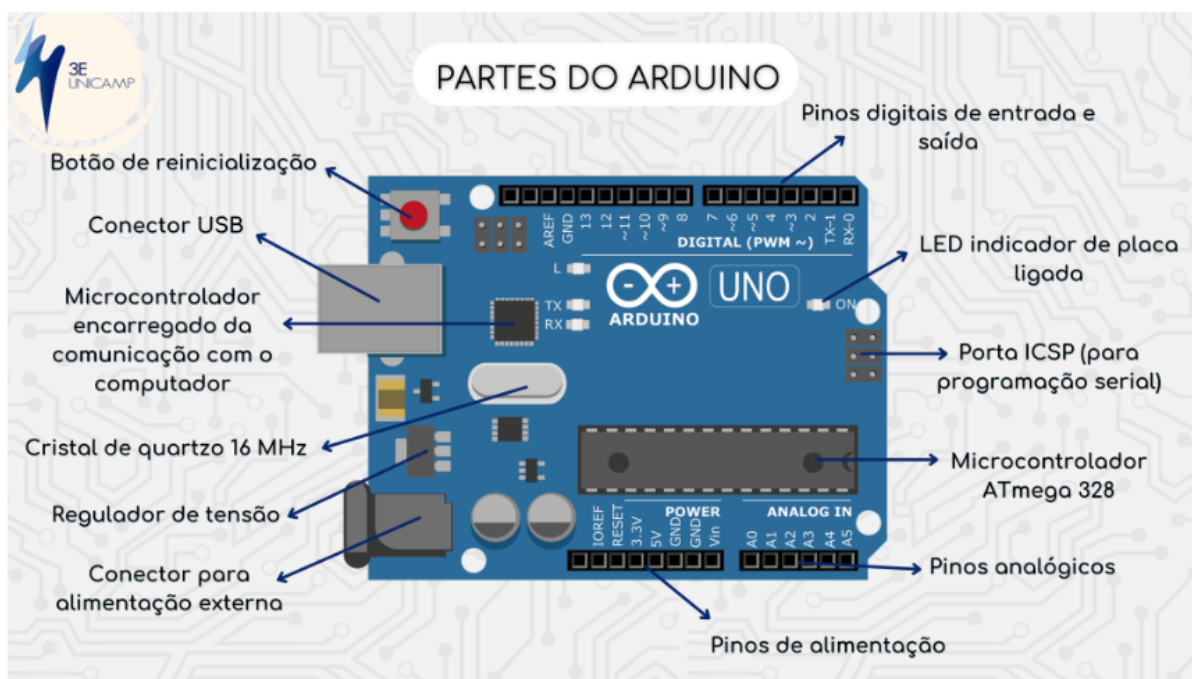
inovação e *software*. O que torna o Arduino ainda mais especial no uso de vários projetos é a facilidade de utilização, uma vez que as pessoas que não são da área técnica, conseguem aprender de forma rápida e prática pela sua alta flexibilidade sem a necessidade de um estudo complexo e amplo (MCROBERTS, 2011).

O Arduino possui portas digitais que podem operar como entradas ou saídas de dados binários (0 ou 1, correspondendo a 0V ou 5V). Elas são identificadas na placa por números e permitem o controle de dispositivos digitais, como LEDs, botões e relés. Algumas portas digitais também possuem a função PWM (Pulse Width Modulation), indicada por um símbolo (~). Geralmente, essas saídas permitem o controle de algum sensor, por exemplo, controlar a velocidade de motores ou o brilho de LEDs.

Ele também possui as entradas analógicas. Essas permitem a leitura de sinais analógicos (variáveis) e são identificadas por letras, como A0, A1, etc. Elas convertem os valores, por exemplo, a tensão de um sensor em um valor digital e faz a leitura mais precisa.

Além disso, o microcontrolador também possui portas de comunicação específicas. A porta Serial (UART), que possibilita a comunicação com computadores e outros dispositivos, sendo usada para enviar e receber dados. A porta I2C, que permite a comunicação entre o Arduino e outros dispositivos compatíveis com I2C, como sensores e displays e por fim a porta SPI, que basicamente é um protocolo de comunicação rápida usado para conectar o Arduino com periféricos como cartões SD (RFID) e módulos de comunicação. Na Figura 3 há uma representação da arquitetura de uma placa Arduino.

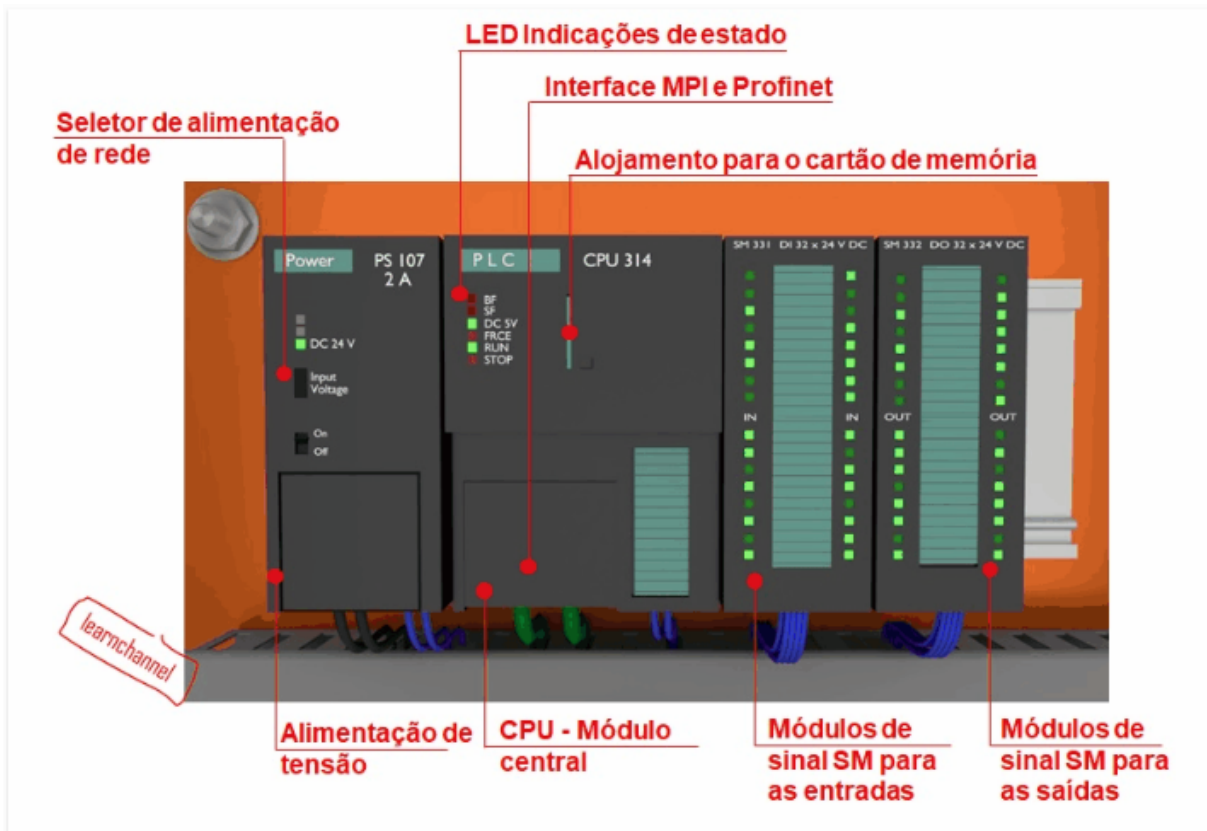
Figura 3 – Arquitetura da placa Arduino.



### 3.3.2 Controladores Lógicos Programáveis

Desenvolvido em 1960 pela General Motors (GM), os Controladores Lógicos Programáveis (CLP) tinham como objetivo substituir o uso de relés que eram muito usados para um sistema de controle na indústria. Na Figura 4 há a estrutura de um CLP. Vale ressaltar que cada controlador possui uma maneira diferente de conexão, porém, a forma de comunicação é praticamente a mesma para todos, independentemente de qual seja sua marca, mudando apenas alguns detalhes específicos, como por exemplo, a programação, regida pela Norma PLCopen (2003).

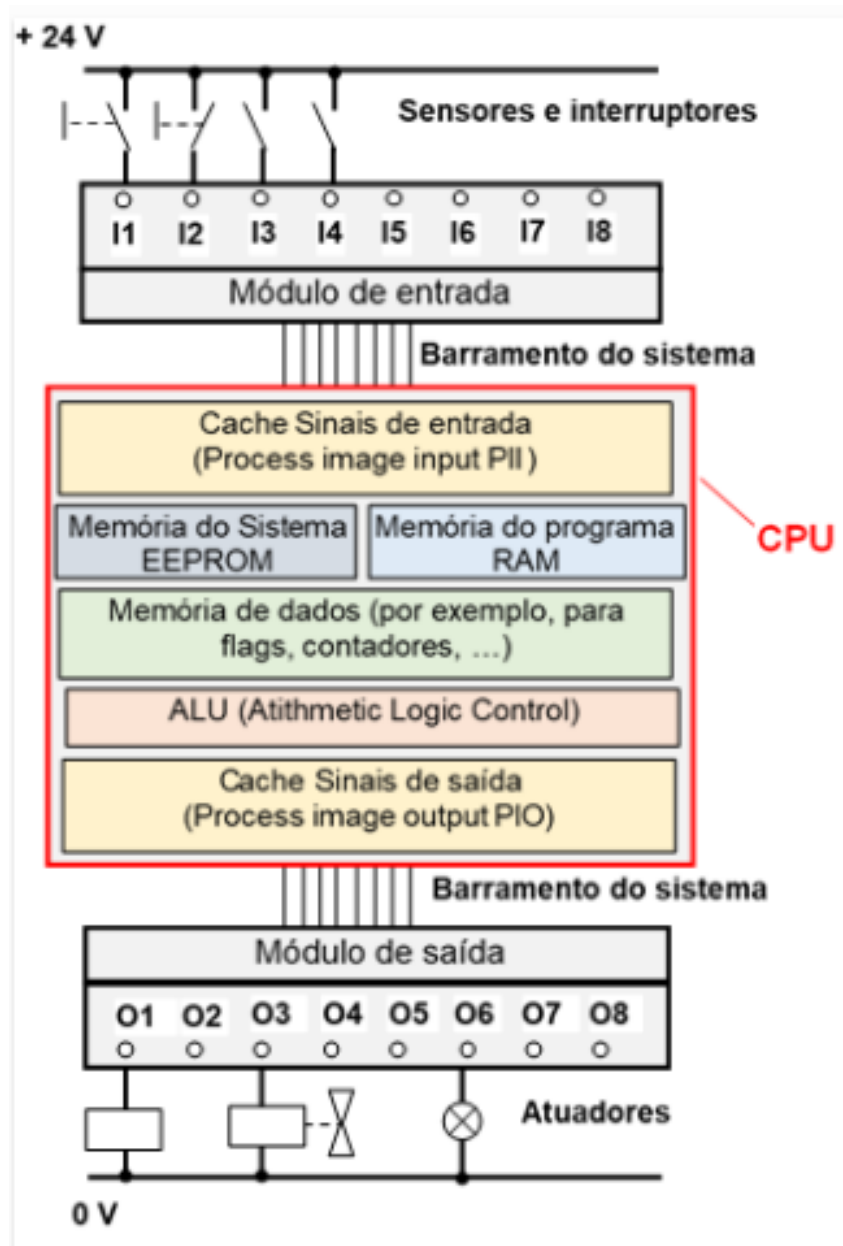
Figura 4 – Estrutura de *Hardware* do CLP.



Fonte: <https://learnchannel-tv.com/pt/clp/o-que-e-um-clp/>.

Parecido com o Arduino, o CLP também se comporta como se fosse um cérebro eletrônico, usado para controlar e automatizar vários processos de produção. A partir de entradas fornecidas, os Controladores Lógicos Programáveis são capazes de tomar decisões baseadas nas informações que recebe por meio de sensores e da programação feita pelo usuário. Sua arquitetura é composta por um processador, uma unidade de memória, fonte de energia, módulos de entrada e saída (E/S) e um dispositivo de programação. Sua estrutura de comunicação com os sensores, relacionados as entradas e saídas do CLP estão representadas na Figura 5.

Figura 5 – Estrutura CLP.



Fonte: <https://learnchannel-tv.com/pt/clp/o-que-e-um-clp/>.

O CLP possui em sua arquitetura, entradas digitais (ou Discretas), entradas analógicas e saídas digitais e analógicas. As entradas digitais suas saídas digitais recebem sinais binários (0 ou 1) de dispositivos como botões, sensores de presença, chaves fim de curso, entre outros. Quando o sensor ou dispositivo está "ligado"(HIGH), a entrada recebe um sinal, indicando uma condição como presença, ativação, ou fim de curso. O funcionamento é semelhante as entradas digitais do Arduino.

Já as entradas e saídas analógicas recebem sinais variáveis e contínuos, normalmente em uma faixa de tensão de 0 a 10 V ou 4 a 20 mA. São usadas para sensores que monitoram variáveis físicas com valores que variam continuamente, como temperatura, pressão, nível ou

velocidade. O CLP converte o sinal analógico em um valor digital, através de um conversor A/D que representa o estado da variável medida.

Com a alta evolução das empresas, o controle começou a ser mais valorizado, sendo visto como a opção mais eficiente e rápida ao longo dos anos. Com isso, o CLP se destacou no mercado industrial, ampliando seu uso até nas áreas da saúde (ÅSTROM; HAGGLUND, 2001). A Tabela 3 apresenta as vantagens e desvantagens entre o uso do CLP e Arduino.

Tabela 3 – Comparação entre Arduino e CLP.

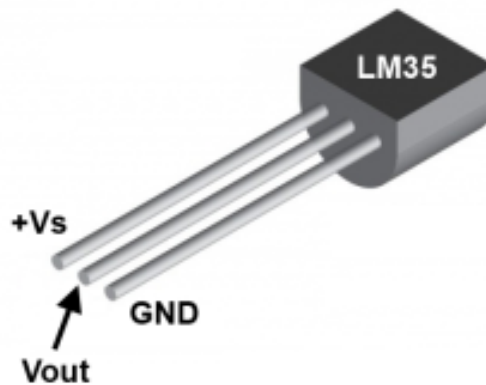
<b>Critério</b>	<b>Arduino</b>	<b>CLP</b>
<b>Custo</b>	Mais barato	Mais caro
<b>Linguagem de Programação</b>	Fácil programação para iniciantes	Requer aprendizado de software específico
<b>Robustez</b>	Menos resistente	Projetado para ambientes industriais (robusto)
<b>Entrada/Saída (I/O)</b>	Limitação de entradas e saídas	Expansível
<b>Comunicação</b>	Integração limitada (protocolos limitados)	Suporta uma variedade de protocolos industriais
<b>Confiabilidade</b>	Pode ser menos confiável em aplicações industriais	Alta confiabilidade para processos industriais
<b>Atualização de Software</b>	Fácil de atualizar e modificar o código	Difícil de modificar e geralmente mais caro para atualizar
<b>Energia e Consumo</b>	Consumo geralmente baixo	Consumo maior

Fonte: Elaborado pelo autor.

### 3.3.3 Sensor de temperatura LM35

O LM35 é um sensor de temperatura extremamente usado em projetos que envolvem partes eletrônicas. Para o projeto, o sensor será utilizado para a medição da temperatura nos quartos do protótipo. Ele possui alta precisão e sensibilidade. Além disso, o LM35 é um sensor analógico, ou seja, ele é capaz de fornecer uma tensão proporcional à temperatura medida (GEDDES, 2016). Sua faixa de medição está entre  $-55^{\circ}\text{C}$  e  $150^{\circ}\text{C}$  e pode ser alimentado com uma tensão entre 4V e 30V. Possui três terminais, sendo eles, para alimentação (VCC), um para o terra (GND) e um para a saída analógica (VOUT). A Figura 6 mostra os terminais do sensor LM35.

