

AUTOMAÇÃO E SUPERVISÃO DA OXIGENAÇÃO DO PACIENTE

Douglas Ferreira Gomes, Eduardo Rodrigo de Souza, Elmir Jose Barbosa Ferraz, Robson Pereira de Almeida, Prof. Dr. Wellington Batista de Sousa

Faculdade de Tecnologia de São Bernardo do Campo

douglas.gomes8, eduardo.souza79, elmir.ferraz, robson.almeida21, wellington.sousa5@fatec.sp.gov.br

RESUMO: Este projeto que se intitula Automação e Supervisão da Oxigenação do Paciente corresponde à pesquisa de estudo de caso referente à automação aplicada na medicação e na supervisão durante o processo de medicação com oxigênio suplementar. No início de 2024, foi observado na região da Grande São Paulo que não havia disponibilidade de um sistema que pudesse iniciar e alterar a quantidade de suplementação de oxigênio de forma eletrônica e remota. Além disso, também foi observado que não havia uma forma de registro que pudesse demonstrar efetivamente se ocorreu à devida atuação do profissional de saúde. O principal objetivo foi o desenvolvimento de um dispositivo protótipo para supervisionar os dados de oxigenação do paciente e notificar os responsáveis pelo ajuste em caso de variações relevantes desta necessidade. Ao final do projeto, é possível identificar a evolução do dispositivo protótipo construído, atendendo todas as tarefas e as atividades que foram propostas. É observado o avanço tecnológico aplicado, porém é importante destacar a necessidade de diferentes testes de validação fora do ambiente didático e com diversas durações antes que um dispositivo similar seja o principal responsável no processo em questão. Por isso, este projeto contribui para trazer a atenção ao que pode ser desenvolvido e evoluído a partir do mesmo.

Palavras-chave: Arduino. Oxigenação Suplementar. Oxímetro. SCADA. Saturação de Oxigênio.

ABSTRACT: This project, titled "Automation and Supervision of Patient Oxygenation", corresponds to case study research focused on automating medication and supervision during the medication related to supplemental oxygen. In early 2024, it was noted that there was no available system in the Grande São Paulo region that could initiate and adjust the amount of oxygen supplementation electronically and remotely. Furthermore, it was also noted that no

form of recording could effectively demonstrate whether health professionals had acted appropriately in this situation. The primary objective of this project was to develop a prototype device that could monitor the patient's oxygenation data and alert responsible parties when adjustments were necessary due to significant variations. By the end of the project, it is possible to identify the progress made with the prototype device, successfully meeting all tasks and activities proposed. The technological advancement achieved is noted, however, it is essential to emphasize the need for different validation tests outside of the educational environment and over different extension periods before deploying a similar device as the primary solution for this process. Therefore, this project contributes to bringing attention to what can be developed and evolved from it.

Keywords: Arduino. Supplemental Oxygen. Oximeter. SCADA. Oxygen Saturation.

1. Introdução

O hospital é um organizador de caráter médico-social, que deve garantir assistência médica, tanto curativa como preventiva, para a população, além de ser um centro de medicina e pesquisa. Existem hospitais de organizações públicas e de instituições privadas, porém há momentos históricos quando todo o sistema de saúde não conseguiu atender a demanda populacional, gerando um colapso nos serviços de saúde. O mais recente evento deste tipo foi a pandemia global causada pelo vírus Sars-Cov-2, popularmente chamado de covid-19 quando houve escassez de respiradores e de cilindros de oxigênio necessários para aplicação de oxigenoterapia e ventilação mecânica dos pacientes, e houve também sobrecarga de trabalho para os profissionais da área da saúde.

Das pontuações mencionadas, o objetivo deste trabalho que se intitula Automação e Supervisão da Oxigenação do Paciente é desenvolver um dispositivo para supervisionar os dados de oxigenação do paciente e notificar os responsáveis pelo ajuste em caso de variações relevantes desta necessidade, validando o funcionamento deste protótipo no ambiente acadêmico. A administração autônoma do oxigênio ao paciente não foi implementada ao escopo porque o oxigênio hospitalar é um medicamento e, portanto, ele deve ser prescrito pelos profissionais responsáveis.

A temática justifica-se para aliviar a carga de trabalho dos profissionais da área de saúde e melhorar o controle da aplicação de oxigenação suplementar, possibilitando que mais pessoas possam ser atendidas simultaneamente. Para desenvolvimento do trabalho, os objetivos específicos envolvem pesquisa bibliográfica atualizada sobre oxigenação

suplementar, oxímetro, microcontroladores, sistemas supervisórios, sistemas de controle, válvulas, e linguagens de programação, para construção do protótipo funcional aplicando os conceitos adquiridos.

2. Fundamentação teórica

Para a realização deste projeto científico foi necessário pesquisar sobre a teoria de autores renomados, que dão sustentação ao desenvolvimento e a construção do protótipo. Essas teorias estão relacionadas à oxigenação suplementar, emprego de dispositivos como oxímetro, microcontroladores, sistemas supervisórios, sistemas de controle, válvulas e a aplicação das linguagens de programação.

2.1. Oxigenação suplementar

SILVA (2020) destaca que a oxigenoterapia consiste na administração de oxigênio suplementar acima da concentração do ar ambiente que é de aproximadamente 21%, para garantir a oxigenação dos tecidos. Ela corrige a hipoxemia, que é o baixo nível de oxigênio no sangue e, conseqüentemente, diminui a sobrecarga de trabalho do sistema cardiorrespiratório, mediante a elevação do nível de oxigênio alveolar e sanguíneo.

A causa mais frequente para a utilização da oxigenoterapia é a insuficiência respiratória aguda (IRpA), em que o sistema respiratório não consegue manter os valores da pressão arterial de oxigênio (PaO₂) e da pressão arterial de gás carbônico (PaCO₂).

LISSA (2022) afirma que a oxigenação suplementar é um recurso que deve ser aplicado de forma correta e individualizada. Houve uma atualização relevante na forma de prescrição de oxigênio, e que foi ainda mais estudada após a pandemia da covid-19, pois a utilização indevida da suplementação de oxigênio pode ser extremamente danosa. Tanto a hipóxia (falta de oxigenação) quanto à hiperóxia (excesso de oxigenação) podem trazer prejuízos à vida.

Segundo SALERNO (2023), o oxigênio necessário para a oxigenoterapia pode ser disponibilizado de três formas distintas:

- **Cilindros de oxigênio:** são armazenadores de oxigênio e necessitam de recarga sempre que o manômetro indicar que a pressão de oxigênio está baixa;

- **Concentrador:** são equipamentos que filtram impurezas e buscam enriquecer o seu recipiente com o oxigênio (O₂) do ar, gerando oxigênio concentrado e pressurizado (devido à compressão). São utilizados para suplementar até 5 L/min. e tem como principais limitantes a dificuldade de locomoção e a necessidade de uma fonte elétrica de energia;

• **Equipamento de Oxigênio Líquido:** esse equipamento central comprime e resfria o ar ambiente a uma temperatura de aproximadamente -200°C , onde ele está na fase líquida. Em sequência, o equipamento aquece o ar líquido até uma temperatura inferior a -183°C . Nessa etapa, grande parte dos outros componentes do ar já retornou a fase gasosa, restando O_2 praticamente puro na fase líquida. Para administração de Oxigênio ao paciente, o equipamento retorna adequadamente o Oxigênio à temperatura adequada para a aplicação. As principais características desta fonte são a mobilidade e a autonomia de um período sem energia elétrica. Isto é possível porque o equipamento central reabastece o reservatório de um equipamento portátil que é recarregável.