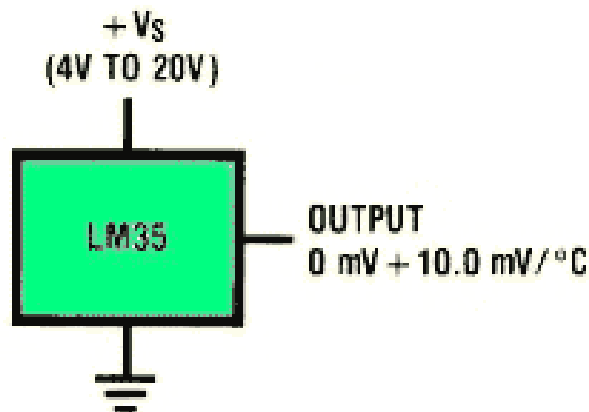
Figura 6 – Terminais do sensor LM35.



Fonte: <https://portal.vidadesilicio.com.br/lm35-medindo-temperatura-com-arduino/>.

O diagrama do circuito de ligação do sensor está representado na Figura 7

Figura 7 – Diagrama do circuito de conexão do sensor LM35.



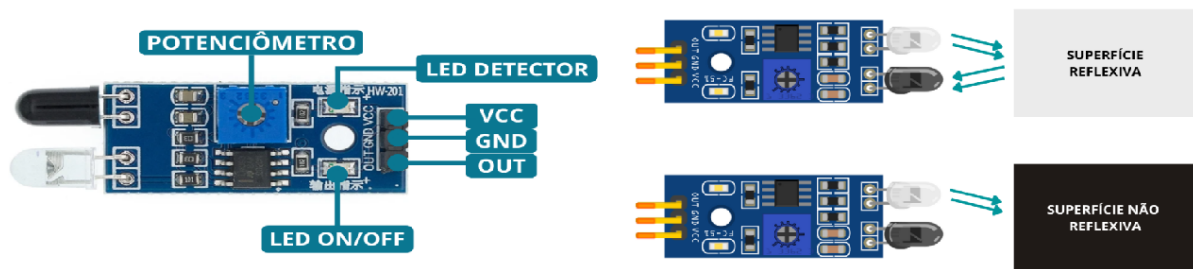
Fonte: <https://blog.novaeletronica.com.br/lm35-o-sensor-de-temperatura-mais-popular/>.

### 3.3.4 Sensor Infravermelho Reflexivo de Obstáculo

O sensor infravermelho reflexivo de obstáculo (Figura 8) é um dispositivo usado para detecção de objetos ou obstáculos em vários projetos. No projeto desenvolvido, esse sensor terá o intuito de fazer o controle de saída e entrada de pessoas dos quartos do hospital.

Esse tipo de sensor é composto por um emissor de luz infravermelho (LED infravermelho) e um receptor de luz (fototransistor), também infravermelho. Seu funcionamento se dá a partir de um feixe de luz que é emitido em uma certa direção. Quando atinge algum obstáculo, a luz é refletida de volta ao sensor, permitindo a identificação ou ausência de um objeto. O fototransistor varia sua resistência elétrica de acordo com a sensibilidade da luz que é refletida.

Figura 8 – Sensor Infravermelho Reflexivo de Obstáculo.



Fonte: <https://www.masterwalkershop.com.br/sensor-infravermelho-reflexivo-de-obstaculo>.

Sua tensão de operação pode variar de 3,3V a 5V. A distância de detecção do sensor é de 2cm até 30cm e possui saída digital e analógica.

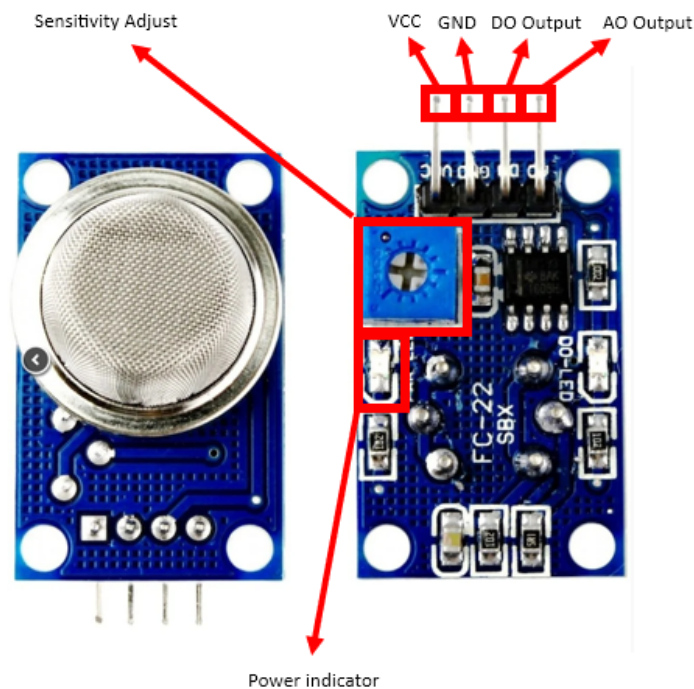
### 3.3.5 Sensor de gás MQ-2

O sensor MQ-2 é um sensor que permite a detecção de gases inflamáveis e fumaça em um determinado ambiente. A forma de detecção desses gases se dá por meio do contato do gás com uma camada de óxido metálico que é aquecida de forma interna. Ao entrar em contato, a resistência elétrica do sensor varia, fazendo com que o gás seja identificado (GEDDES, 2016). A medida que a concentração de gás aumenta, o valor de tensão na saída também aumenta.

O sensor MQ-2 (Figura 9) é bastante utilizado em aplicações que envolvem alarmes de incêndio, vazamento de gases, entre outras questões. O sensor possui quatro terminais, sendo um para alimentação (VCC), o outro para o aterramento (GND), uma saída analógica (AOUT) e uma saída digital (DOUT).

No hospital esse tipo de sensor pode ser interessante para identificar gases presentes dentro de um estabelecimento. No projeto em questão, o sensor MQ-2 estará associado a um *Buzzer* que emitirá um alarme sonoro de forma intermitente em caso de presença de gás no ambiente.

Figura 9 – Sensor de Gás MQ-2 e seus terminais de conexão.



Fonte: <https://www.smartcomponentes.com/produto/sensor-de-gas-mq-2.html>.

A tensão de operação do sensor MQ-2 é de 5V. A concentração de detecção dos gases varia de 300 a 10.000 ppm (partes por milhão) e possui um LED para identificar a presença de tensão.

### 3.3.6 Sensor de *Radio Frequency Identification (RFID)*

O Sensor RFID utiliza ondas de rádio para capturar e armazenar informações em etiquetas, famosamente conhecidas como *Tags* que são anexadas a objetos. O RFID possui três principais componentes:

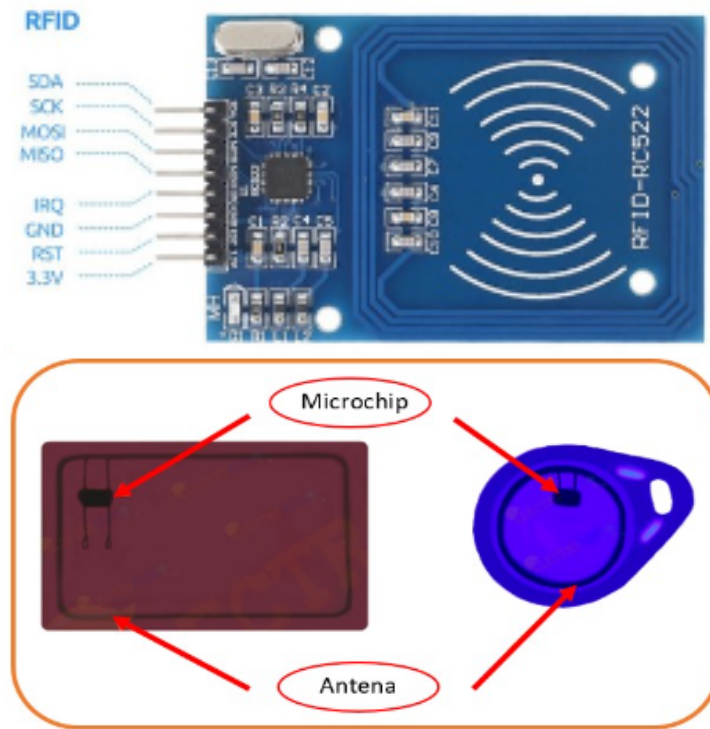
- *Tag* RFID: Essa etiqueta possui um microchip que armazena dados e uma antena que permite a comunicação com o leitor.
- Leitor RFID: Emite ondas de rádio para detectar e ler as informações armazenadas na *Tag*.
- Antena: Permite a troca de dados entre a *Tag* e o leitor de forma prática.

O RFID é amplamente utilizado em diversas aplicações, como controle de acesso, rastreamento de objetos, identificação de usuários, entre outros. No caso da aplicação do projeto, o sensor será utilizado para realizar o controle de entrada e saída dos pacientes dentro de cada

quarto através de um sistema de "crachá". O mesmo sensor será aplicado para fazer a gestão de estoque dos fármacos que são recebidos no hospital.

A Figura 10 mostra os pinos de conexão do sensor e onde ficam instalados os chips e as antenas do mesmo.

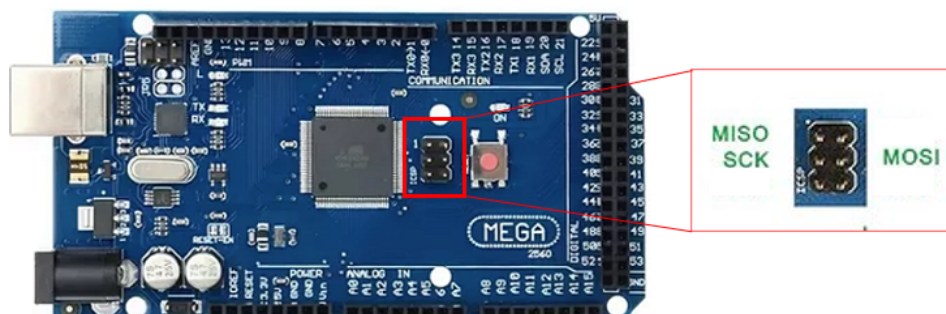
Figura 10 – Sensor RFID.



Fonte: <https://forum.arduino.cc/t/pin-connection/613444/4>.

A comunicação do sensor é feita através da comunicação SPI, que utiliza o protocolo SPI (*Serial Peripheral Interface*) para se comunicar com o microcontrolador através dos pinos ICSP do arduino. Nesses mesmos pinos é necessário conectar o SCK, que envia o sinal para sincronizar a comunicação e os pinos MOSI/MISO, que permitem a troca de dados entre o arduino e o RFID, conforme mostrado na Figura 11.

Figura 11 – Conexão dos pinos ICSP do Arduino.



Fonte: <https://forum.arduino.cc/t/pin-connection/613444/4>.



### 3.4 Internet das Coisas

Internet das Coisas (em inglês *Internet of Things*), ou simplesmente IoT, começou a ser introduzida em 1999. Em um comentário feito por Kevin Ashton em uma entrevista dada para o *RFID Journal*, Ashton disse que na época, esse termo foi utilizado pela primeira vez em um trabalho realizado por ele na *Procter & Gamble* (ASHTON, 2009).

A IoT é um resultado da evolução da comunicação de redes. O termo se refere a capacidade da conectividade de rede se comunicar com os objetos, sensores, computadores entre outros dispositivos sem a necessidade de intervenção humana (ROSE; ELDRIDGE; CHAPIN, 2015).

A Internet das Coisas é uma rede de comunicação global (FILHO, 2015). Nesse contexto, é possível realizar automação de processos no cotidiano através de celulares, dentre outros dispositivos capazes de reunir e transmitir dados (LEITE; MARTINS; URSINI, 2017).

A aplicação da IoT vai desde a indústria até a área da agricultura e saúde. Ela torna os ambientes mais seguros (dependendo do nível de aplicação ou sistema) e otimizados, melhorando inúmeros processos. Geralmente, os objetos são conectados por meio da tecnologia sem fio, como por exemplo, a *Wi-Fi*, o *Bluetooth* e outras redes. As informações e dados são enviados através da nuvem e podem ser armazenados, analisados e processados, fazendo com que qualquer pessoa que tenha acesso consiga gerenciar os dispositivos de qualquer lugar.

A Internet das Coisas tem crescido exponencialmente no mercado e tem mostrado cada vez mais força. Diversas máquinas e dispositivos estão sendo construídos com sensores que conseguem gerar informações para que os mesmos possam receber instruções para obter resultados. No setor da saúde, por exemplo, pessoas são equipadas com sensores para ter um acompanhamento do seu estado de saúde (MANYIKA; AL., 2013).

### 3.5 Heurísticas de Usabilidade de Nielsen

Com a transformação digital e a popularização das máquinas, tornou-se necessário facilitar a interação do ser humano com esse sistema cada vez mais evoluído. Sendo assim, em 1990 os cientistas Jakob Nielsen e Rolf Molich propuseram as Heurísticas de Nielsen.

As Heurísticas de Usabilidade de Nielsen são um conjunto de 10 princípios gerais para o *design* de interfaces gráficas, que são aplicáveis inclusive em telas de supervisão e controle.

Esses princípios são utilizados para avaliar principalmente a experiência do usuário (UX), bem como permite identificar problemas de usabilidade em sistemas. As heurísticas tem como foco diversos aspectos, como *feedback* claro, consistência, prevenção de erros, entre outros. Pode-se concluir então que as Heurísticas de Nielsen são hipóteses que precisam ser validadas com intuito de atender as necessidades e expectativas do projeto relacionado ao *design* dessas interfaces.

As dez Heurísticas de Nielsen estão representadas na Tabela 4 com uma breve descrição relacionada a cada critério.

Tabela 4 – Heurísticas de Nielsen.

HEURÍSTICA	DESCRIÇÃO
1. Visibilidade do estado do sistema	O sistema deve informar aos usuários o que está acontecendo, com feedback adequado em tempo razoável.
2. Correspondência entre o sistema e o mundo real	O sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares, em vez de termos técnicos.
3. Controle e liberdade do usuário	Ofereça opções de desfazer e refazer ações, permitindo que o usuário se sinta no controle.
4. Consistência e padrões	Usuários não devem se perguntar se diferentes palavras, situações ou ações significam a mesma coisa. Mantenha a consistência em todas as plataformas.
5. Prevenção de erros	Melhor do que mensagens de erro é prevenir que erros ocorram desde o início.
6. Reconhecimento em vez de memorização	Minimize a carga de memória do usuário, tornando elementos, ações e opções visíveis.
7. Flexibilidade e eficiência de uso	O sistema deve permitir que usuários experientes executem ações rapidamente por meio de atalhos.
8. Estética e design minimalista	Interfaces devem conter apenas as informações essenciais, sem sobrecarregar o usuário com dados irrelevantes.
9. Ajudar usuários a reconhecer, diagnosticar e corrigir erros	As mensagens de erro devem ser expressas em linguagem simples, indicar o problema e sugerir uma solução.
10. Ajuda e documentação	Mesmo que o sistema deva ser fácil de usar sem documentação, é necessário fornecer ajuda e informações acessíveis quando necessário.

<https://rockcontent.com/br/blog/heurísticas-de-nielsen/>.

### 3.6 Sistemas Supervisórios

Sistemas supervisórios são sistemas que armazenam informações em um banco de dados. Com isso, é possível rastrear e monitorar inúmeras aplicações em um processo. Esses sistemas também são conhecidos como *Supervisory Control and Data Acquisition* (SCADA). A principal função de um sistema supervisório está na maneira em que o usuário consegue controlar e obter as informações, praticamente, tudo em tempo real (VIANNA; BRINGHENTI; MARTINS, 2008).

Pode-se destacar três principais papéis do sistema SCADA:

- Supervisão: De maneira geral, é a parte de monitoramento do processo.
- Operação: Função de enviar comandos aos dispositivos e equipamentos interligados.
- Controle: A maioria possui funções que permite a atividade automática do sistema de acordo com a programação do usuário.