Igualmente, SALERNO (2023) descreve que essas fontes de oxigênio disponíveis podem ser administradas no paciente de três formas:

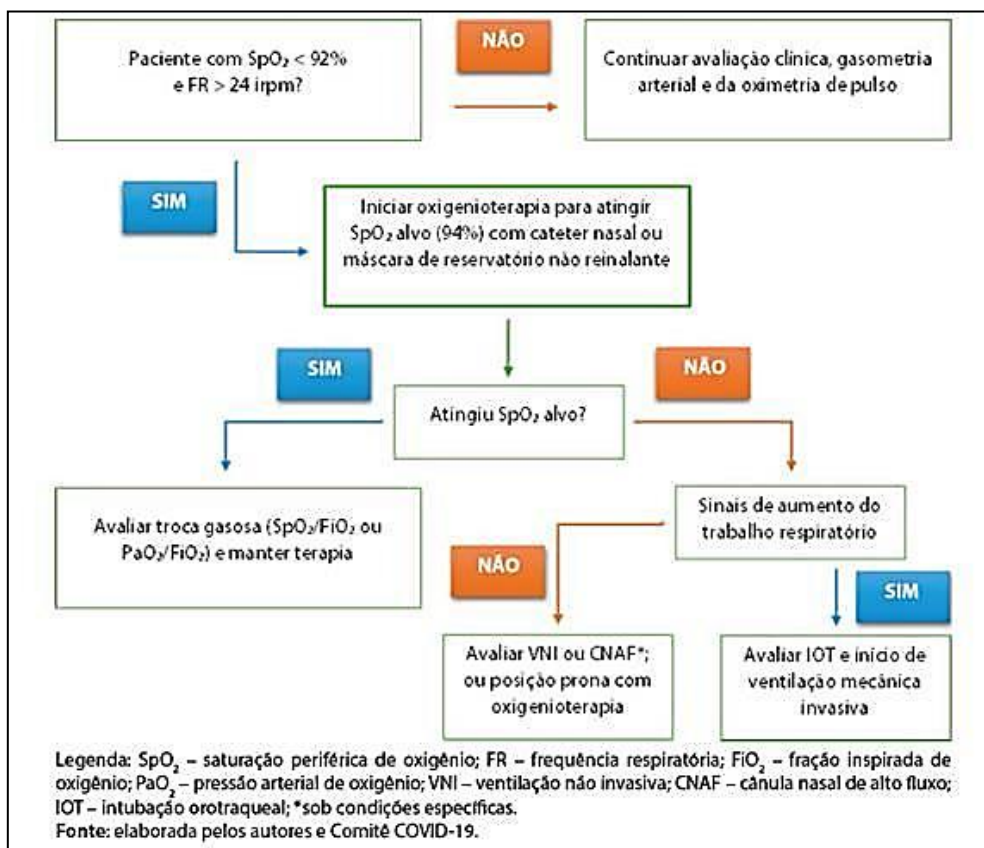
• **Cateter ou Cânula Nasal:** a cânula nasal é o dispositivo mais utilizado, tanto pela disponibilidade quanto pela facilidade do uso. Ela é um dispositivo simples, de baixo fluxo, com fornecimento de oxigênio de 0,5 a 5L/min. e não há a necessidade de instalar umidificação (recipiente de água) no sistema, pois não há evidências de benefícios clinicamente significantes, e aumenta os riscos de infecção. A aplicação de fluxos altos pode levar a ressecamento da mucosa nasal ou até lesões na mucosa;

• **Máscara Simples:** a máscara simples pode aumentar a Fração Inspirada de Oxigênio (FiO_2) em até 60%. Ela deve ser usada com um fluxo mínimo de 5 L/min. para prevenir retenção de dióxido de carbono (CO_2). Tem a vantagem de ser mais acessível e leve (podendo ser utilizada em casa), porém não tem garantia de selamento, além de precisar ser removida se o paciente precisar falar ou se alimentar;

• **Máscara de Venturi:** a máscara de Venturi é composta por um sistema de válvulas acopladas à máscara facial que possibilita a oferta de diferentes Frações Inspiradas de Oxigênio (FiO_2). Na região inferior de cada válvula, encontram-se a porcentagem oferecida e os litros necessários para alcançar a FiO_2 desejada. Os valores de porcentagem mais comuns são 24%, 28%, 31%, 35%, 40% e 50%. É um sistema de alto fluxo. Quanto maior o orifício de entrada, maior a concentração de oxigênio.

A seguir, a Figura 2.1 ilustra o protocolo de oxigenoterapia desenvolvido pelo Comitê covid-19 no ano de 2020 para padronização e divulgação das melhores práticas até aquele momento. Na figura é possível observar muitos dos conceitos abordados até aqui.

Figura 2.1 - Protocolo de oxigenoterapia covid-19



Fonte: SILVA, 2020, p.12

2.2. Oxímetro

SILVA (2020) indica que na aplicação de oxigenação suplementar são necessários a avaliação e o acompanhamento dos profissionais em relação ao Transporte Sistêmico de Oxigênio (TO₂) no paciente. Esse indicador pode ser verificado principalmente por meio de dois exames:

- **Gasometria arterial**, na qual obtemos a saturação arterial de oxigênio (SaO₂), que em indivíduos saudáveis varia de 96 a 98%;
- **Oximetria de pulso**, método não invasivo que por meio de um feixe de luz consegue medir a Saturação Periférica de Oxigênio (SpO₂), a qual geralmente tem uma variação de 2 a 3% em relação à SaO₂.

AMARAL (1992) destaca que entre as técnicas não invasivas também é possível a utilização da Capnografia (estudo gráfico e de monitoração do gás carbônico CO₂ ao final da expiração [ETCO₂], uma vez que o CO₂ presente nos alvéolos pulmonares é proporcional ao débito cardíaco [fluxo sanguíneo em litros produzido pelo coração a cada batimento] e ao

fluxo sanguíneo pulmonar). Entre as técnicas invasivas, também é possível a utilização da oximetria sanguínea.

AMARAL (1992) continua explicando que a leitura da SpO₂ através do oxímetro de pulso acontece porque uma fonte de luz em uma das extremidades da estrutura do oxímetro emite e, esta luz emitida atravessa os tecidos sendo parcialmente absorvida. Através de um sensor (detector de luminosidade) situado na superfície oposta da extremidade, o oxímetro analisa a absorção de luz.

O mesmo autor indica que por volta de 1975, os oxímetros passaram a utilizar apenas dois comprimentos de onda, a luz vermelha e a luz infravermelha (940 e 660 nm, respectivamente) e assim, os aparelhos disponíveis até então tornaram-se mais precisos, apresentando precisão de cerca de 95% nas leituras quando a SpO₂ é maior que 70%.

A luz vermelha atravessa facilmente a desoxihemoglobina (hemoglobina [proteína dos glóbulos vermelhos do sangue] não ligada com o oxigênio), enquanto é absorvida pela oxihemoglobina (hemoglobina ligada com oxigênio). Assim, quanto mais luz for absorvida pelos tecidos durante a verificação, maior será o nível de oxigênio no sangue e, por consequência, maior o nível de SpO₂. Os oxímetros realizam essa verificação indireta e utilizam equações que levam em consideração a pulsação, resultando na leitura indicada.

AMARAL (1992) indica ainda que outras regiões do corpo possam ser alternativamente utilizadas para a verificação da oximetria de pulso como o septo nasal, artelhos e bochechas. No caso dos recém-nascidos, a monitoração pode ser realizada na região palmar e no dorso da mão ou no braço.

2.3 Microcontroladores

Os microcontroladores surgiram na década de 70, quando os sistemas embarcados começaram a se desenvolver, sendo utilizados onde antes a tarefa era executada de forma manual. Segundo KERSCHBAUMER (2018), microcontroladores são circuitos integrados que possuem em seu interior todos os componentes necessários ao seu funcionamento dependendo unicamente da fonte de alimentação externa. Pode-se dizer que os microcontroladores são computadores de um único chip.

Esse autor explica que um sistema microprocessado é composto por uma unidade central de processamento CPU (do inglês *Central Processor Unit*) e um conjunto de periféricos necessários ao seu funcionamento. Dentre estes periféricos destacam-se a memória de dados, a memória de programa e o circuito de *clock* (sinal de pulso responsável pela sincronização das operações). Os microcontroladores diferem dos sistemas tradicionais principalmente por já integrarem os seus periféricos dentro do próprio componente.

KERSCHBAUMER (2018) ressalta que esta integração é uma das principais vantagens dos microcontroladores, pois ao conter os periféricos em um mesmo componente, a sua utilização é mais fácil e mais barata. Assim sendo, os sistemas microcontrolados não necessitam de muitos componentes externos, o que também torna mais simples a construção de placas de circuito e diminui o custo de componentes e de produção.

KERSCHBAUMER (2018) indica que os microcontroladores são utilizados em praticamente todos os dispositivos eletrônicos digitais que nos cercam, como por exemplo, centrais de alarme, teclados do computador, monitores, discos rígidos de computador, relógio de pulso, máquinas de lavar, forno de micro-ondas, telefones, rádios, televisores, automóveis, aviões, impressoras, marca passos, calculadores, entre outros. Além disso, os microcontroladores também são muito utilizados na indústria como, por exemplo, nos controladores de processos, sensores inteligentes, inversores, soft starters, interfaces homem máquina (IHMs), controladores lógicos programáveis (CLPs), balanças, indicadores digitais, e outras aplicações.

Para que se ter ideia da importância dos microcontroladores, KERSCHBAUMER (2018) elenca uma lista considerável com os principais fabricantes encontrados no mercado atualmente. São eles: AMCC, ATmel, Cypress MicroSystems, Expressif, Freescale Semiconductor, Fujitsu, Holtek, Intel, Microchip Technology, National Semiconductor, NXP (antiga Philips Semiconductors), NEC, Parallax, Inc., Renesas Tech. Corporation, STMicroelectronics, Silicon Laboratories, Texas Instruments, Western Design Center e ZiLOG. KERSCHBAUMER (2018) ainda salienta que estes são apenas os fabricantes, e que cada um destes possui diversas famílias de microcontroladores, e em cada família há dezenas de componentes diferentes.

Neste contexto, o ATmega328P é um chip microcontrolador desenvolvido pela ATmel e utilizado na placa microcontrolada Arduino Uno e o ESP32 é um chip microcontrolador SoC (*System on a Chip*) desenvolvido pela Expressif utilizado na placa microcontrolada de mesmo nome.

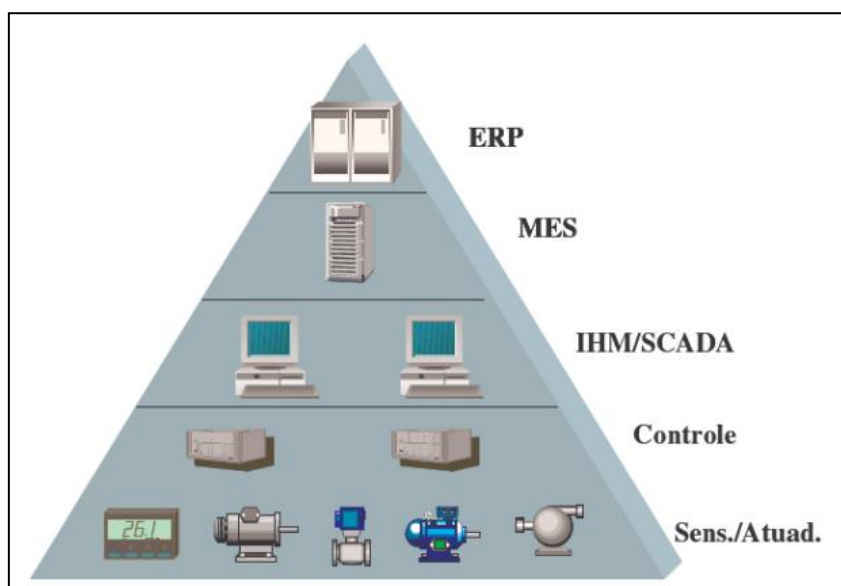
2.4 Sistemas Supervisórios

Segundo COELHO (2010), os sistemas supervisórios, também chamados de SCADA (*Supervisory Control and Data Acquisition*), são sistemas que permitem monitorar e rastrear informação de um processo produtivo ou instalação física. Esta informação é verificada em campo (no processo ou instalação) através de equipamentos de aquisição de dados (comumente chamados de sensores e transdutores) e, em seguida, a informação é armazenada e tratada para disponibilização ao usuário. Através de lógicas programáveis e do acionamento

de comandos pelo usuário, a informação também pode ser enviada novamente para o campo mudando os estados de equipamentos de ação (comumente chamados de atuadores e motores).

COELHO (2010) descreve que os primeiros sistemas SCADA eram basicamente telemétricos e possibilitavam apenas informar periodicamente o estado contínuo do processo industrial ou da instalação física, monitorando sinais representativos de medidas e estados (dos sensores, transdutores, atuadores e motores) através de painéis de lâmpadas e displays indicadores. Com a evolução tecnológica em relação aos *hardwares*, a utilização da rede de *Internet*, a evolução das Tecnologias de Automação (TA) e com o desenvolvimento das Tecnologias da Informação (TI), além da função inicial, os sistemas SCADA passaram a ter um papel essencial na interação entre a 3ª e a 4ª camada na Pirâmide de Automação, conforme ilustrado na Figura 2.2).

Figura 2.2 - Pirâmide de Automação¹



Fonte: COELHO, 2010, p.4

Assim, COELHO (2010) destaca que atualmente, os sistemas SCADA têm a capacidade de utilizar tecnologias de computação e comunicação para automatizar tarefas de monitoração e controle, utilizando recursos gráficos elaborados e conteúdo multimídia para apresentação intuitiva ao operador, identificando todas as variáveis numéricas e alfanuméricas

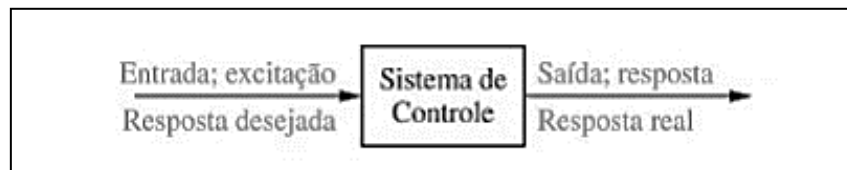
¹ Legenda: sensores e atuadores estão representados na 1ª camada, referente ao campo (processo ou instalação). Os equipamentos que controlam a 1ª camada assim como os CLPs estão representados na 2ª camada, referente ao controle. As IHM e os sistemas SCADA estão representados na 3ª camada referente à Supervisão e ao Monitoramento. A pirâmide representa as Tecnologias de Automação da 1ª até a 3ª camada e as Tecnologias da Informação nas camadas superiores. A 4ª camada refere-se ao gerenciamento do planejamento da planta e a 5ª camada refere-se ao gerenciamento estratégico corporativo.

envolvidas no campo através de Tags, gravando registros históricos da informação desejada em bancos de dados (no servidor local ou na nuvem), emitindo relatórios da informação desejada, enviando os sinais de comando do operador, executando funções lógicas computacionais que podem alterar os sinais das saídas (comumente os atuadores e motores) em função dos resultados, verificando e notificando alarmes de acordo com limites configuráveis através de emissão sonora, mensagens de erro e mudança de cor dos objetos na tela.