Esses sistemas possibilitam uma série de vantagens para um ambiente automatizado, como a garantia de qualidade através da supervisão, a redução de custos operacionais, o maior desempenho do processo pela rapidez de leitura, a identificação de problemas que podem ser paralisados e por ser um sistema que armazena dados em um banco de dados, proporciona uma base para outros sistemas serem integrados.

## 4 METODOLOGIA

Duas metodologias foram utilizadas para o trabalho. A primeira delas foi o método do Duplo Diamante para delimitar a ideia e o escopo do projeto para posteriormente desenvolver a segunda metodologia, a Matriz GUT, para que dessa forma pudessem ser tomadas as melhores decisões sobre o que automatizar no ambiente hospitalar seguindo uma classificação específica, mostrando o processo de desenvolvimento do projeto proposto.

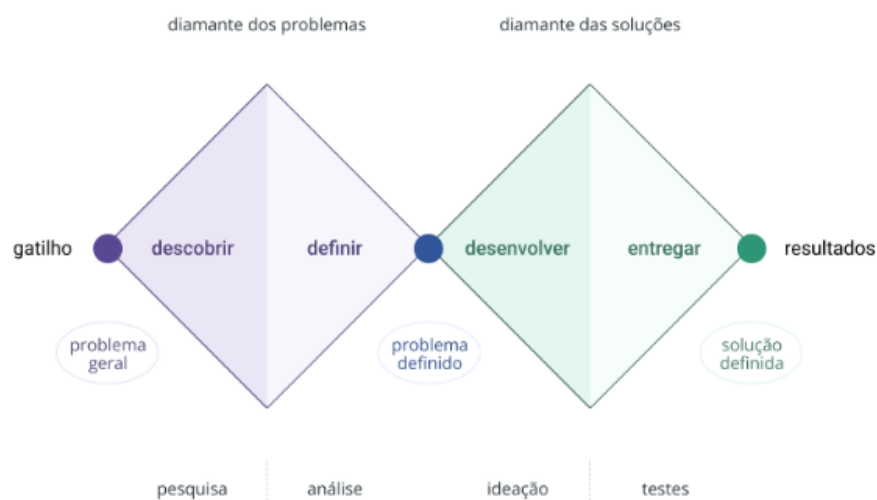
### 4.1 Método do Duplo Diamante

Duplo Diamante, também conhecido como *Double Diamond*, em inglês, é um método que foi desenvolvido pelo Conselho de *Design* do Reino Unido em 2005. O método é bastante aplicado para resoluções de problemas e desenvolvimento de soluções inovadoras. O nome de Duplo Diamante é dado por conta do formato de dois diamantes que representam algumas fases do projeto, conforme apresentado na Figura 13.

Para ajudar a pensar nas principais aplicações do trabalho que serão desenvolvidas, foi considerado usar esse tipo de método pela facilidade de trabalho e para que pudesse ser construída uma Matriz GUT, elaborada pelo autor e que está representada na Tabela 5.

O Duplo Diamante é um método simples e bastante intuitivo de lidar e possui seis partes distintas e importantes. Para o desenvolvimento do projeto, os seis passos foram descritos de acordo com o pensamento do autor.

Figura 13 – Fases do Método do Duplo Diamante.



De maneira geral, as etapas do Duplo Diamante representam um processo estruturado e iterativo para a resolução de problemas complexos, garantindo uma compreensão mais profunda das necessidades do indivíduo e a criação de soluções eficazes em relação à situação proposta.

- 1 - Gatilho/Desafio: Melhorar a logística hospitalar. De maneira geral, foi considerado produzir uma forma da gestão hospitalar ser mais eficiente, de forma que evitasse o máximo possível de gargalos.
- 2 - Descobrir: Focar na gestão de pontos específicos do hospital. Controlar, monitorar e melhorar o ambiente hospitalar automatizando diversas aplicações, baseado na construção da Matriz GUT para auxiliar nessa decisão.
- 3 - Definir: Ideia do passo 2 em definitivo. Será desenvolvido a automação para gestão do ambiente hospitalar.
- 4 - Desenvolver: Desenvolvimento da ideia definida e esquematização do escopo do projeto. O grande objetivo é melhorar a gestão hospitalar para solucionar possíveis problemas dentro do ambiente automatizado a partir de um protótipo.
- 5 - Entregar: Encaminhando para a parte final, nessa fase, é feita a prototipagem em impressora 3D, bem como a maquete montada para demonstrar a aplicação da automação do projeto final e o desenvolvimento do sistema supervisorio.
- 6 - Resultados/Solução: Nessa última parte está o lançamento da solução. Nessa fase há o questionamento se de fato tudo o que foi desenvolvido no projeto possui um resultado satisfatório em relação à automação para uma melhor gestão hospitalar.

## 4.2 Matriz GUT

A Matriz GUT é uma ferramenta de tomada de decisão baseada em três critérios específicos, sendo eles, Gravidade, relacionado ao impacto do problema, Urgência, que indica se o problema tem necessidade de ação imediata ou não e Tendência, que é a probabilidade do problema se agravar conforme o tempo passa. Essa ferramenta é amplamente usada para gerenciar projetos e solucionar problemas a partir do grau de atenção de uma determinada questão.

Os critérios são avaliados de 1 a 5, onde o número 1 representa um caso pouco grave e o grau 5 representa uma situação de maior risco, precisando de uma tomada de decisão imediata. Após a atribuição desses valores para cada critério, é necessário realizar a multiplicação dos três valores ( $G \times U \times T$ ) a fim de obter um resultado que identificará qual a prioridade será seguida de acordo com a classificação.

Tabela 5 – Desenvolvimento da Matriz GUT.

MATRIZ GUT					
PROBLEMAS	GRAVIDADE G	URGÊNCIA U	TENDÊNCIA T	RESULTADO G x U x T	CLASSIFICAÇÃO (PRIORIDADE)
Acesso aos Leitos Hospitalares	5	4	5	100	2°
Logística Farmacêutica (Remédios e Estoques)	5	4	4	80	3°
Tempo de espera dos pacientes	4	4	4	64	4°
Acompanhamento dos pacientes	3	4	5	60	5°
Controle do ambiente (Leitos) – Camas, Iluminação	4	4	5	80	3°
Sistemas de alarme e emergência do Hospital	5	5	5	125	1°
Segurança Hospitalar (Monitoramento e Rastreabilidade)	4	5	4	80	3°

Fonte: Elaborado pelo autor.

Para o trabalho atual, foi pensado em construir a Matriz GUT para delimitar quais os maiores impactos do ambiente hospitalar, destacando que os critérios analisados foram de escolha do autor, baseados em situações vivenciadas e observadas em alguns locais.

Após a avaliação dos dados obtidos ao construir a Matriz GUT, pode-se inferir quais são as principais necessidades mais relevantes para melhorar o ambiente a ser automatizado.

Da listagem feita na Tabela 5, os principais problemas solucionados pelo trabalho foram a geração de um alarme de emergência do hospital, indicado como ponto central, o acesso aos leitos através do controle de acessos com um cartão de paciente, bem como a logística farmacêutica, controlando os estoques dos remédios, o controle de ambiente do hospital, permitindo posicionar a cama e a luz dos quartos e por último, o acompanhamento dos pacientes de forma remota, pois conforme pontuado no trabalho, como algumas unidades não permitem acompanhamento, dependendo do caso e do internado, o monitoramento através do supervisor possibilita a supervisão dos pacientes por meio de telas.

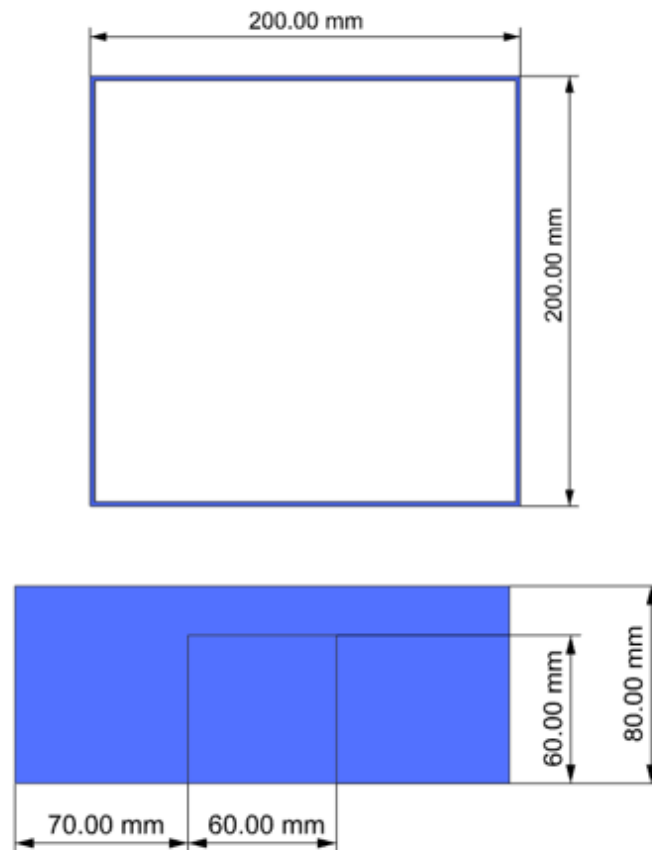
## 5 DESENVOLVIMENTO E RESULTADOS

Nesse capítulo, primeiramente será apresentado o desenvolvimento do protótipo do hospital na Seção 5.1. No item 5.2 há a construção da cama hospitalar. Na sequência, na Seção 5.3 está a construção do sistema supervisorio, desde a adição do protocolo de comunicação até o desenvolvimento das telas.

### 5.1 Maquete do hospital

Para o desenvolvimento da maquete, foi utilizado o *software* NX, da Siemens. As dimensões da estrutura de cada quarto está referenciada na Figura 14. Após a criação do desenho, o mesmo foi exportado para uma extensão em .STL para realizar a impressão do material em 3D, levando em consideração as limitações da impressora 3D do Laboratório de Impressão disponível para o uso dos discentes do IFMG.

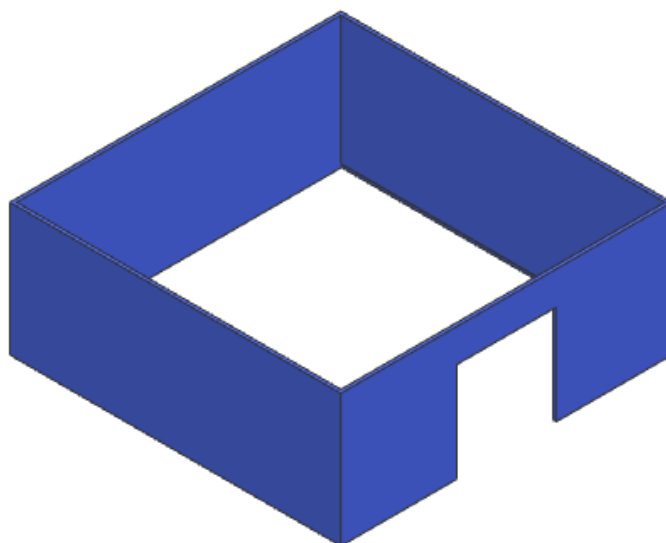
Figura 14 – Dimensões das partes do protótipo do hospital.



Fonte: Elaborado pelo autor.

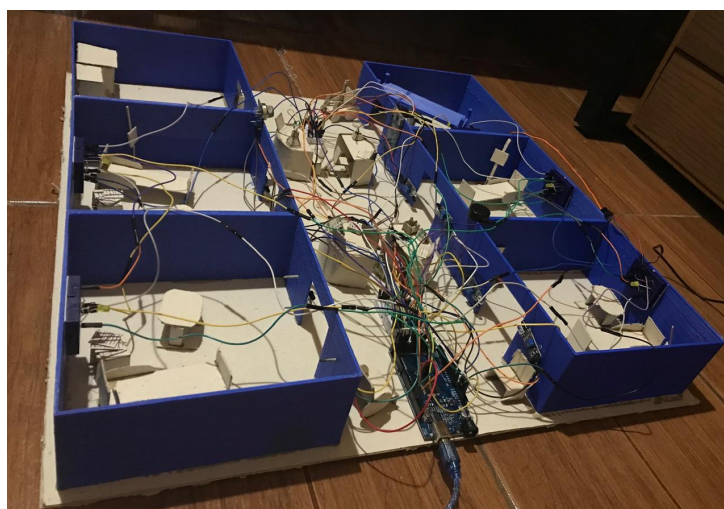
Já a Figura 15 representa a vista isométrica de uma parte do protótipo. Vale ressaltar que a maquete em sua versão final possui 6 dessas estruturas e estão apresentadas na Figura 16.

Figura 15 – Vista isométrica do quarto do hospital.



Fonte: Elaborado pelo autor.

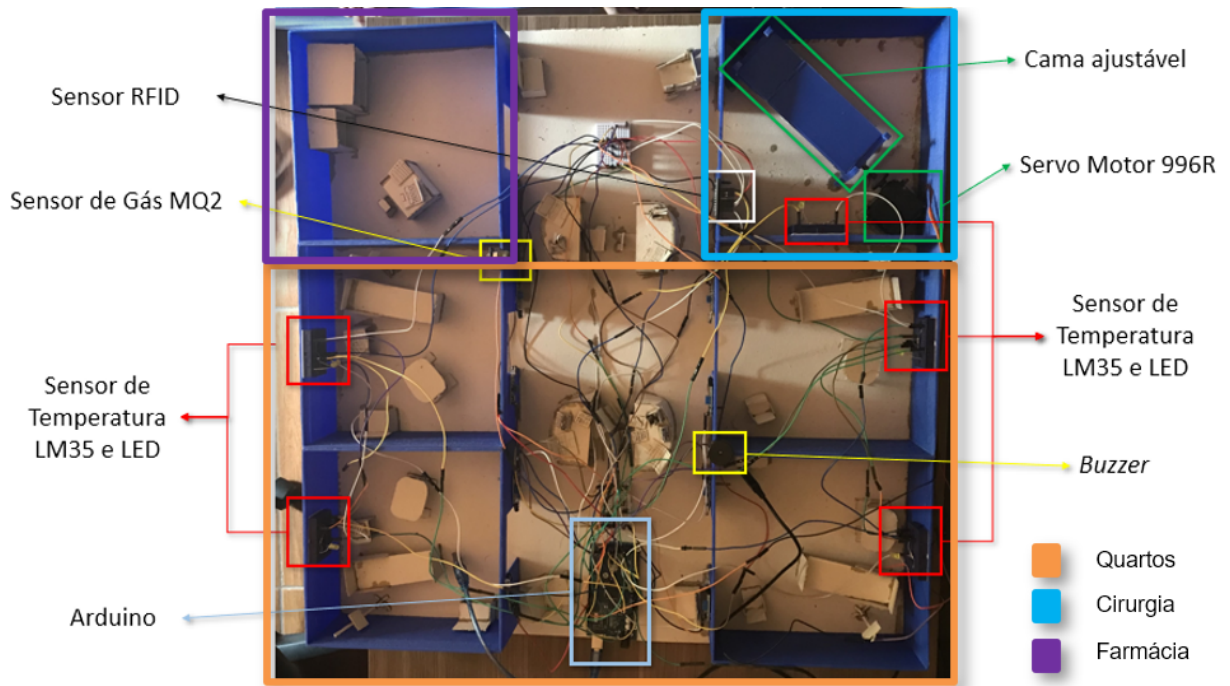
Figura 16 – Protótipo na versão final.



Fonte: Elaborado pelo autor.

A Figura 17 detalha onde estão instalados os sensores e como ficou a distribuição final do protótipo. Ele está dividido em três ambientes, sendo um ambiente denominado de quartos, sendo quatro no total, um ambiente chamado de cirurgia e o último a área da farmácia.

Figura 17 – Vista superior do protótipo.