2.5 Sistemas de controle

Segundo NISE (2013), um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de se obter uma saída desejada com um desempenho desejado, dada uma entrada especificada. A Figura 2.3 mostra um sistema de controle em sua forma mais simples, na qual a entrada representa uma saída desejada.

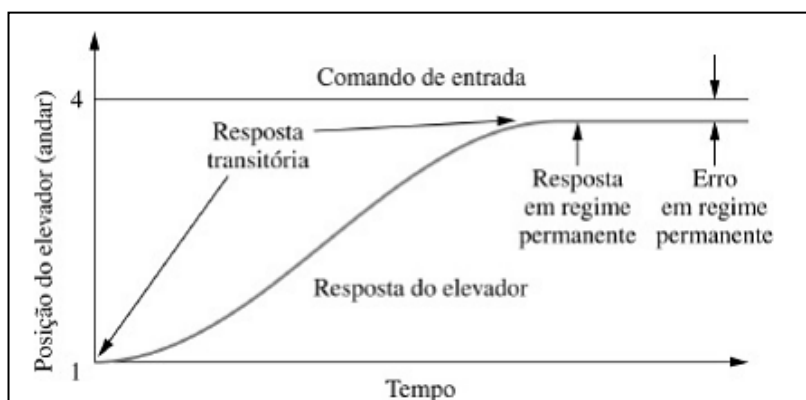
Figura 2.3 - Sistema de controle simplificado



Fonte: NISE, 2013, p.28

Considerando por exemplo um elevador, quando o botão do quarto andar é pressionado no primeiro andar, o elevador sobe até o quarto andar com uma velocidade e uma exatidão de nivelamento projetadas para o conforto do passageiro. Pressionar o botão do quarto andar é uma entrada que representa a saída desejada, ilustrada como uma função degrau na Figura 2.4 O desempenho do elevador pode ser verificado a partir da curva de resposta do elevador.

Figura 2.4 - Exemplo do acionamento de um botão de comando do elevador e da sua curva de resposta



Fonte: NISE, 2013, p.29

NISE (2013) elenca que duas das principais medidas de desempenho são evidentes: (1) a resposta transitória e (2) o erro em regime permanente. Neste exemplo, o conforto e a paciência do passageiro dependem da resposta transitória. Se esta resposta for muito rápida, o conforto do passageiro é sacrificado; se for muito lenta, a paciência do passageiro é sacrificada. O erro em regime permanente é outra especificação de desempenho importante, uma vez que a segurança do passageiro e a conveniência podem ser sacrificadas se o elevador não nivelar apropriadamente no patamar dos andares.

O mesmo autor ainda destaca a importância dos sistemas de controle, exemplificando que com a utilização dos sistemas de controle é possível mover equipamentos pesados com uma precisão que, de outra forma, seria impossível. Também é possível apontar grandes antenas para os confins do universo buscando captar sinais de rádio muito fracos. Controlar estas antenas manualmente seria impossível. Por causa dos sistemas de controle, os elevadores transportam seus passageiros rapidamente ao destino, parando automaticamente no andar correto. Sem sistemas de controle não é possível fornecer a potência necessária para a carga e a velocidade uma vez que motores fornecem a potência, e sistemas de controle regulam a posição e a velocidade.

2.6 Válvulas

MATHIAS (2024) define que as válvulas são componentes de tubulações utilizados para bloquear, direcionar, limitar a pressão de entrada ou controlar a pressão de um fluido na saída de algum equipamento ou permitir o escoamento do fluido num único sentido. São dispositivos mecanicamente operados, sendo utilizados sob diversas condições operacionais.

O autor elenca que dentro de um processo industrial seja ele químico, farmacêutico,

petroquímico, papel e celulose, siderúrgico etc., sempre existem situações na qual o controle do fluido é necessário para a conclusão do produto. Esse controle pode ser feito sobre a pressão, a temperatura, a vazão, o pH, o volume, o nível etc. num determinado fluido para que o produto seja produzido nas condições de qualidade desejadas.

Ainda segundo o autor, o controle deve ser feito através de uma válvula corretamente dimensionada e selecionada para cada aplicação em particular, para que não fiquem elevados os custos com sua instalação e manutenção. Sendo assim, em princípio, a correta seleção de uma válvula depende especificamente da aplicação e de acordo com as reais necessidades do processo.

MATHIAS (2024) indica que as válvulas podem ser classificadas quanto à:

- ***Função***
- ***Classe de pressão***
- ***Material construtivo***
- ***Corpo***
- ***Castelo (sistema de acionamento)***
- ***Disponibilidade de By-pass***
- ***Conexões (com o sistema de tubulação)***
- ***Movimentação***

Além destas classificações citadas acima, existe também a classificação comercial segundo a utilização industrial. Com relação à classificação comercial, MATHIAS (2024) descreve que os tipos de válvula mais comuns são as válvulas gaveta, globo, retenção, guilhotina, diafragma, esfera, borboleta e macho.

2.7 Linguagens de programação

Segundo PEREIRA (2013), a programação consiste na tradução do pensamento lógico necessário para o cumprimento de determinada tarefa em uma sequência de comandos que podem ser interpretados e executados por uma máquina, e a linguagem consiste no conjunto de regras ou codificação para que o programador desenvolva os comandos e a máquina execute adequadamente o que foi solicitado.

O mesmo autor descreve que os primeiros dispositivos programáveis tinham seus programas escritos com códigos chamados códigos de máquinas, que consistiam basicamente em códigos binários inseridos por meio de um dispositivo de entrada para então serem

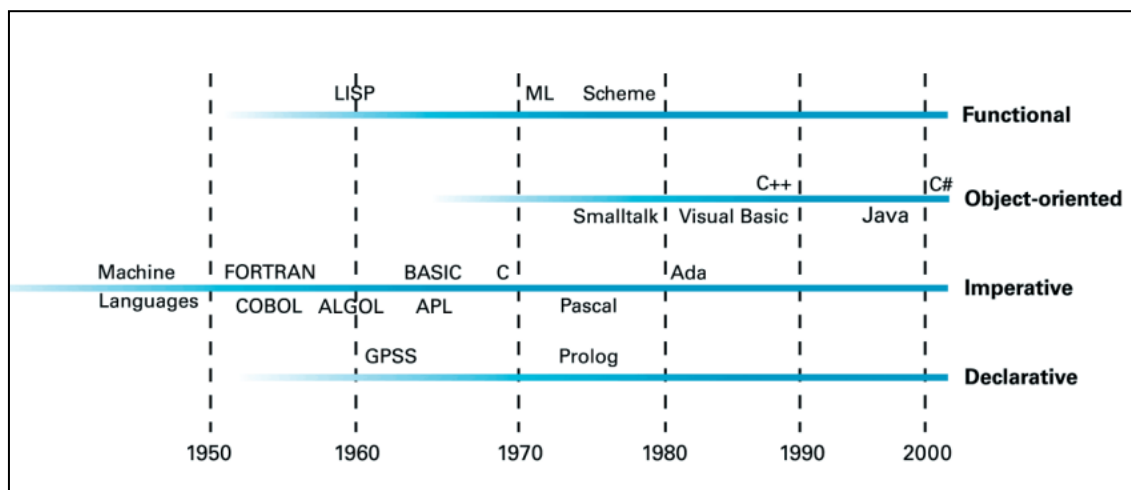
executados pela máquina.