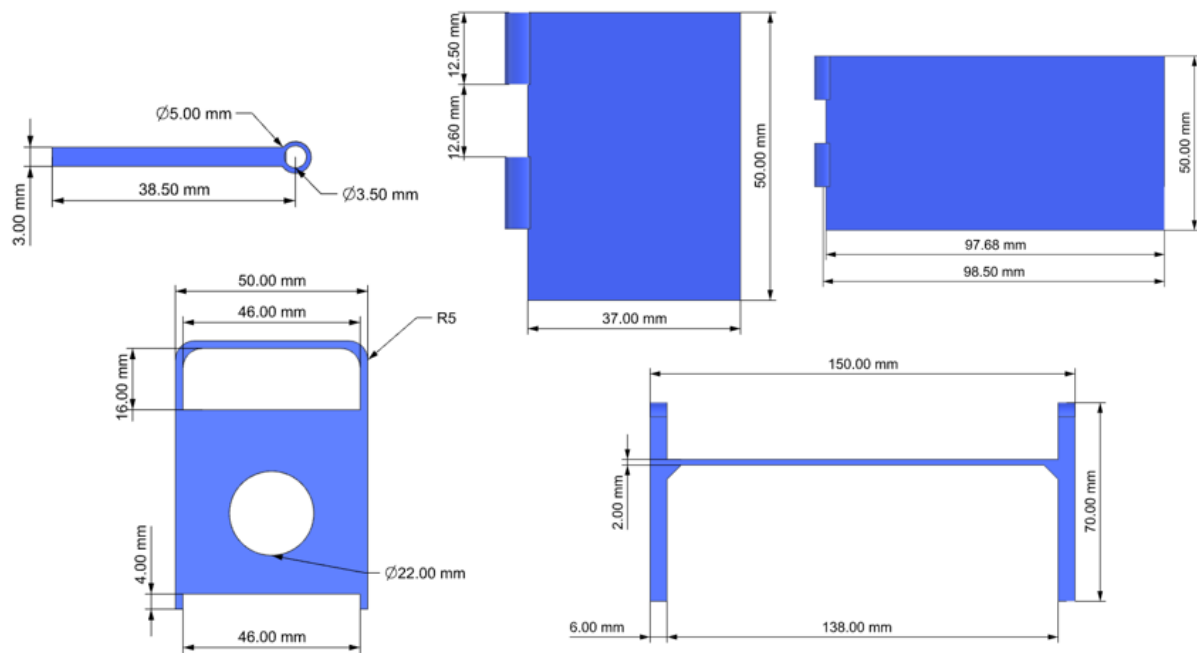
Fonte: Elaborado pelo autor.

## 5.2 Projeto da cama hospitalar

Para o desenvolvimento do modelo 3D do protótipo da cama hospitalar, assim como na montagem das partes do hospital, o *software* NX foi o escolhido.

Novamente, considerando as mesmas limitações, foram estabelecidas as medidas do protótipo que não excedessem as dimensões da impressora 3D (300mm x 300mm x 300mm). A Figura 18 evidencia as referências dos valores escolhidos para a montagem da cama hospitalar, desde a cabeceira até o apoio da cama.

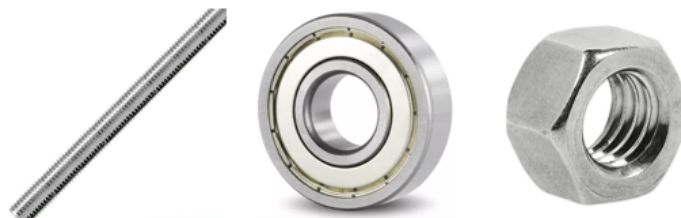
Figura 18 – Dimensões das partes do protótipo da cama.



Fonte: Elaborado pelo autor.

Após a impressão de todos os elementos necessários para a montagem do protótipo, foi definido o mecanismo de funcionamento da cama, para permitir que o usuário escolha a posição da cabeceira. Tal mecanismo de rosca sem fim funciona de maneira que ao girar o parafuso, fixado em dois rolamentos 608 ZZ/C3, a parte construída com papelão tende a rotacionar, fazendo com que a cabeceira da cama se eleve de acordo com a posição definida pelo usuário, que será indicado nas seções do desenvolvimento do sistema supervisor. A Figura 19 mostra os itens utilizados para o funcionamento do mecanismo em questão.

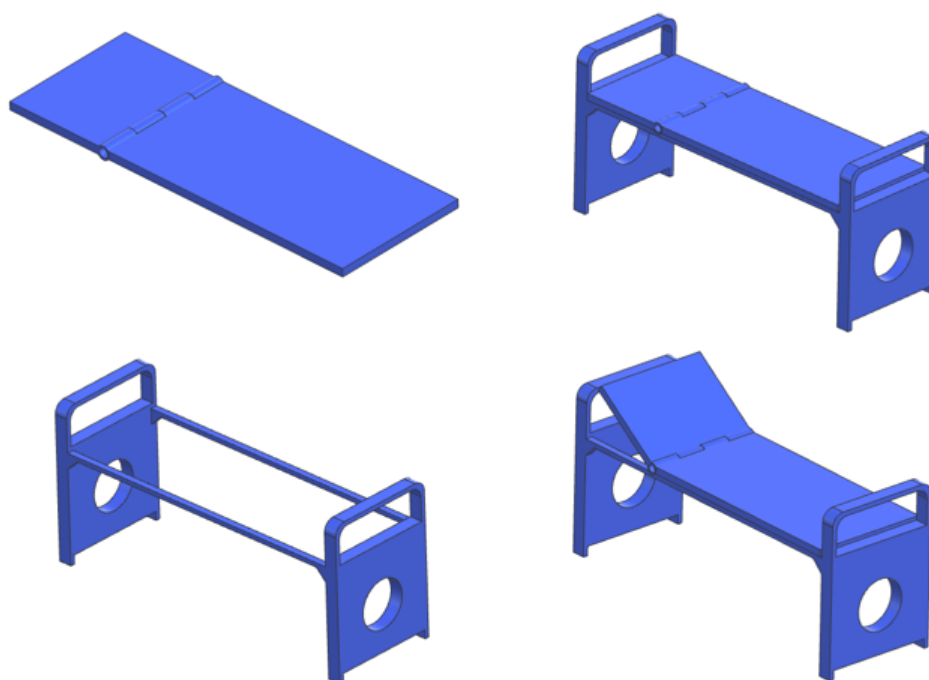
Figura 19 – Parafuso de rosca sem fim, porca e rolamento.



Fonte: <https://pt.aliexpress.com/?spm=a2g0o.productlist.logo.1.1b26UIEEUIEEvzgatewayAdapt=glo2bra>.

A construção do projeto final da cama está representado na Figura 20.

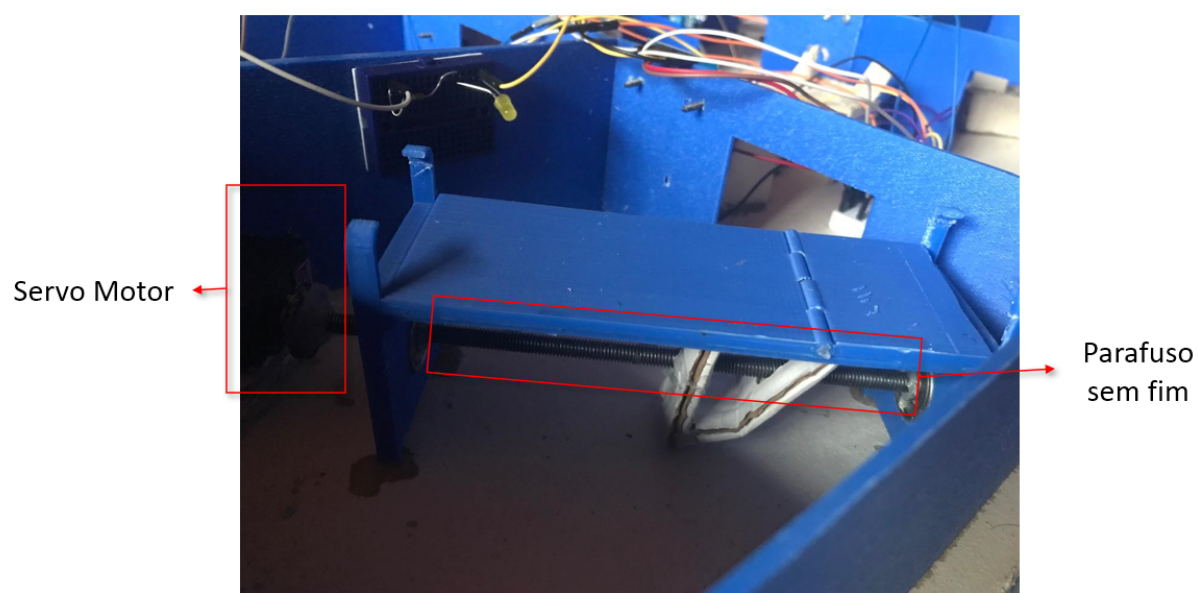
Figura 20 – Vista geral da montagem final da cama.



Fonte: Elaborado pelo autor.

A Figura 21 mostra como ficou a montagem da cama final.

Figura 21 – Protótipo da cama na versão final.



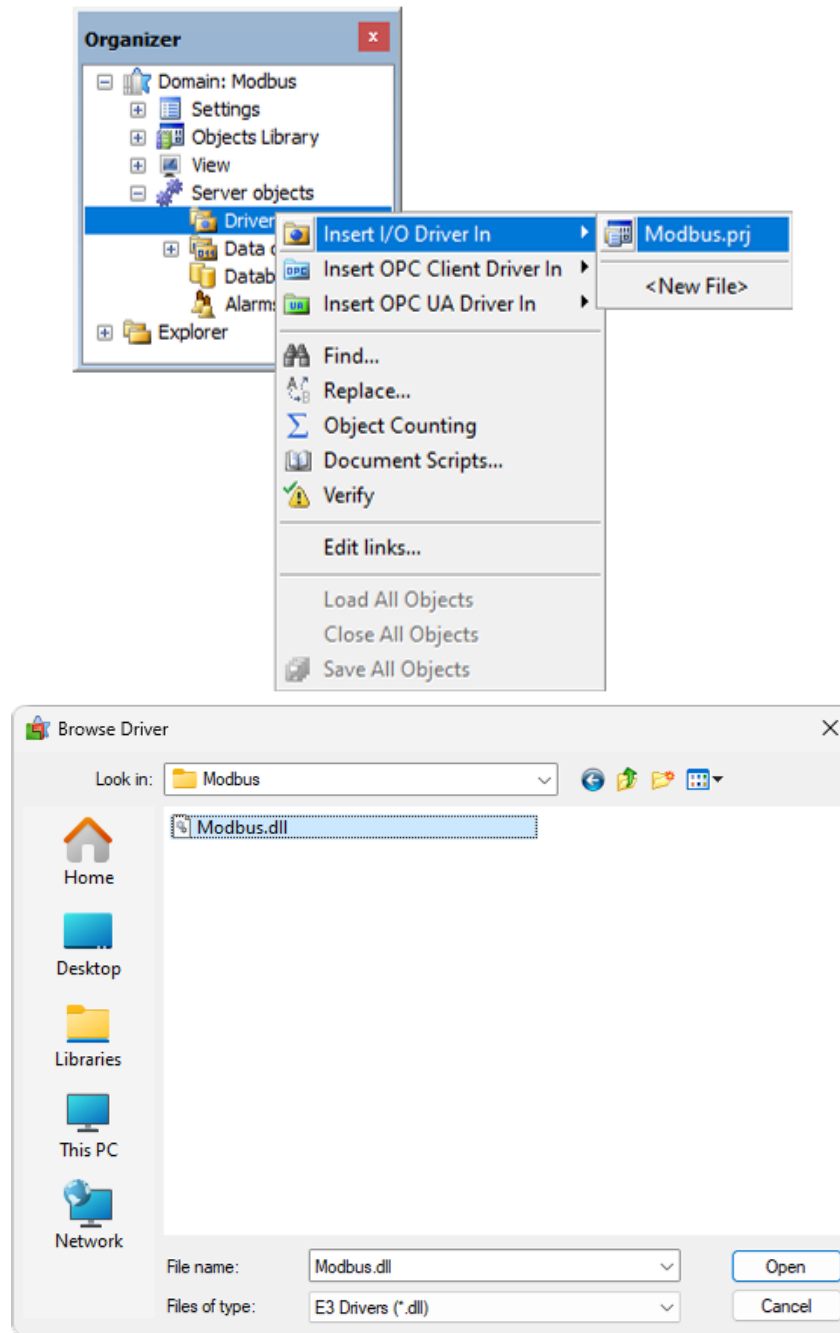
Fonte: Elaborado pelo autor.

## 5.3 Construção do Sistema Supervisório

### 5.3.1 Protocolo e *Driver* de comunicação

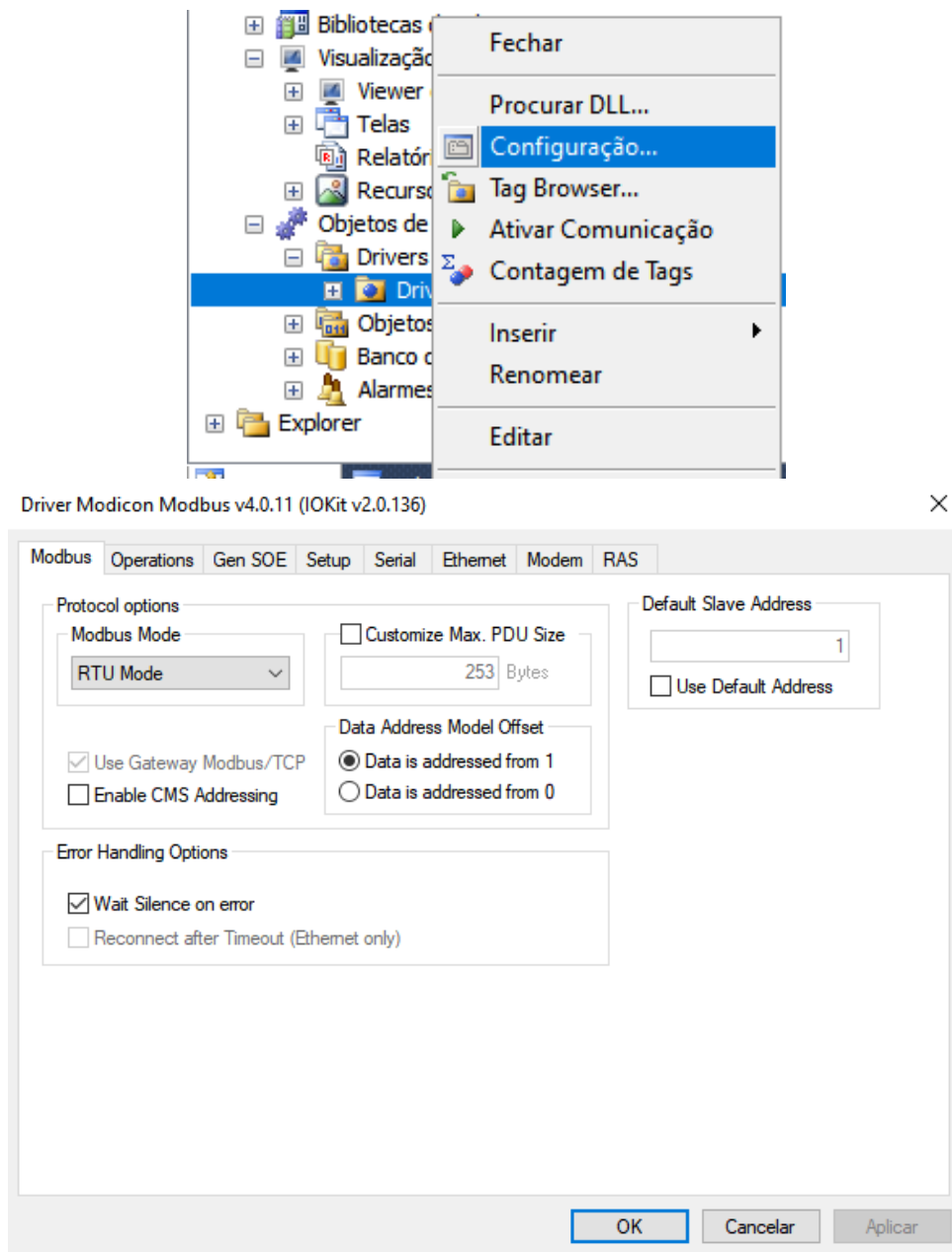
Para a construção do sistema supervisório, foi utilizada a ferramenta Elipse E3, no qual é necessário, inicialmente, configurar o protocolo de comunicação, no caso, o escolhido foi o protocolo Modbus, por ser um dos protocolos mais utilizados na indústria, possuindo compatibilidade universal, sendo bem versátil, bem eficaz em aplicações mais críticas, garantindo mais confiabilidade no sistema e também ser o mais prático e estável do mercado.

Como primeiro passo, deve-se inserir o *driver* no sistema, que será capaz de realizar a troca de informações com o sistema supervisório, conforme mostrado na Figura 22. O *driver* Modbus.dll permite a uma aplicação do software Elipse E3 comunicar com qualquer equipamento escravo nos modos ASCII, RTU ou TCP. Para o trabalho, foi determinado o uso do modo RTU. O modo RTU é vantajoso por causa da simplicidade, baixo custo, eficiência de comunicação serial, principalmente com o arduino e sua compatibilidade com o hardware, confiabilidade em ambientes industriais e facilidade de integração com dispositivos industriais. O *driver* Modbus.dll atua sempre como mestre de uma rede Modbus.

Figura 22 – Inserindo o *Driver* Modbus.dll.

Fonte: Elaborado pelo autor.

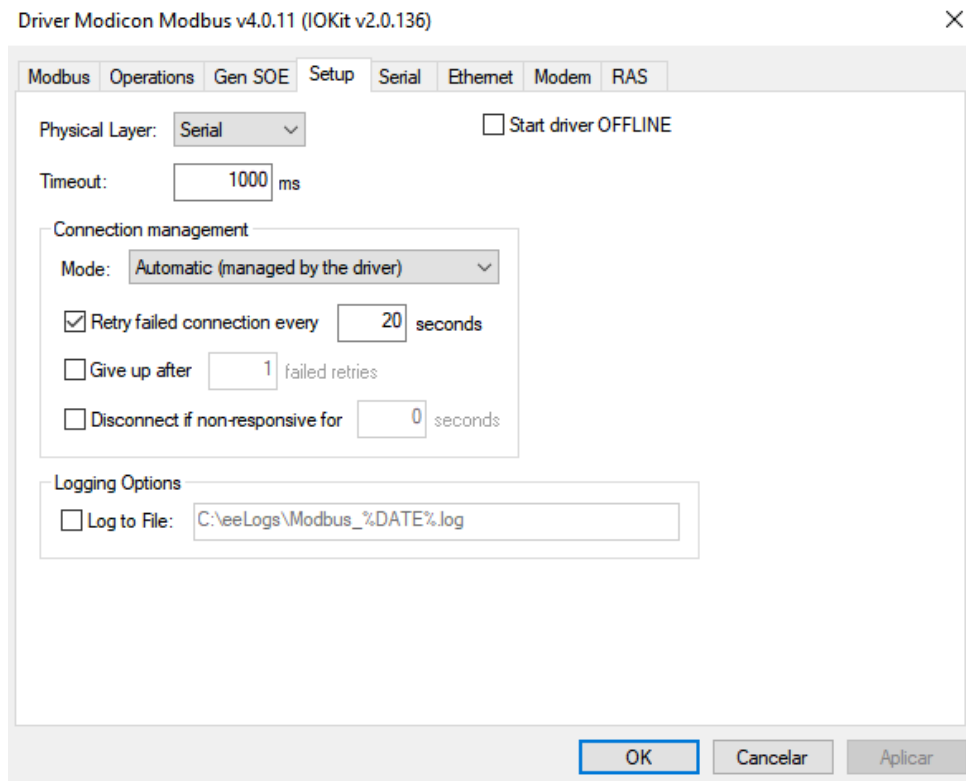
Após a inserção do *driver* é preciso configurar o protocolo de acordo com a necessidade de uso. A Figura 23 mostra como realizar essa configuração depois de selecionar a opção "Configuração" dentro do software.

Figura 23 – Configurando o *Driver Modbus.dll*.

Fonte: Elaborado pelo autor.

Uma vez com a configuração do RTU Mode estabelecido, foi definida qual a camada física seria trabalhada no projeto. A Figura 24 demonstra a opção Serial selecionada.

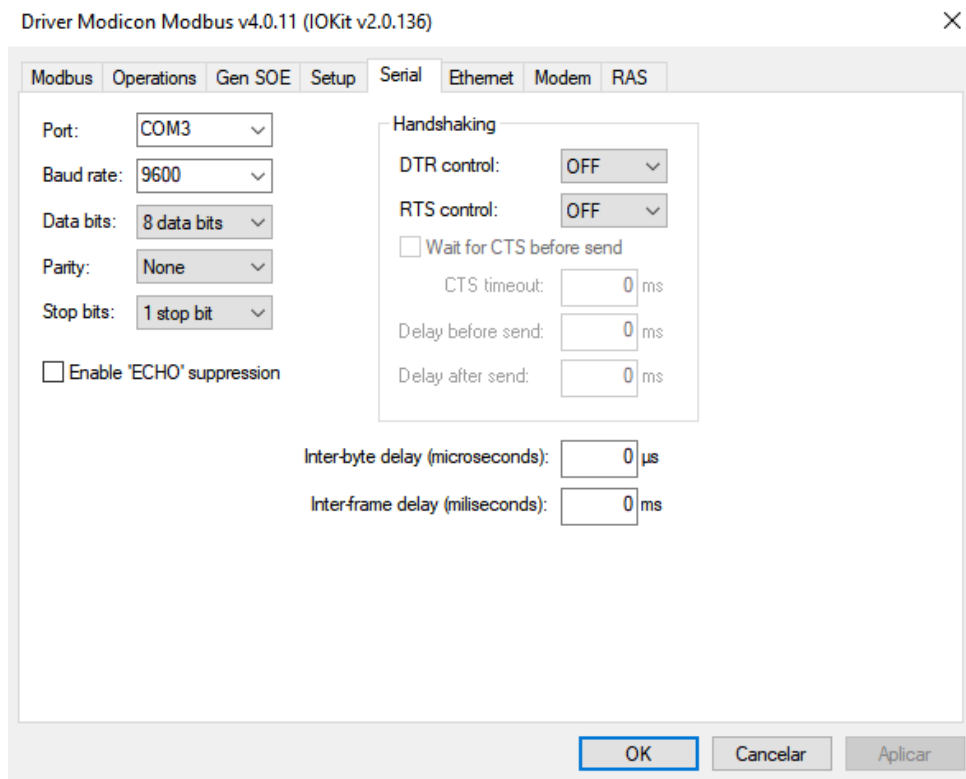
Figura 24 – Selecionando a camada física trabalhada.



Fonte: Elaborado pelo autor.

Com a camada física já determinada, estabeleceu-se qual a porta serial seria utilizada para a conexão do arduino, bem como a seleção do *Baud rate* para 9600, para que possa se comunicar com o arduino, garantindo mais confiabilidade no processamento, facilidade e compatibilidade, principalmente por ser a mais comum e bem suportadas dos dispositivos seriais. Essas especificações estão apresentadas na Figura 25.

Figura 25 – Estabelecendo a porta serial e a taxa de baud.



Fonte: Elaborado pelo autor.

### 5.3.2 Configuração das Tags de comunicação

A configuração de Tags no Elipse E3 foi realizada por meio da configuração por *strings*, no qual as variáveis que representam dados de campo, como valores de sensores ou estados de um dispositivo, por exemplo, são identificadas e configuradas usando nomes descritivos em formato de texto ao invés de utilizar números ou códigos, nesse caso, denominada configuração numérica.

Para adicionar Tags no sistema supervisorório foi considerada apenas Tags de entrada e saída (I/O), referenciada por Tag de Comunicação na tela de adição. Algumas das Tags criadas estão representadas na Figura 26.

Figura 26 – Adicionando Tags à aplicação.

Nome	Dispo...	Item	P1/N1...	P2/N2...	P3/N3...	P4/N4...	Ta...	Var...	Leitura?	Escrita?	Escala?	Min. UE	Máx. UE	UE	Min. E/S	Máx. E/S
Driver 1			0	0	0	0										
• Temperatura			1	1	0	1		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Pessoas			1	1	0	3		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Gas			1	1	0	4		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• RFID			1	1	0	5		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Pessoas2			1	1	0	6		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Pessoas3			1	1	0	7		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Pessoas4			1	1	0	8		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Motor			1	3	0	9		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• Led			1	1	0	10		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• LedQ1			1	1	0	11		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• LedQ2			1	1	0	12		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• LedQ3			1	1	0	13		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• LedQ4			1	1	0	14		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• TemperaturaQ2			1	1	0	15		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• TemperaturaQ3			1	1	0	16		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000
• TemperaturaQ4			1	1	0	17		1000	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1000		0	1000

Fonte: Elaborado pelo autor.

Vale ressaltar uma grande importância para configuração das Tags criadas. Deve-se sempre se atentar aos parâmetros que se destina a cada uma delas. Os parâmetros "P/N" permitem a configuração de endereçamento para o Tag de acordo com o *driver* selecionado. Para o projeto do sistema supervisão, o parâmetro P1/N1 se refere ao endereço do equipamento (*Slave ID*), que no caso está destinado ao endereço 1.

O parâmetro P2/N2 é referente ao código da operação. Como a maioria dos sensores utilizados possuem apenas saídas digitais que podem ser ligadas ou desligadas, nos referimos ao código 1, como leitura de bobinas (*coils*), exceto para o motor, que indica uma leitura do dispositivo, permitindo o armazenamento dos valores analógicos lidos. Essas informações de parâmetros estão apresentadas mais detalhadamente no capítulo da Fundamentação Teórica.

O atributo P3/N3 não foi alterado pois não há necessidade de uso de acordo com a aplicação do trabalho, porque são parâmetros adicionais usados em situações específicas. Ele está relacionado com funções de arquivos, *Use Bit Mask*, função especial Gen SOE, entre outros tipos de dados. Como última especificação há o P4/N4, que indica o endereço do registro Modbus ou bit em um equipamento ou dispositivo escravo. Esse atributo está relacionado com o registrador referenciado dentro do código que está presente no arduino. As demais configurações das Tags não foram modificadas, mantendo apenas o padrão já estabelecido pelo próprio software.

As Tags de Temperatura estão relacionadas ao sensor de temperatura LM35. Já as Tags cujo nome das variáveis são "Pessoas" estão associadas aos sensores infravermelhos. Vinculado ao servo motor está a Tag Motor, enquanto as Tags dos LEDs estão interligadas com os LEDs distribuídos pelo protótipo. O sensor de gás instalado na maquete está sendo conectado com a referência da Tag Gas e por fim, a Tag RFID é relacionado ao próprio sensor RFID.

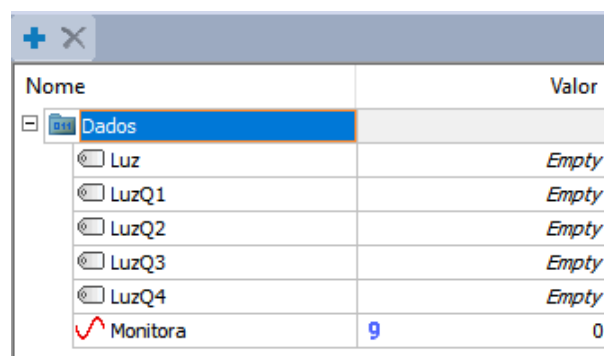
### 5.3.3 Tags de comunicação internas e tags demo

Assim como as Tags de comunicação do *driver* Modbus, existem outras configurações de tags relevantes que foram utilizadas para simulação básica no sistema supervisório. A Tag interna, que são variáveis internas usadas dentro do sistema que armazenam e processam informações temporárias ou intermediárias. Essas Tags não estão associadas aos sensores de campo. No caso do projeto, foi utilizada para armazenar o valor digital das lâmpadas de cada quarto.

Em relação às Tags demo, geralmente são usadas para simulação de dados dentro do supervisório, também sem estarem conectadas com os sensores físicos ou fontes de dados reais. Essa Tag "Monitora" foi inserida para simular de forma aleatória o comportamento da frequência cardíaca de um paciente a partir de um gráfico gerado. As Tags Luz tem a função de armazenar os valores lógicos (referente a situação da luz ligada ou apagada) de cada ícone presente nas telas dos quartos do supervisório.

A Figura 27 mostra como foram inseridas as Tags internas e demo no sistema supervisório.

Figura 27 – Tags internas e demo.



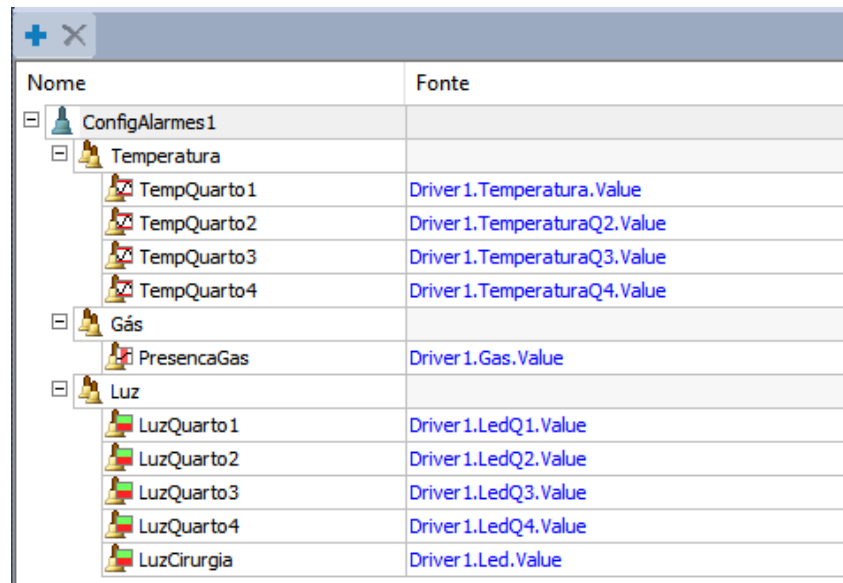
Nome	Valor
Dados	
Luz	Empty
LuzQ1	Empty
LuzQ2	Empty
LuzQ3	Empty
LuzQ4	Empty
Monitora	0

Fonte: Elaborado pelo autor.

### 5.3.4 Configuração dos alarmes

O projeto do hospital também conta com sistema de alarmes. Ele é responsável por gerenciar e monitorar todos os alarmes instalados no protótipo, bem como armazenar as informações em um histórico de alarmes inseridos em uma tela do supervisório. Para a aplicação foi necessário criar um servidor de alarmes para gerenciar e processar os dados configurados no próprio sistema. Após a inserção do servidor, os alarmes foram configurados, definindo as condições e ações específicas para cada alarme. A Figura 28 mostra como os alarmes estão separados em suas respectivas áreas.

Figura 28 – Alarmes configurados.



Nome	Fonte
ConfigAlarmes1	
Temperatura	
TempQuarto1	Driver 1.Temperatura.Value
TempQuarto2	Driver 1.TemperaturaQ2.Value
TempQuarto3	Driver 1.TemperaturaQ3.Value
TempQuarto4	Driver 1.TemperaturaQ4.Value
Gás	
PresencaGas	Driver 1.Gas.Value
Luz	
LuzQuarto1	Driver 1.LedQ1.Value
LuzQuarto2	Driver 1.LedQ2.Value
LuzQuarto3	Driver 1.LedQ3.Value
LuzQuarto4	Driver 1.LedQ4.Value
LuzCirurgia	Driver 1.Led.Value

Fonte: Elaborado pelo autor.

Para visualização do processo, na tela de monitoramento foi introduzido um visualizador de alarmes para que o usuário do sistema supervisório possa acompanhar a situação do ambiente hospitalar.