Com o passar do tempo e a necessidade crescente da programação de sistemas, surgiram outras linguagens, e entre elas a *Assembly*, que consiste em uma forma alternativa da representação dos códigos utilizando abreviações dos termos usuais que descrevem a operação executada pela máquina.

O autor ainda destaca que o *Assembly* já era superior aos códigos de máquina, porém ainda se tratava de uma linguagem de baixo nível (linguagem que só possui instruções e funções que estão definidas no processador em utilização). O problema até então era a extrema dificuldade de inserir rotinas ou operações que não tinham função e instrução definidas no processador.

Devido a esta problemática, PEREIRA (2013) elenca que surgiram as linguagens de alto nível, sendo a linguagem FORTRAN (*FORmula TRANslator*) a primeira destas que foi mais aceita pelos programadores. Seguida da linguagem FORTRAN vieram as linguagens COBOL (*COmmon Business Oriented Language*), ALGOL (*ALGOrithmic Language*) e tantas outras ao decorrer do tempo. A criação das linguagens C e Pascal no início da década de 70, segundo o mesmo autor, ressalva que ambas as linguagens são originadas da linguagem ALGOL. Na Figura 2.5, está ilustrada grande parte das linguagens de programação em ordem cronológica da criação.

Figura 2.5 - Linha do tempo com as linguagens de programação



Fonte: BRAGA, 2006, p.17

Segundo PEREIRA (2013), atualmente a maior parte dos microcontroladores utiliza a linguagem C para o desenvolvimento do *software* e programação, por ter uma grande velocidade na criação do projeto, já que esta é uma linguagem considerada fácil e acessível ao

desenvolvimento.

Além disso, para se programar as placas microcontroladas Arduino Uno e ESP32, é comumente utilizada a plataforma IDE (*Integrated Development Environment*) disponibilizada pela plataforma Arduino. Segundo autores como ARAUJO (2019), este ambiente possui programação muito parecida com a linguagem C/C++.

3. Metodologia

Ao deparar-se com a necessidade da oxigenação suplementar em pessoas próximas e através da conversa ocasional com profissionais da área da saúde pública e privada na região da Grande São Paulo no início do ano de 2024, foi observado que não havia disponibilidade de um sistema que pudesse iniciar e alterar a quantidade de suplementação de oxigênio de forma eletrônica e remota. Além disso, também foi observado que não havia uma forma de registro que pudesse demonstrar efetivamente se ocorreu à devida atuação do profissional de saúde.

Neste cenário, foi identificada a oportunidade e então foi proposto o tema Automação e Supervisão da Oxigenação do Paciente, com o objetivo principal de desenvolver um dispositivo para supervisionar os dados de oxigenação do paciente, notificando os responsáveis pelo ajuste em caso de variações relevantes desta necessidade. Como objetivo secundário, também foi proposto gerar um banco de dados com registro histórico das leituras e das alterações realizadas pelos funcionários no decorrer do tempo.

Para o desenvolvimento deste projeto de estudo de caso, a partir da definição da temática, dos objetivos e do conhecimento complementar adquirido na pesquisa da seção de Fundamentação Teórica, foi elencado o seguinte planejamento a ser executado e concluído durante o ano de 2024 nas dependências físicas da FATEC Adib Moisés Dib, em São Bernardo do Campo:

- **1ª Etapa:** seleção dos equipamentos
- **2ª Etapa:** disponibilização dos equipamentos selecionados
- **3ª Etapa:** montagem e programação com o oxímetro
- **4ª Etapa:** programação para modular a saída da válvula
- **5ª Etapa:** montagem das telas e do controle no sistema supervisório
- **6ª Etapa:** conexão do supervisório com a placa microcontrolada
- **7ª Etapa:** finalização gráfica do supervisório
- **8ª Etapa:** montagem final e teste funcional

4. Desenvolvimento do projeto

Na primeira etapa, em acordo com a Fundamentação Teórica, foi definida a verificação da saturação através da oximetria de pulso utilizando o oxímetro de dedo porque este é um método menos evasivo ao paciente e, ao mesmo tempo, é um método comumente utilizado no ambiente hospitalar. Porém, devido à um desafio ocorrido e explicado futuramente nos resultados, foi definido seguir com o módulo sensor de batimento cardíaco e oxímetro MAX30100, módulo adequado as placas microcontroladas Arduino Uno e ESP32.

De igual modo, foi definida a utilização da placa microcontrolada ESP32 porque é uma placa microcontrolada mais robusta, com maior capacidade de processamento e que permite a conexão via *Bluetooth* e *WiFi*, foi definida a utilização do *software* SCADA da Elipse E3, por ser um *software* que permite maior quantidade de Tags e tem interface mais intuitiva ao programador, e foi definida a utilização de uma válvula do tipo borboleta impressa em 3D e acionada por servo motor devido à simplicidade de montagem para aplicação, buscando aproveitar o que já havia sido adquirido durante o curso para redução de custo, no caso, o servo motor.

Na segunda etapa, foram pesquisados os componentes conforme as definições da primeira etapa e foram feitas as aquisições necessárias através da Internet. Primeiramente, foi feita a aquisição da placa microcontrolada ESP32. Na sequência, foi feita a aquisição de duas unidades do módulo sensor MAX30100. Durante as buscas para estes componentes, devido à condição explicada nos resultados, foi definido realizar a aquisição de uma válvula eletrônica comercial de controle proporcional de pressão de 0,05 a 5 bar atuada por tensão de 0 a 10 V em estado já usado, porém em nível adequado de conservação ao invés de seguir com a impressão 3D da válvula.

Na terceira etapa, foi feita a montagem do módulo MAX30100 em placa de ensaio, a programação e o teste de funcionamento deste módulo. Este módulo funciona com alimentação de tensão de 3,3 V a 5 V e transmitindo os dados através do protocolo de comunicação serial I2C (*Inter-Integrated Circuit*), utilizando as conexões SCL e SDA da placa microcontrolada. Devido à situação observada e descrita nos resultados, foi definido prosseguir as etapas de programação utilizando o Arduino Uno e retornar para a adequação ao ESP32 na sexta etapa (conexão do supervisor e da placa microcontrolada).

Durante a nova montagem e os primeiros testes no Arduino Uno, devido à evento observado e descrito nos resultados, foi acrescentada a montagem de 3 resistores de 10k Ω cada na placa de ensaio para *pull up* nas conexões SDA, SCL e INT, conforme uma das

recomendações disponíveis do fabricante do módulo MAX30100. A programação também foi escrita e editada com base nas recomendações do fabricante do módulo MAX30100.

Na quarta etapa, devido às situações descritas nos resultados, foi necessário disponibilizar uma fonte de alimentação 24 V dedicada para a válvula, conexões de engate rápido pneumático para mangueira de diâmetro de 6 mm e alguns metros desta mangueira. Entretanto, devido ao escopo do nosso projeto, foi necessário principalmente realizar uma correlação entre a vazão de oxigênio na suplementação, monitorada em L/min, e a pressão de controle da válvula, medida e indicada em bar.

Devido ao ambiente hospitalar utilizar o fluxômetro de oxigênio para regulação e monitoração da vazão do oxigênio suplementar, foi definido realizar a aquisição de um fluxômetro hospitalar para realizar a correlação de um entre os possíveis cenários através do método prático, consistindo em alterar o valor do sinal que sai da placa microcontrolada para atuação da válvula em PWM (*Pulse Width Modulation*), anotando a tensão equivalente gerada pelo PWM da placa microcontrolada em V, o valor indicado no visor da válvula em bar e o valor de vazão indicado no fluxômetro em L/min.