### 5.3.5 Desenvolvimento das telas

Para o desenvolvimento das telas do sistema supervisório foi levado em consideração uma interface que pudesse ser o mais simples e intuitiva possível, com o propósito de que todos que abrirem o supervisório consigam interagir sem a necessidade de um manual ou um passo a passo para o uso do mesmo. O código utilizado para realizar a comunicação entre o supervisório e o arduino está apresentado no Apêndice 7.

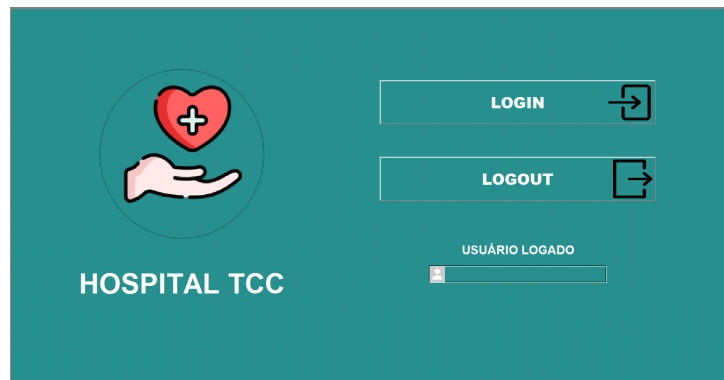
Dessa forma, para a construção de cada tela foi estudada uma maneira de demonstrar as aplicações mais próximas da realidade, principal objetivo do projeto.

Foram divididas quatro telas de aplicação. Uma tela de *Login* para que os usuários possam acessar o sistema supervisório, uma para o monitoramento geral do hospital que é a principal tela do supervisório, outra para visualizar as informações dos médicos e a quarta tela destinada a farmácia para controlar e armazenar os remédios do hospital.

#### 5.3.5.1 Tela de *Login*

Tela de login do sistema supervisório representada na Figura 29.

Figura 29 – Tela de login.



Fonte: Elaborado pelo autor.

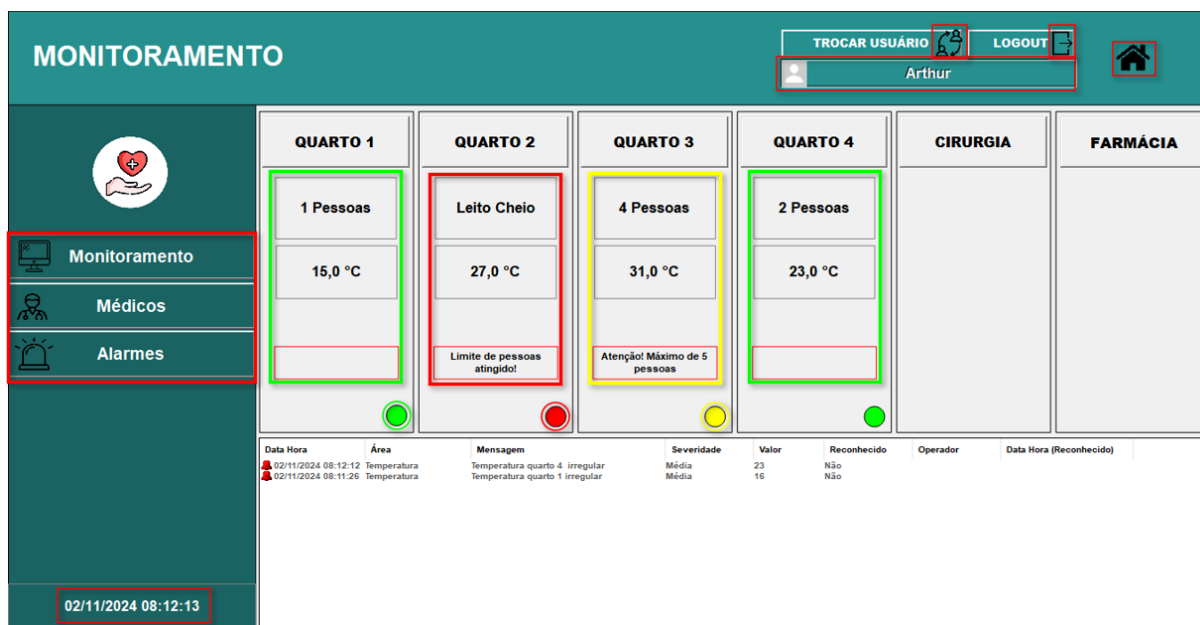
Nessa tela, a interface ficou distribuída de forma que ao usuário pressionar o botão de *Login* há o espaço para o preenchimento das credenciais. Na mesma tela é possível identificar qual o usuário está logado e caso precise sair, pode-se pressionar o botão de *Logout*.

### 5.3.5.2 Tela de Monitoramento

Na tela de monitoramento, foram desenvolvidas subtelas, nas quais estão dedicadas à visualização dos quartos, sendo quatro no total, a área de cirurgia, esta em específico possui o protótipo da cama desenvolvido já apresentado, uma tela para fazer a consulta dos alarmes gerados no ambiente e também, a tela para gerenciar a farmácia. Além disso, foram inseridos botões para que o usuário possa trocar de usuário e fazer o *logout* quando desejar, assim como na tela de *Login*.

A principal tela de monitoramento do sistema supervisorio está representada na Figura 30. Nela é possível identificar a quantidade de pessoas presente em cada quarto, bem como monitorar a temperatura e os alarmes do sistema (inferior da tela). Nela também é permitido, a partir de botões inseridos no ambiente, trocar de tela ao pressioná-los. O retângulo vermelho exibe uma mensagem de atenção quando o número de pessoas dentro do quarto passam de três e o círculo em verde, em seu formato padrão, informa que o leito não está em sua lotação máxima. Conforme a quantidade de pessoas vai subindo, esse ícone troca de cor para amarelo para indicar um ponto de atenção e vermelho quando chega ao limite de indivíduos.

Figura 30 – Tela de monitoramento principal.



Fonte: Elaborado pelo autor.

A Figura 31 mostra como o sistema supervisor se comporta com o acionamento e notificações de alguns alarmes. Nessa aba é possível identificar a data e hora da ativação, a área de instalação do alarme. Há também a severidade e se o alarme foi reconhecido pelo usuário logado. Os textos em vermelho indicam alarmes irregulares e os amarelos informam que a situação está regular.

Figura 31 – Indicação do acionamento dos alarmes no sistema supervisor

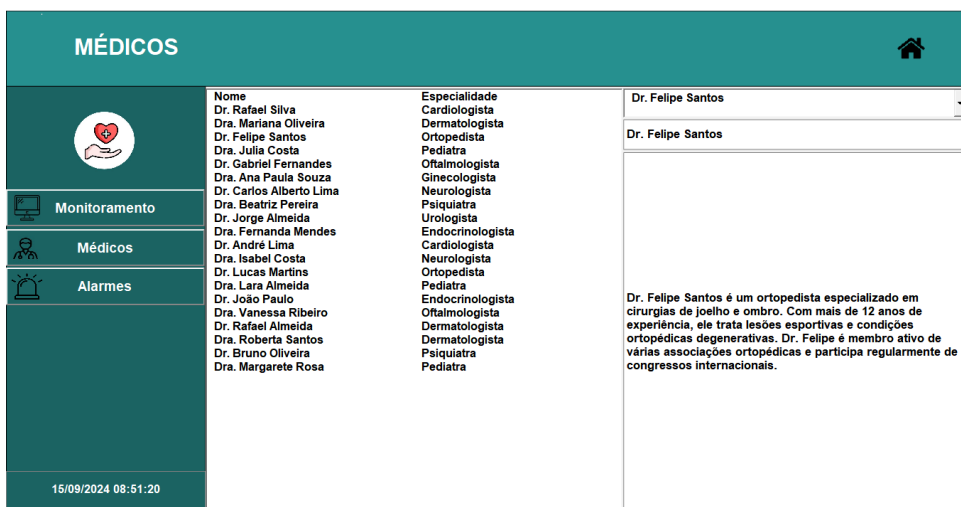
Data Hora	Área	Mensagem	Severidade	Valor	Reconhecido	Operador	Data Hora (Reconhecido)
13/09/2024 23:03:01	Gás	Sem presença de Gás	Crítica	176	Não		
13/09/2024 22:54:29	Luz	Luz do quarto 1 apagada	Baixa	0	Não		
13/09/2024 20:00:50	Luz	Luz do quarto 4 apagada	Alta	0	Não		
13/09/2024 20:00:47	Luz	Luz do quarto 3 apagada	Alta	0	Não		
13/09/2024 20:00:44	Luz	Luz do quarto 2 apagada	Baixa	0	Não		
13/09/2024 19:50:29	Temperatura	Temperatura quarto 4 irregular	Alta	85	Não		
13/09/2024 19:50:29	Temperatura	Temperatura quarto 2 irregular	Alta	88	Não		
13/09/2024 19:50:28	Temperatura	Temperatura quarto 3 irregular	Alta	96	Não		
13/09/2024 19:50:21	Luz	Luz do quarto 1 apagada	Baixa	0	Não		
13/09/2024 19:11:26	Temperatura	Temperatura irregular	Alta	78	Não		

Fonte: Elaborado pelo autor.

### 5.3.5.3 Tela dos Médicos

A tela de médicos, demonstrada na Figura 32 mostra a lista de médicos do hospital. Vale ressaltar que todos os nomes presentes são fictícios e estão apenas para representação. Ao clicar na caixa de texto em forma de lista, podemos ver as informações do médico escolhido.

Figura 32 – Tela da lista de médicos.

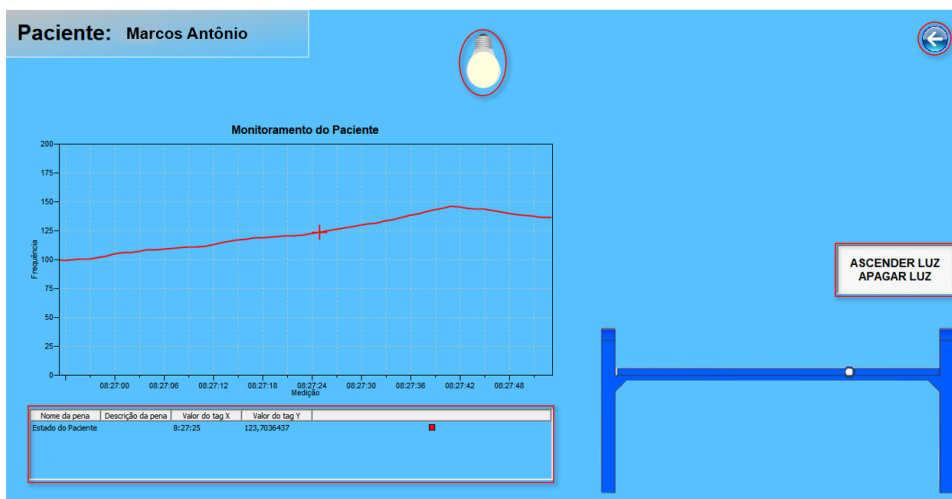


Fonte: Elaborado pelo autor.

### 5.3.5.4 Tela dos Quartos

A Figura 33 mostra como ficou a versão final das telas dos quartos do sistema supervisorio.

Figura 33 – Tela dos quartos.



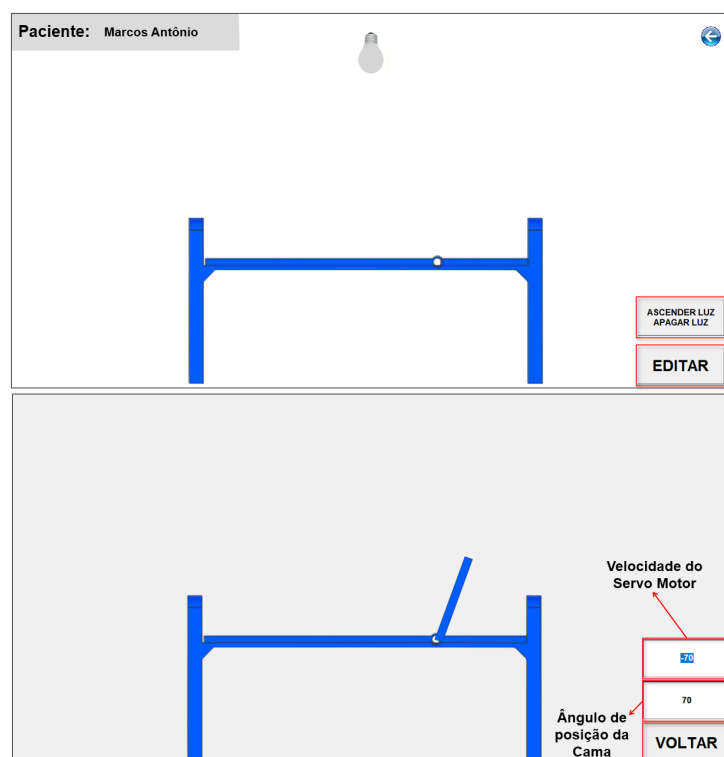
Fonte: Elaborado pelo autor.

A interface das telas dos quartos possui um botão para acender e apagar a luz quando necessário. Na mesma tela pode-se notar o ícone de uma lâmpada que ao acender muda de cor para um branco mais destacado. Também é possível identificar qual o paciente está presente naquele quarto em específico. O gráfico presente na tela, representa os batimentos cardíacos do paciente que está sendo monitorado, enquanto a seta em azul pode ser pressionada para caso o usuário queira voltar à tela anterior.

### 5.3.5.5 Tela de Cirurgia

A tela de cirurgia, retratada na Figura 34 é semelhante a tela dos quartos. A diferença é que ela pode ser configurada a partir de um botão "Editar". Na tela de configuração é possível ajustar a posição da cama através da velocidade inserida no primeiro retângulo. A partir da velocidade informada, é mostrado no indicador de baixo qual o ângulo que a cama está sendo ajustada. Vale ressaltar que as velocidade devem ser inseridas de forma negativa, caso o paciente queira subir a cabeceira da cama e positiva, para o sentido inverso. Quanto maior a velocidade, maior é o ângulo de posição da cama ajustado.

Figura 34 – Tela de cirurgia e configuração da cama.



Fonte: Elaborado pelo autor.

### 5.3.5.6 Tela da Farmácia

A Figura 35 mostra a versão final da tela da farmácia, simulando um sistema de estoque dos medicamentos.

Figura 35 – Tela da farmácia.



Fonte: Elaborado pelo autor.

Na tela da farmácia, todos os remédios que chegam no estabelecimento possui uma *Tag* associada a ele. Ao passar essas *Tags* no leitor presente no ambiente, é possível identificar qual o remédido está sendo recebido pela farmácia e em qual prateleira ele se encontra.

### 5.3.5.7 Tela de Alarmes

Na tela de alarmes, ilustrada na Figura 36, é possível identificar o histórico de todos os alarmes que foram disparados durante todo o período do sistema supervisorio.

Figura 36 – Tela dos quartos.

The screenshot shows a web interface titled "CONSULTA DE ALARMES". On the left, there is a sidebar with navigation icons and labels: "Monitoramento", "Médicos", and "Alarmes". The main area displays a table of alarm events. Below the table, there is a summary section with columns for "Data Hora", "Área", "Mensagem", "Severidade", "Valor", "Reconhecido", "Operador", and "Data Hora (Reconhecido)".