Uma vez que os componentes estavam sendo disponibilizados, seria necessária a utilização de ar comprimido. Em acordo com os equipamentos disponíveis na FATEC São Bernardo do Campo e com o corpo docente, foi disponibilizado ao nosso grupo um compressor portátil para o desenvolvimento e apresentação do projeto.

Como o comando da válvula funciona com sinal positivo de 0 a 10 V enquanto as placas microcontroladas enviam sinal de tensão através de PWM, de até 5V, foi definido realizar este teste sem amplificar o sinal de comando do Arduino Uno para análise do resultado, uma vez que a teoria sobre a válvula de tensão proporcional indica que ela mantém o funcionamento dentro da faixa de tensão de atuação de 0 a 5V na zona de trabalho de abertura e fechamento equivalente.

Para finalizar as considerações necessárias no teste da válvula e no levantamento dos dados de correlação, foi definida a pressão do compressor portátil em 3 bar e foi definido um pré-ajuste que ficou inalterado no fluxômetro.

Na quinta etapa, foi definida cada informação que deveria conter no sistema supervisor e em quais telas cada uma delas deveria ser distribuída. Desta análise e definição, foram elencadas as telas Menu, Home, Monitor, Paciente, Usuários, Histórico e Relatório.

O Menu foi definido como uma tela de acesso rápido das outras telas e demonstração de informações do estado atual da operação como, por exemplo, indicar a data e a hora atual, o nome do paciente em atendimento e o usuário que está responsável pelo paciente. Esta tela ficou fixa na parte superior da tela, independente da tela inferior que estiver sendo acessada.

O Home foi definido como a tela inicial, aquela que fica visível quando o programa é carregado inicialmente e que contém os créditos do projeto. O monitor foi definido como a tela que deverá mostrar as informações gerais do paciente de forma gráfica, e também deverá mostrar a mensagem de alerta. A tela do Paciente foi definida como aquela que deve ter o nome do paciente inserido, demonstra a informação lida do módulo MAX30100, e também deverá conter os botões de controle da válvula.

A tela de Usuários foi definida para configurar o nível de acesso, as permissões e para que a gerência possa modificar essa configuração, incluindo funcionários novos e removendo o acesso de usuários antigos. Foram definidos inicialmente quatro usuários: ADMIN, MEDICO, ENFERMAGEM e GERENCIA. O usuário ADMIN é o desenvolvedor e tem todo controle sobre o projeto. Os usuários MEDICO e ENFERMAGEM conseguem visualizar as telas, fazer o controle de saturação de paciente, emitir relatórios, e visualizar o histórico. O usuário GERENCIA consegue criar o acesso de novos usuários e editar os usuários já cadastrados.

A tela Relatório foi definida para emitir relatórios em alguns formatos de arquivo, permitindo confirmação da informação desejada, a análise da informação e a impressão da informação de acordo com o nome do paciente desejado. A tela Histórico foi definida para demonstrar a coleta de dados do banco de dados local da máquina onde está funcionando o supervisor em Microsoft Access.

Grande parte das telas e dos objetos indicados foram montados neste período utilizando as ferramentas disponíveis no *software* supervisor definido, porém sem formatações de design. Conforme motivo descrito nos resultados, a alteração definitiva em relação a utilização do Arduino Uno foi definida neste mesmo momento.

Na sexta etapa, a principal tarefa foi estabelecer a conexão entre o supervisor e o Arduino Uno. Para esta tarefa, foi utilizado o conceito do protocolo Modbus RTU e seu driver instalado na biblioteca do sistema supervisor Elipse E3

No supervisor e no programa foi considerada a conexão de 04 variáveis analógicas: a leitura do batimento e da saturação, que devem ser enviadas para o supervisor, a saída do valor do botão de comando do supervisor acionado para alteração do estado da válvula e a leitura do sinal de comando que está sendo enviado para a válvula. A lógica de alteração do sinal de saída da placa microcontrolada em relação a conexão com os botões de comando do supervisor foi adotada durante os testes e, por isso está indicada nos resultados.

Deste modo foi finalizada a sexta etapa e, então, foi realizada a sétima etapa referente à finalização gráfica do supervisor, também utilizando as ferramentas disponíveis no *software* supervisor definido.

Em paralelo com a finalização da sétima etapa, foi iniciada a oitava etapa referente à montagem final e ao teste funcional, deixando a instalação mais robusta e visualmente agradável. Conforme motivos descritos nos resultados, foi comprada uma máscara de oxigênio e uma cabeça de boneca, além de uma caixa plástica de montagem elétrica, uma chave de energia liga/desliga e uma *LED* verde.

Concluindo a parte de montagem final da oitava etapa, foram realizados testes funcionais quanto ao funcionamento de todas as leituras, de todos os comandos, de todas as conexões, de todos os equipamentos e de todas as telas e objetos do supervisor buscando validar a demonstração didática.

Esse evento teste ocorreu com todas as partes individuais finalizadas sendo conectadas para o funcionamento do trabalho em sua totalidade e em nível de revisão mais atual. Com todas as conexões realizadas conforme indicado nos resultados, os integrantes do grupo verificaram se todo o *hardware* estava energizado e ligado. Em sequência, foi verificado se o supervisor estava ativo e se havia a devida conexão. Feito isso, foram realizadas leituras dos dados dos integrantes do grupo através do oxímetro MAX30100 por alguns minutos cada e repetidas vezes, gerando registros no supervisor, verificando a demonstração dos valores lidos na tela e verificando a atualização do alarme.

Além disso, foi feito *login* e *logout* de todos os usuários, foram realizadas tentativas de burlar a permissão de acesso dos usuários, foram abertas todas as telas, foram acionados todos os botões, foram preenchidos todos os *setpoints* do supervisor, foi solicitada a filtragem de todos os campos que possuem este recurso, foi acompanhada a correta demonstração do registro dos dados do paciente no banco de dados e foram gerados relatórios nos formatos habilitados para esta tarefa.

5. Testes e resultados

Na primeira etapa referente à seleção dos equipamentos conforme descrito no desenvolvimento, houve um desafio na escolha do modelo e da marca do oxímetro de dedo: identificar o protocolo de dados, a conexão dos fios ou do *Bluetooth*, e a identificação do sinal para leitura na placa microcontrolada, pois cada fabricante demonstrava sua informação proprietária. Assim, após mais tempo de pesquisa e tentativas de contato com fabricantes, foi definido seguir com o módulo sensor de batimento cardíaco e oxímetro MAX30100, módulo adequado as placas microcontroladas Arduino Uno e ESP32.

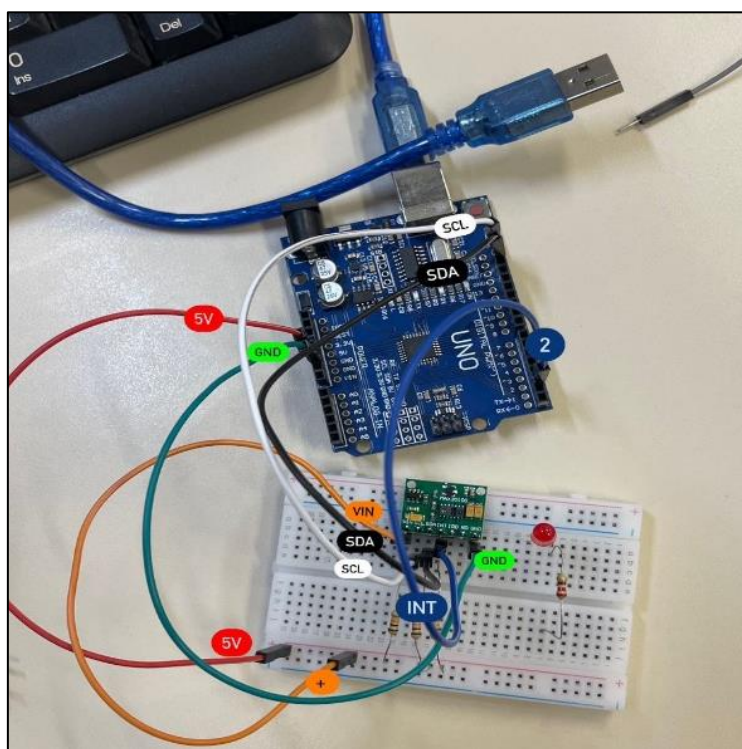
Na segunda etapa, durante as buscas para os componentes, foi observada uma ótima oportunidade de aquisição de uma válvula eletrônica comercial de controle proporcional de

pressão de 0,05 a 5 bar atuada por tensão de 0 a 10 V em estado já usado, porém em nível adequado de conservação. Assim, foi definido aproveitar a aquisição desta válvula ao invés de seguir com a impressão 3D da válvula.

Na terceira etapa, durante a programação da placa microcontrolada ESP32 foi observado que, a cada alteração no programa, a compilação e o carregamento demoravam cerca de 10 minutos. Por este motivo, foi definido prosseguir as etapas de programação com o Arduino Uno e retornar para a adequação ao ESP32 no momento da sexta etapa (conexão do supervisor e da placa microcontrolada).

Durante a nova montagem e os primeiros testes no Arduino Uno, também foi identificada uma sensibilidade do módulo MAX30100 na tensão de alimentação de 5V, ocasionando a queima desta unidade do componente. Então, foi feita a montagem de 3 resistores de 10kΩ cada na placa de ensaio para *pull up* nas conexões SDA, SCL e INT, conforme uma das recomendações disponíveis do fabricante do módulo MAX30100. A montagem está demonstrada na Figura 5.1, localizadas abaixo.

Figura 5.1 - Montagem: Módulo MAX30100 e Placa Microcontrolada Arduino Uno²



Fonte: Autoria própria, 2024

O programa desenvolvido para esta etapa está nos anexos deste artigo.

² Legenda: GND - Conexão terra. INT - Conexão de interrupção para o pulso, atuando como dreno e conectado ao pino 2 do Arduino Uno. VIN - Conexão de alimentação. SDA - Pino de dados para conexão I2C. SCL - Pino de *clock* para conexão I2C. Presentes na placa de ensaio, 3 resistores de 10 kΩ cada para *pull up* nas conexões SDA, SCL e INT.

Na quarta etapa, a principal dificuldade foi o tempo para a chegada da válvula adquirida, pois devido a um evento extraordinário, a válvula demorou mais de um mês para chegar após a aquisição. Com a chegada da válvula, foi verificada a necessidade de uma fonte de alimentação 24 V dedicada para ela, de conexões de engate rápido pneumático para mangueira de diâmetro de 6 mm e de alguns metros desta mangueira. Estes itens foram disponibilizados por integrantes do próprio grupo.

Como a medida hospitalar para a suplementação de oxigênio é a vazão em L/min e não é a pressão em bar, foi necessário realizar uma correlação entre as duas unidades. Devido ao ambiente hospitalar utilizar o fluxômetro de oxigênio para regulagem da vazão do oxigênio suplementar, ilustrado na Figura 5.2, foi definido realizar a aquisição de um fluxômetro e realizar a correlação de um ajuste no método prático.

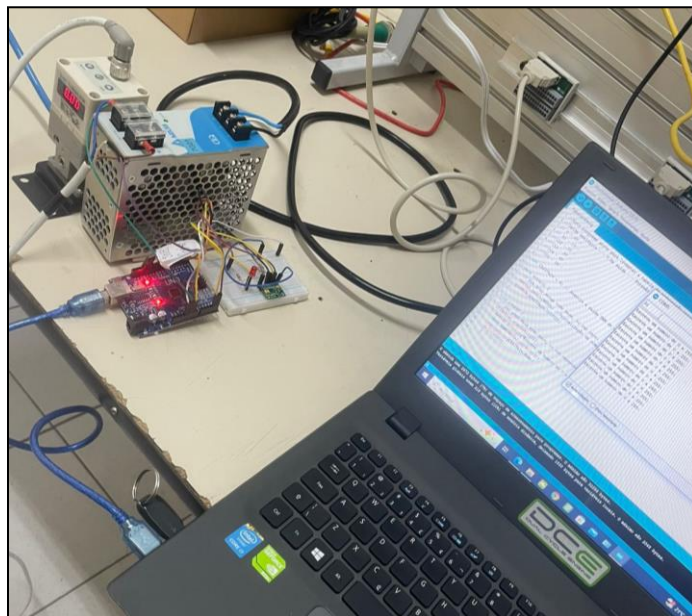
Figura 5.2 - Fluxômetro de oxigênio



Fonte: Autoria própria, 2024

Com estes componentes disponíveis em acordo com a descrição do desenvolvimento, foi realizada a montagem e o levantamento dos dados de correlação. A montagem geral desta etapa está ilustrada na Figura 5.3, demonstrada a seguir:

Figura 5.3 - Interação da fonte com o Arduino Uno e ligação pneumática na montagem da válvula³



Fonte: Autoria própria, 2024

A conexão do fluxômetro com a mangueira no diâmetro de 6 mm foi realizada conforme a Figura 5.4, localizada abaixo:

Figura 5.4 - Montagem para controle da válvula - Conexão com o fluxômetro



Fonte: Autoria própria, 2024

A montagem elétrica em relação à válvula está demonstrada na Figura 5.5, a seguir:

³ Legenda: Mangueira de diâmetro de 6 mm conectada na saída do compressor até a entrada pneumática da válvula de controle de pressão. Mangueira de diâmetro de 6 mm conectada na saída pneumática da válvula de controle de pressão até o fluxômetro.

Figura 5.5 - Detalhe da montagem elétrica em relação à válvula⁴



Fonte: Autoria própria, 2024

O programa utilizado para o teste com levantamento de dados está disponível nos anexos deste artigo.

Seguindo a informação e as considerações descritas até então, foi gerada a Tabela 5.1, demonstrada abaixo:

Tabela 5.1 - Tabela de dados de correlação levantados

Correlação entre o sinal de comando do Arduino (0 a 255 PWM) e a vazão no fluxômetro em L/min			
<i>Premissas: pressão de entrada em 3 bar e pré-regulagem da vazão do fluxômetro fixa e inalterada.</i>			
Saída Analógica Arduino Uno em PWM	Tensão equivalente em V	Pressão lida no visor da válvula de controle em bar	Vazão lida no fluxômetro
0	0	0,00	0,0
51	1	0,22	2,0
102	2	0,43	3,0
153	3	0,64	4,0
204	4	0,83	4,5
255	5	1,02	5,0

Fonte: Autoria própria, 2024

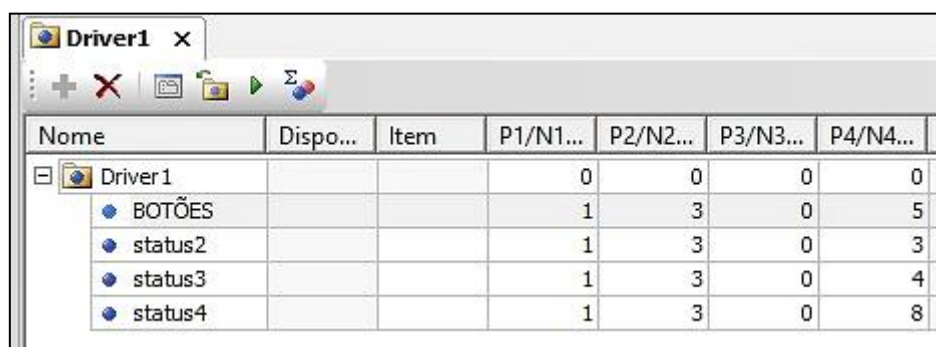
⁴ Legenda: Condutor 0V da válvula conectado no pino (-) da fonte de alimentação. Condutor 24V da válvula conectado no pino (+) da fonte de alimentação. Condutor GND da fonte de alimentação conectado no pino GND do Arduino para igualar o potencial elétrico. Condutor de sinal de comando da válvula (sinal positivo) conectado ao pino 9 do Arduino Uno.