E3TimeStamp	Area	Message	Sev...	For...	Acked	AckTime	ActorID
13/09/2024 23:03:02	Gás	Sem presença de Gás	-2	176	0	00:00:00	
13/09/2024 23:02:54	Gás	Há um vazamento de Gás no Hos...	-2	555	0	00:00:00	
13/09/2024 23:02:51	Gás	Sem presença de Gás	-2	815	1	13/09/2024 23:02:51	Arthur
13/09/2024 23:02:51	Gás	Há um vazamento de Gás no Hos...	-2	756	1	13/09/2024 23:02:51	Arthur
13/09/2024 23:02:50	Gás	Há um vazamento de Gás no Hos...	-2	756	0	00:00:00	
13/09/2024 23:02:47	Gás	Sem presença de Gás	-2	807	0	00:00:00	
13/09/2024 23:02:45	Gás	Há um vazamento de Gás no Hos...	-2	919	0	00:00:00	
13/09/2024 23:02:44	Gás	Sem presença de Gás	-2	819	0	00:00:00	
13/09/2024 23:02:43	Gás	Há um vazamento de Gás no Hos...	-2	770	0	00:00:00	
13/09/2024 23:02:42	Gás	Sem presença de Gás	-2	904	0	00:00:00	
13/09/2024 23:02:38	Gás	Há um vazamento de Gás no Hos...	-2	995	0	00:00:00	
13/09/2024 22:54:29	Luz	Luz do quarto 1 apagada	2	0	0	00:00:00	
13/09/2024 20:01:18	Luz	Luz da cirurgia apagada	2	0	1	13/09/2024 20:01:18	(Sem usuário)
13/09/2024 20:00:53	Luz	Luz da cirurgia apagada	2	0	0	00:00:00	
13/09/2024 20:00:50	Luz	Luz do quarto 4 apagada	0	0	0	00:00:00	
13/09/2024 20:00:47	Luz	Luz do quarto 3 apagada	0	0	0	00:00:00	

Registro	Data Hora	Área	Mensagem	Severidade	Valor	Reconhecido	Operador	Data Hora (Reconhecido)
1	13/09/2024 23:03:01	Gás	Sem presença de Gás	Crítica	176	Não		
2	13/09/2024 22:54:29	Luz	Luz do quarto 1 apagada	Baixa	0	Não		
3	13/09/2024 20:00:50	Luz	Luz do quarto 4 apagada	Alta	0	Não		
4	13/09/2024 20:00:47	Luz	Luz do quarto 3 apagada	Alta	0	Não		
5	13/09/2024 20:00:44	Luz	Luz do quarto 2 apagada	Baixa	0	Não		
6	13/09/2024 19:50:29	Temperatura	Temperatura quarto 4 irregular	Alta	85	Não		
7	13/09/2024 19:50:29	Temperatura	Temperatura quarto 2 irregular	Alta	88	Não		
8	13/09/2024 19:50:28	Temperatura	Temperatura quarto 3 irregular	Alta	96	Não		
9	13/09/2024 19:50:24	Luz	Luz do quarto 1 apagada	Baixa	0	Não		
10	13/09/2024 19:11:26	Temperatura	Temperatura irregular	Alta	78	Não		

Fonte: Elaborado pelo autor.

Nessa tela pode-se verificar em qual horário e qual foi o alarme disparado, além de apontar o responsável pelo reconhecimento e o horário no qual o usuário está reconhecendo aquele alarme. Reconhecer um alarme significa que o usuário está ciente do que está acontecendo no estabelecimento.

## 6 CONCLUSÃO

O desenvolvimento deste trabalho não foi isento de desafios, como o primeiro contato com softwares não aprendidos durante a formação do autor e diversas outras aplicações relevantes para o projeto, mas proporcionou um aprendizado valioso sobre sistemas supervisórios e sua integração com o arduino. No trabalho, também foi desenvolvido protótipos para melhor demonstrar o funcionamento do sistema completo em escala reduzida.

O uso do sistema supervisório, relacionado à automação de processos permite o gerenciamento mais centralizado, garantindo maior segurança e precisão no acompanhamento dos parâmetros críticos do hospital.

A implementação de tecnologias de automação em ambientes hospitalares, estudados ao longo do trabalho, contribuiu, de certa forma, para uma melhor desfecho do projeto.

A conclusão geral do projeto proposto é que foi bem sucedido, considerando o principal objetivo de permitir que o hospital possa monitorar os ambientes de sua instalação por meio de um sistema supervisório.

Embora o resultado do trabalho tenha sido satisfatório, algumas dificuldades foram enfrentadas ao longo do desenvolvimento. A comunicação do arduino com o software Eclipse E3 possui um limite de variáveis que pode ser usada, o que não permite uma aplicação mais completa do sistema. Dessa forma, faz-se necessário a garantia de uma licença do próprio sistema para que as ferramentas de uso sejam mais completas. Além disso, nota-se que o projeto em escala real terá um custo mais elevado, visto que é preciso adequar as instalações do hospital com os sensores voltados para um serviço mais industrial, com intuito de suportar a implementação do supervisório. Destaca-se também um atraso de alguns milissegundos na comunicação entre o arduino e o supervisório, mas que não interferem a aplicação do projeto.

O trabalho evidencia o grande potencial que a automação oferece à saúde, especialmente quando tratamos da redução de problemas enfrentados por um hospital. O projeto também reforça a importância da contínua pesquisa e desenvolvimentos tecnológicos, visando uma maior integração entre a tecnologia e a medicina.

### 6.1 Trabalhos futuros

Várias áreas do projeto em questão podem ser aprimoradas. Em relação ao protocolo de comunicação, pode-se utilizar outros protocolos que sejam capazes de se comunicar com o sistema supervisório, como por exemplo, o protocolo CAN, mais adequado para ambientes que necessitam de comunicação rápida e robusta entre vários dispositivos.

Além disso, a aplicação do sistema envolvendo a internet das coisas pode ser um grande avanço para o projeto, pois as informações podem ser coletadas e transmitidas em tempo real. Outra vantagem do uso da IoT está nos grandes volumes de dados que são gerados e coletados,

utilizando uma integração de dados (*Data Analytics e Big Data*), bem como a manutenção preditiva com as análises dos sensores instalados.

Destaca-se também a possível incorporação de um CLP, que permite um sistema industrial mais robusto, possuindo mais capacidade de expansão, de processamento e memória. O uso do CLP possibilita a utilização dos dispositivos por mais tempo, por possuir um bom ciclo de vida (feito para funcionar de forma contínua por anos sem falhas) e por ter alta confiabilidade.

Com o CLP, pode-se integrar sensores industriais alternativos que possam comunicar com o dispositivo para garantir ainda mais robustez na aplicação. Como opções de sensores estão os sensores termopares e termoresistores para medição de temperatura, detectores de proximidade, como sensores de temperatura infravermelho (PIR - Passive Infrared) que detectam mudanças de calor emitido pelo corpo humano. Outra solução interessante é a utilização de câmeras para monitorar os ambientes integrados ao supervisório, entre outras possibilidades industriais.

No que diz respeito ao aprimoramento das telas pode ser alterado de acordo com a necessidade da aplicação de cada estabelecimento, trazendo mais dinamismo para sua funcionalidade.

Sugere-se que algumas dessas informações sejam implementadas, pois garantem que futuros hospitais sejam cada vez mais eficientes e seguros com esse tipo de sistema integrado. A ideia do projeto é que, futuramente, possa ser expandido e focado no universo da Internet das Coisas, permitindo a troca de dados sem qualquer intervenção humana e possibilitando ainda mais uma melhor comunicação entre os dispositivos, a fim de tornar o sistema mais robusto.

## REFERÊNCIAS

- ANDRADE, N. G. de. **Automação de Leitos Hospitalares Voltada ao Paciente**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Eng. de Controle e Automação, Porto Alegre, 2019. Citado na página 13.
- ASHTON, K. That ‘internet of things’ thing. **RFiD Journal**, v. 22, n. 7, p. 97–114, 2009. Citado na página 25.
- BALBI, P. A.; RIBEIRO, P. C. C. Monografia, **Análise da Tecnologia RFID nas Operações Hospitalares**. Niterói: [s.n.], 2018. Citado na página 12.
- BÁRBARA, L. K.; GABRIEL, N. E. **Automação do processo de conferência dos itens de carros de emergência em um hospital**. 2017. São Paulo. Citado na página 7.
- CORREIA Érica C.; OLIVEIRA, B. Q. de; MEDEIROS, A. M. de M.; PAGOTTI, L. F. **Engenharia Elétrica**. 2021. PUC Goiás Goiânia. Citado na página 12.
- DATE, C. J. **Bancos de Dados: Fundamentos**. Rio de Janeiro: Ed. Campus, 1985. Citado na página 14.
- DONG, J. Q.; WU, W.; ZHANG, Y. S. The faster the better? innovation speed and user interest in open source software. **Information & Management**, v. 55, n. 7, p. 841–854, 2018. Citado na página 15.
- FALCAO, J. **Hospitais de São Paulo investem em farmácias inteligentes com robôs**. 2015. Acessado em: 8 de Agosto de 2024. Disponível em: <<https://oglobo.globo.com/saude/hospitais-de-sao-paulo-investem-em-farmacias-inteligentes-com-robos-16804745#:~:text=S%C3%83O%20PAULO%20%E2%80%94%20Hospitais%20como%20S%C3%ADrio,o%20m%C3%ADnimo%20de%20contato%20humano.>> Citado na página 7.
- FILHO, M. F. Bms 2.0 - nova geração de sistemas de automação e gestão predial. In: ARANDA EVENTOS. **Congresso Netcom**. São Paulo, 2015. Citado na página 25.
- GAMBUS, P.; SHAFER, S. Artificial intelligence for everyone. **Anesthesiology**, v. 128, n. 3, p. 431–433, Mar 2018. Citado na página 13.
- GEDDES, M. **Arduino Project Handbook, Vol. 1: 25 Practical Projects to Get You Started**. [S.l.: s.n.], 2016. ISBN: 978-1-59327-690-4. Citado 2 vezes nas páginas 19 e 21.
- GONSALVES, A. d. S.; SISCOOTTO, R. A. Solução de baixo custo para o monitoramento de sinais vitais, em tempo real, fazendo uso de sensores e arduino. **Colloquium Exactarum**. ISSN: 2178-8332, v. 12, n. 2, p. 102–118, Dezembro. 2020. Disponível em: <<https://revistas.unoeste.br/index.php/ce/article/view/3817>>. Citado na página 7.
- HAN, E.; YEO, S.; KIM, M. *et al.* Medical education trends for future physicians in the era of advanced technology and artificial intelligence: an integrative review. **BMC Medical Education**, v. 19, p. 460, 2019. Citado na página 13.
- JUNIOR, J. **Projeto transforma em lei o direito do paciente hospitalar a acompanhante, Fonte: Agência Câmara de Notícias**. 2017. Disponível em: <<https://www.camara.leg.br/noticias/507307-projeto-transforma-em-lei-o-direito-do-paciente-hospitalar-a-acompanhante/>>. Citado na página 8.

KORTH, H.; SILBERSCHATZ, A. **Sistemas de Bancos de Dados**. 2ª edição revisada. ed. [S.l.]: Makron Books, 1994. Citado na página 14.

LEITE, C.; ARAUJO, B.; VALENTIM, M.; BRANDÃO, G.; GUEIRREIRO, A. Novas tecnologias para automação hospitalar. January 2015. Citado na página 12.

LEITE, J. E.; MARTINS, P. S.; URSINI, E. L. A internet das coisas (iot) : Tecnologias e aplicações. In: SCHOOL OF TECHNOLOGY, UNIVERSITY OF CAMPINAS (UNICAMP). **Proceedings of the 2017 Brazilian Technology Symposium**. Limeira-SP, Brazil, 2017. Citado na página 25.

MANYIKA, J.; AL. et. **Disruptive Technologies: Advances that will transform life, business, and the global economy**. 2013. Citado na página 25.

MCCROHAN. 2011. Citado na página 14.

MCROBERTS, M. **Arduino básico**. [tradução rafael zanolli]. São Paulo: Novatec Editora, 2011. 22–24 p. Citado na página 16.

NITZAN, D.; ROSEN, C. Programmable industrial automation. **Transactions on Computers**, C-25, n. 12, p. 1259–1270, Dec. 1976. Citado na página 7.

PLCOPEN. **IEC 61131-3: a norma para programação**. [S.l.], 2003. Citado na página 17.

PRONTPED. **Diferenças entre Prontuário Físico e Eletrônico: Entenda as Vantagens e Desvantagens de cada (2023)**. 2022. Acessado em: 8 de Junho de 2024. Disponível em: <<https://blog.prontped.com.br/diferencas-entre-prontuario-fisico-e-eletronico/>>. Citado na página 12.

ROSE, K.; ELDRIDGE, S.; CHAPIN, L. **The Internet of Things: An Overview**. Geneva, Switzerland: The Internet Society (ISOC), 2015. Citado na página 25.

SALOMI, M. J. A.; MACIEL, R. F. Gestão de documentos e automação de processos em uma instituição de saúde sem papel. **J. Health Inform.**, v. 8, n. 1, p. 31–8, Janeiro-Março 2016. Citado na página 7.

SIMOES, J. D. S.; FERNANDES, A. M.; SEABRA, E. A. R. de. Use arduino as a modbus slave over rtu. 2016. Citado na página 14.

SOUZA, A. R. e. a. A placa arduino: uma opção de baixo custo para experiências de física assistidas pelo pc. **Revista Brasileira de Ensino de Física**, São Paulo, v. 33, n. 1, p. 1702, jan./mar. 2011. Citado na página 15.

SWEDBERG, C. **Solução Reduz Custos do Hospital de Virgínia**. 2018. <<https://brasil.rfidjournal.com/estudos-de-caso/vision?16631>>. Acessado em: 18, outubro, 2023. Citado na página 12.

TELEMEDICINA, P. **Telecirurgia: Revolucionando a Medicina com Cirurgias à Distância**. 2024. Disponível em: <<https://portaltelemedicina.com.br/telecirurgia-revolucionando-a-medicina-com-cirurgias-a-distancia>>. Citado na página 12.

VIANNA, W.; BRINGHENTI, P.; MARTINS, L. **Sistema SCADA Supervisório**. Campos dos Goytacazes, Rio de Janeiro: [s.n.], 2008. Instituto Federal Fluminense de Educação Ciência e Tecnologia. Citado na página 26.

ÅSTROM, K. J.; HAGGLUND, T. The future of pid control. **Control Engineering Practice**, v. 9, n. 11, p. 1163–1175, 2001. Citado na página 19.

## 7 APÊNDICE A - CÓDIGO ARDUINO PARA A COMUNICAÇÃO COM O SISTEMA SUPERVISÓRIO

Código arduino utilizado para fazer a comunicação dos sensores instalados no protótipo.

```

/*****
    PROJETO 2 - TCC HOSPITAL
    ARTHUR HENRIQUE PIMENTA SILVA
    ARDUINO - COMUNICACAO SERIAL (MODBUS) - PROJETO TCC
*****/

/* Inclusão de Bibliotecas */
#include <Modbusino.h>
#include <SPI.h> //INCLUSÃO DE BIBLIOTECA
#include <MFRC522.h> //INCLUSÃO DE BIBLIOTECA
#include <Servo.h> // Inclusão da biblioteca Servo

/* Inicializa o ID do Dispositivo como 1 */
ModbusinoSlave modbusino_slave(1);

/* Aloca o número de Registradores = 20 */
uint16_t tab_reg[20];

/* Variaveis do Projeto */

/* RFID */
#define SS_PIN 10 // PINO SDA - RFID
#define RST_PIN 9 // PINO DE RESET - RFID
MFRC522 rfid(SS_PIN, RST_PIN); // PASSAGEM DE PARÂMETROS REFERENTE AOS PINOS
const int Cartoes = 7;
String nomes[Cartoes]; // Array para armazenar nomes
int CartoesArmazenados = 0; // Contador de nomes armazenados
bool chave7 = false;

/* SERVO MOTOR 996R */
const int pinoServo = 11; // Pino digital utilizado pelo servo
const int velocidadeMaxima = 180; // Velocidade máxima do servo
Servo s; // Objeto do tipo Servo
int velocidade; // Velocidade desejada para o servo

```

```
//const int LED = 2;
//int estado_led;

/* LMR35 - TEMPERATURA */
const int LM35 = A2; // Pino Analógico para conectar o LM35
const int LM35_2 = A3;
const int LM35_3 = A4;
const int LM35_4 = A5;
float temperatura = 0;
float temperatura2 = 0;
float temperatura3 = 0;
float temperatura4 = 0;

//int Comando = 0;
//int BuzzerBot = 0;
//const int Botao = 2;
//const int Buzzer = 50; // Pino Digital para conectar o Buzzer
//int estado_botao = 0;

/* LEDs - LUZ/LÂMPADA */
const int pinoLed = 12; // Pino Digital para conectar o LED
const int pinoLed2 = 13; // Pino Digital para conectar o LED
const int pinoLedLuz = 37; // Pino Digital para conectar o LED
const int pinoLedLuz2 = 38; // Pino Digital para conectar o LED
const int pinoLedLuz3 = 39; // Pino Digital para conectar o LED
const int pinoLedLuz4 = 40; // Pino Digital para conectar o LED
const int pinoLedLuz5 = 41; // Pino Digital para conectar o LED

/* INFRAVERMELHO - PRESENÇA */
const int Infravermelho1 = 8; // Pino Digital para conectar o Infravermelho 1
const int Infravermelho2 = 22; // Pino Digital para conectar o Infravermelho 2
const int Infravermelho3 = 23; // Pino Digital para conectar o Infravermelho 3
const int Infravermelho4 = 24; // Pino Digital para conectar o Infravermelho 4
const int Infravermelho5 = 25; // Pino Digital para conectar o Infravermelho 5
const int Infravermelho6 = 26; // Pino Digital para conectar o Infravermelho 6
const int Infravermelho7 = 27; // Pino Digital para conectar o Infravermelho 7
const int Infravermelho8 = 28; // Pino Digital para conectar o Infravermelho 8
```

```
bool statusIR1 = true; // normalmente em HIGH
bool statusIR12 = true; // normalmente em HIGH
bool statusIR23 = true; // normalmente em HIGH
bool statusIR24 = true; // normalmente em HIGH
bool statusIR25 = true; // normalmente em HIGH
bool statusIR26 = true; // normalmente em HIGH
bool statusIR27 = true; // normalmente em HIGH
bool statusIR28 = true; // normalmente em HIGH

bool statusIR1_anterior = true; // normalmente em HIGH
bool statusIR12_anterior = true; // normalmente em HIGH
bool statusIR23_anterior = true; // normalmente em HIGH
bool statusIR24_anterior = true; // normalmente em HIGH
bool statusIR25_anterior = true; // normalmente em HIGH
bool statusIR26_anterior = true; // normalmente em HIGH
bool statusIR27_anterior = true; // normalmente em HIGH
bool statusIR28_anterior = true; // normalmente em HIGH

int contador = 0; // Variável para contagem de pessoas
int contador2 = 0; // Variável para contagem de pessoas
int contador3 = 0; // Variável para contagem de pessoas
int contador4 = 0; // Variável para contagem de pessoas

/* SENSOR DE GÁS + BUZZER - IDENTIFICAÇÃO DE GÁS E ALARME */
int buzzerPin = 42; // Pino Digital para conectar o Buzzer
//const int MQ2 = 52; // Pino Digital para conectar o MQ2
const int MQ2_Analogico = A0; // Pino Analógico para conectar o MQ2
int valor_analogico;
int Nivel_Gas = 270; // Variável para comparar o nível de gás do ambiente
bool estadoled = 0; // Estado do LED inicial (Desligado)

void setup()
{
  /* Definindo a taxa de transferencia em bauds */
  modbusino_slave.setup(9600);

  /* Definindo o comportamento dos pinos */
  Serial.begin(9600); //INICIALIZAÇÃO DA SERIAL
```

```
s.attach(pinoServo);
s.write(90); // Inicializa o motor na posição de parada (0 velocidade)

//pinMode(LED,OUTPUT); // SAÍDA
//pinMode(Botao,INPUT); // ENTRADA
//pinMode(Buzzer,OUTPUT); // SAÍDA

pinMode(pinoLedLuz, OUTPUT); // SAÍDA
digitalWrite(pinoLedLuz, LOW); //LED INICIA DESLIGADO

pinMode(pinoLedLuz2, OUTPUT); // SAÍDA
digitalWrite(pinoLedLuz2, LOW); //LED INICIA DESLIGADO

pinMode(pinoLedLuz3, OUTPUT); // SAÍDA
digitalWrite(pinoLedLuz3, LOW); //LED INICIA DESLIGADO

pinMode(pinoLedLuz4, OUTPUT); // SAÍDA
digitalWrite(pinoLedLuz4, LOW); //LED INICIA DESLIGADO

pinMode(pinoLedLuz5, OUTPUT); // SAÍDA
digitalWrite(pinoLedLuz5, LOW); //LED INICIA DESLIGADO

pinMode(buzzerPin,OUTPUT); // SAÍDA
//pinMode(MQ2_Analogico, INPUT); // ENTRADA
//pinMode(MQ2, INPUT);

pinMode(Infravermelho1, INPUT); // ENTRADA
pinMode(Infravermelho12, INPUT); // ENTRADA
pinMode(Infravermelho23, INPUT); // ENTRADA
pinMode(Infravermelho24, INPUT); // ENTRADA
pinMode(Infravermelho25, INPUT); // ENTRADA
pinMode(Infravermelho26, INPUT); // ENTRADA
pinMode(Infravermelho27, INPUT); // ENTRADA
pinMode(Infravermelho28, INPUT); // ENTRADA

//pinMode(Botao, INPUT_PULLUP); // define o pino do botao como entrada

SPI.begin(); //INICIALIZA O BARRAMENTO SPI
```

```
    rfid.PCD_Init(); //INICIALIZA MFRC522
}

void loop()
{
    modbusino_slave.loop(tab_reg, 20);
    /*
    Exemplo de como passar valores para os registradores
    tab_reg[0] = 600;

    Exemplo de como recuperar valores dos registradores
    Variavel = tab_reg[1];
    */

    /*
    Comando = tab_reg[1];
    if (Comando <= 0)digitalWrite(LED, LOW);
    if (Comando >= 1)digitalWrite(LED, HIGH);
    */

    // LEDs - Lâmpada acesa ou apagada
    if (tab_reg[9] == 1)
    {
        digitalWrite(pinoLedLuz, HIGH);
    }
    else
    {
        digitalWrite(pinoLedLuz, LOW);
    }

    if (tab_reg[10] == 1)
    {
        digitalWrite(pinoLedLuz2, HIGH);
    }
    else
    {
        digitalWrite(pinoLedLuz2, LOW);
    }
}
```

```
if (tab_reg[11] == 1)
{
    digitalWrite(pinoLedLuz3, HIGH);
}
else
{
    digitalWrite(pinoLedLuz3, LOW);
}

if (tab_reg[12] == 1)
{
    digitalWrite(pinoLedLuz4, HIGH);
}
else
{
    digitalWrite(pinoLedLuz4, LOW);
}

if (tab_reg[13] == 1)
{
    digitalWrite(pinoLedLuz5, HIGH);
}
else
{
    digitalWrite(pinoLedLuz5, LOW);
}

// Servo Motor 996R - Ajuste de posição da Cama
velocidade = tab_reg[8];

if (velocidade >= -velocidadeMaxima && velocidade <= velocidadeMaxima) {
    /* Ajusta a posição do servo motor */
    s.write(map(velocidade, -velocidadeMaxima, velocidadeMaxima, 0, 180));
}

// Sensor LM35 - Temperatura
```

```
temperatura = (float(analogRead(LM35))) * 1100 / (1023 * 10); //VARIÁVEL RECEBE A TEMPERATURA
//temperatura = (float(analogRead(LM35)) * (5/1023))/0.01; //VARIÁVEL RECEBE A TEMPERATURA
Serial.print("Temperatura = "); //IMPRIME O TEXTO NA SERIAL
Serial.print(temperatura); //IMPRIME NA SERIAL A TEMPERATURA MEDIDA
Serial.println(" C"); //IMPRIME O TEXTO NA SERIAL
//delay(2000); //INTERVALO DE 2 SEGUNDOS
tab_reg[0] = temperatura;

temperatura2 = (float(analogRead(LM35_2))) * 1100 / (1023 * 10); //VARIÁVEL RECEBE A TEMPERATURA
//temperatura = (float(analogRead(LM35)) * (5/1023))/0.01; //VARIÁVEL RECEBE A TEMPERATURA
Serial.print("Temperatura = "); //IMPRIME O TEXTO NA SERIAL
Serial.print(temperatura2); //IMPRIME NA SERIAL A TEMPERATURA MEDIDA
Serial.println(" C"); //IMPRIME O TEXTO NA SERIAL
//delay(2000); //INTERVALO DE 2 SEGUNDOS
tab_reg[14] = temperatura2;

temperatura3 = (float(analogRead(LM35_3))) * 1100 / (1023 * 10); //VARIÁVEL RECEBE A TEMPERATURA
//temperatura = (float(analogRead(LM35)) * (5/1023))/0.01; //VARIÁVEL RECEBE A TEMPERATURA
Serial.print("Temperatura = "); //IMPRIME O TEXTO NA SERIAL
Serial.print(temperatura3); //IMPRIME NA SERIAL A TEMPERATURA MEDIDA
Serial.println(" C"); //IMPRIME O TEXTO NA SERIAL
//delay(2000); //INTERVALO DE 2 SEGUNDOS
tab_reg[15] = temperatura3;

temperatura4 = (float(analogRead(LM35_4))) * 1100 / (1023 * 10); //VARIÁVEL RECEBE A TEMPERATURA
//temperatura = (float(analogRead(LM35)) * (5/1023))/0.01; //VARIÁVEL RECEBE A TEMPERATURA
Serial.print("Temperatura = "); //IMPRIME O TEXTO NA SERIAL
Serial.print(temperatura4); //IMPRIME NA SERIAL A TEMPERATURA MEDIDA
Serial.println(" C"); //IMPRIME O TEXTO NA SERIAL
//delay(2000); //INTERVALO DE 2 SEGUNDOS
tab_reg[16] = temperatura4;

// Sensor Infravermelho 1 - Contagem de pessoas (Entrada +)

statusIR1 = digitalRead(Infravermelho1);
statusIR12 = digitalRead(Infravermelho12);
statusIR23 = digitalRead(Infravermelho23);
statusIR24 = digitalRead(Infravermelho24);
```

```
statusIR25 = digitalRead(Infravermelho25);
statusIR26 = digitalRead(Infravermelho26);
statusIR27 = digitalRead(Infravermelho27);
statusIR28 = digitalRead(Infravermelho28);

if(statusIR1 == false && statusIR1_anterior == true) //SE A LEITURA DO PINO FOR IG
{
    //digitalWrite(pinoLed, HIGH); //ACENDE O LED
    contador = contador + 1;
    //Serial.println(contador);
    //delay(2000);
    //Serial.print(" ");
    tab_reg[2] = contador;
}

if(statusIR23 == false && statusIR23_anterior == true) //SE A LEITURA DO PINO FOR
{
    //digitalWrite(pinoLed, HIGH); //ACENDE O LED
    contador2 = contador2 + 1;
    //Serial.println(contador2);
    //delay(2000);
    //Serial.print(" ");
    tab_reg[5] = contador2;
}

if(statusIR25 == false && statusIR25_anterior == true) //SE A LEITURA DO PINO FOR
{
    //digitalWrite(pinoLed, HIGH); //ACENDE O LED
    contador3 = contador3 + 1;
    //Serial.println(contador3);
    //delay(2000);
    //Serial.print(" ");
    tab_reg[6] = contador3;
}

if(statusIR27 == false && statusIR27_anterior == true) //SE A LEITURA DO PINO FOR
{
    //digitalWrite(pinoLed, HIGH); //ACENDE O LED
    contador4 = contador4 + 1;
```

```
    //Serial.println(contador4);
    //delay(2000);
    //Serial.print(" ");
    tab_reg[7] = contador4;
}

//Sensor Infravermelho 2 - Contagem de pessoas (Saída -)
if(statusIR12 == false && statusIR12_anterior == true)
{
    //digitalWrite(pinoLed, HIGH);
    contador = contador - 1;
    //Serial.println(contador);
    //delay(2000);
    tab_reg[2] = contador;
}

if(statusIR24 == false && statusIR24_anterior == true)
{
    //digitalWrite(pinoLed, HIGH);
    contador2 = contador2 - 1;
    //Serial.println(contador2);
    //delay(2000);
    tab_reg[5] = contador2;
}

if(statusIR26 == false && statusIR26_anterior == true)
{
    //digitalWrite(pinoLed, HIGH);
    contador3 = contador3 - 1;
    //Serial.println(contador3);
    //delay(2000);
    tab_reg[6] = contador3;
}

if(statusIR28 == false && statusIR28_anterior == true)
{
    //digitalWrite(pinoLed, HIGH);
    contador4 = contador4 - 1;
    //Serial.println(contador4);
```

```
    //delay(2000);
    tab_reg[7] = contador4;
}

// Contagem de pessoas permanece em 0 se a contagem estiver abaixo de 0
if(contador < 0)
{
    contador = 0;
    //Serial.println(contador);
    //delay(2000);
    tab_reg[2] = contador;
}

if(contador2 < 0)
{
    contador2 = 0;
    //Serial.println(contador2);
    //delay(2000);
    tab_reg[5] = contador2;
}

if(contador3 < 0)
{
    contador3 = 0;
    //Serial.println(contador3);
    //delay(2000);
    tab_reg[6] = contador3;
}

if(contador4 < 0)
{
    contador4 = 0;
    //Serial.println(contador4);
    //delay(2000);
    tab_reg[7] = contador4;
}

statusIR1_anterior = statusIR1;
statusIR12_anterior = statusIR12;
```

```
statusIR23_anterior = statusIR23;
statusIR24_anterior = statusIR24;
statusIR25_anterior = statusIR25;
statusIR26_anterior = statusIR26;
statusIR27_anterior = statusIR27;
statusIR28_anterior = statusIR28;

//Serial.println("statusIR1 = " + String(statusIR1));
//dSerial.println("statusIR1 anterior = " + String(statusIR1_anterior));

// Sensor MQ2 - Gás e chamas
//int valor_digital = digitalRead(MQ2);
valor_analogico = analogRead(MQ2_Analogico); // VALOR DO GÁS MEDIDO
//Serial.println(valor_digital);
Serial.println(valor_analogico);
if (valor_analogico > Nivel_Gas)
{
  Serial.println("tem gas");
  //delay(50);
  tone(buzzerPin,3000,100);
  delay(300);
  //digitalWrite(pinoLed2, HIGH);
}

/*
if(digitalRead(MQ2) == LOW)
{
  digitalWrite(pinoLed, HIGH);
  tone(buzzerPin,700,200);
}
*/

//tab_reg[3] = Sensor;
tab_reg[3] = valor_analogico;

// RFID - Acesso de pessoas (Pacientes)

if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) //VERIFICA SE O C
```

```
return; //RETORNA PARA LER NOVAMENTE

/* INICIO BLOCO DE CÓDIGO RESPONSÁVEL POR GERAR A TAG RFID LIDA */
String strID = "";
for (byte i = 0; i < 4; i++) {
  strID +=
  (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
  String(rfid.uid.uidByte[i], HEX) +
  (i != 3 ? ":" : "");
}
strID.toUpperCase();
/* FIM DO BLOCO DE CÓDIGO RESPONSÁVEL POR GERAR A TAG RFID LIDA */

int nome = -1; // Inicializa com "Não cadastrado"

// Verifica o ID do cartão e associa o nome correspondente
if (strID == "33:42:4A:FE") {
  nome = 1;
} else if (strID == "FD:69:C7:24") {
  nome = 2;
} else if (strID == "D3:39:ED:FE") {
  nome = 3;
} else if (strID == "8D:D9:9B:20") {
  nome = 4;
} else if (strID == "B1:95:58:0D") {
  nome = 5;
} else if (strID == "E3:DC:D1:A6") {
  nome = 6;
} else if (strID == "A3:32:25:03") {
  nome = 7;
  chave7 = !chave7;
}

//if (nome == 7)
  //chave7 = !chave7;

// Imprime o nome correspondente ao cartão lido
//Serial.println("Nome ID");
//Serial.println(nome);
```

```
tab_reg[4] = nome;

rfid.PICC_HaltA(); //PARADA DA LEITURA DO CARTÃO
rfid.PCD_StopCrypto1(); //PARADA DA CRIPTOGRAFIA NO PCD

/* -----

//Atualizando o Modbus
modbusino_slave.loop(tab_reg, 20);
}
```