Devido à vazão administrada no paciente no hospital durante a oxigenação suplementar ser comumente de até 5,0 L/min, os resultados obtidos nesta etapa permitem definir que a configuração da válvula e o ajuste do fluxômetro estão atendendo a necessidade da aplicação e, que a saída analógica das placas microcontroladas é suficiente para atuação e aplicação do projeto sem amplificação.

A quinta etapa e a sexta etapa ocorreram praticamente de forma paralela. Na quinta etapa, uma vez que foi possível gerar o banco de dados local em Microsoft Access, foi definido não o alterar por ser adequado ao ambiente hospitalar, ser mais seguro em relação ao vazamento externo de dados dos envolvidos e, para não perder dados dos testes realizados. Por este motivo, a alteração definitiva da placa microcontrolada do ESP32 para o Arduino se tornou viável, reduzindo o custo do projeto, aumentando a velocidade de realização dos testes, possuindo maior quantidade de informação disponível e sendo mais eficiente em relação à capacidade da placa microcontrolada que seria utilizada.

Já na sexta etapa, com grande parte das telas e dos objetos do supervisório definidos e montados previamente, porém sem formatação de design, a principal tarefa foi estabelecer a conexão entre o supervisório e o Arduino Uno. Para isto, foi utilizado o conceito do protocolo Modbus RTU e seu driver instalado na biblioteca do sistema supervisório Elipse E3. Em seguida, foi configurado o driver interno do supervisório para conectar na porta USB COM 3 da máquina, referente ao conector USB onde deve estar conectado o Arduino Uno, conforme ilustrado na Figura 5.6.

Figura 5.6 - Configuração do driver interno do Elipse E3⁵



Nome	Dispo...	Item	P1/N1...	P2/N2...	P3/N3...	P4/N4...
Driver1			0	0	0	0
• BOTÕES			1	3	0	5
• status2			1	3	0	3
• status3			1	3	0	4
• status4			1	3	0	8

Fonte: Autoria própria, 2024

Em conformidade com a descrição do desenvolvimento sobre a conexão de 04 variáveis analógicas entre o Arduino e o supervisório, o programa do Arduino Uno utilizando o protocolo Modbus RTU encontra-se nos anexos deste artigo devido à sua extensão.

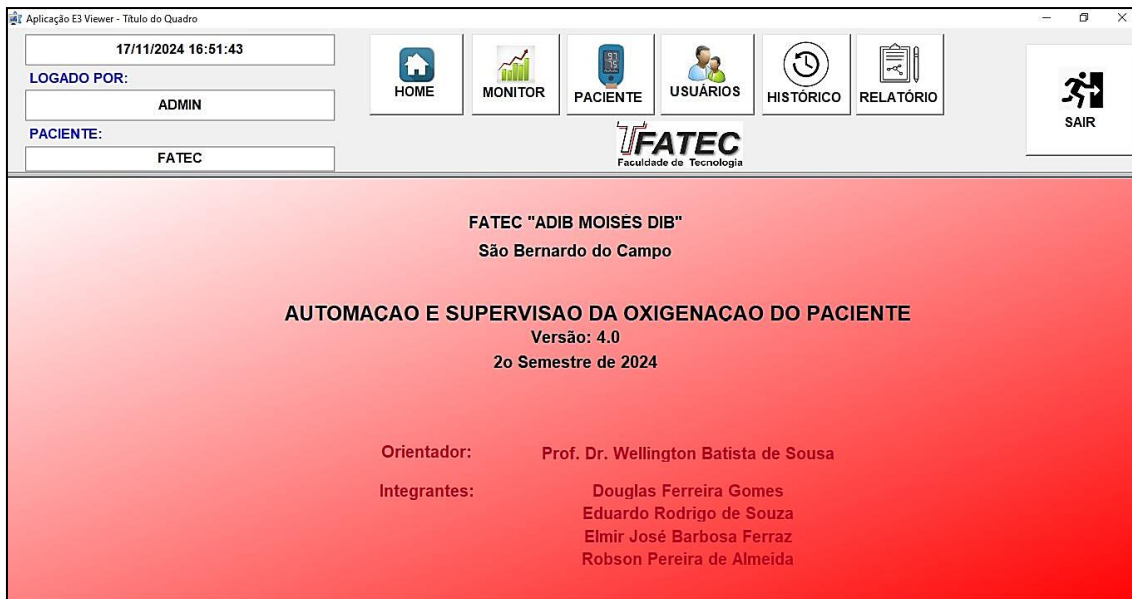
⁵ Legenda: P1/N1 indica o número do equipamento escravo na comunicação. P2/N2 indica a porta USB. P3/N3 indica se o registro começa com 0 ou com 1. P4/N4 indica o endereço do registro para conexão conforme o programa do Arduino Uno.

É importante destacar que, para identificar o botão de comando acionado e executar o comando no Arduino Uno, foi utilizada a seguinte lógica binária:

- Se for pressionado o botão "0 L/min", o valor da variável "BOTÕES" será 0, ou mais especificamente, 00000000 e o Arduino Uno irá verificar que não há 1 nos bits deste registro.
- Se for pressionado o botão "2,0L/min", o valor da variável "BOTÕES" será 1, ou mais especificamente, 00000001 e o Arduino Uno irá verificar que o primeiro bit da direita para a esquerda neste registro é igual a 1.
- Se for pressionado o botão "3,0L/min", o valor da variável "BOTÕES" será 2, ou mais especificamente, 00000010 e o Arduino Uno irá verificar que o segundo bit da direita para a esquerda neste registro é igual a 1.
- Se for pressionado o botão "4,0L/min", o valor da variável "BOTÕES" será 4, ou mais especificamente, 00000100 e o Arduino Uno irá verificar que o terceiro bit da direita para a esquerda neste registro é igual a 1.
- Se for pressionado o botão "4,5L/min", o valor da variável "BOTÕES" será 8, ou mais especificamente, 00001000 e o Arduino Uno irá verificar que o quarto bit da direita para a esquerda neste registro é igual a 1.
- Se for pressionado o botão "5,0L/min", o valor da variável "BOTÕES" será 16, ou mais especificamente, 00010000 e o Arduino Uno irá verificar que o quinto bit da direita para a esquerda neste registro é igual a 1.

Deste modo foi finalizada a sexta etapa. Conforme a sétima etapa, as telas organizadas e formatadas na versão final do sistema supervisor têm sua visualização geral ilustrada nas próximas figuras. A seguir, a Figura 5.7 está ilustrando a aparência final do Menu e da tela Home.

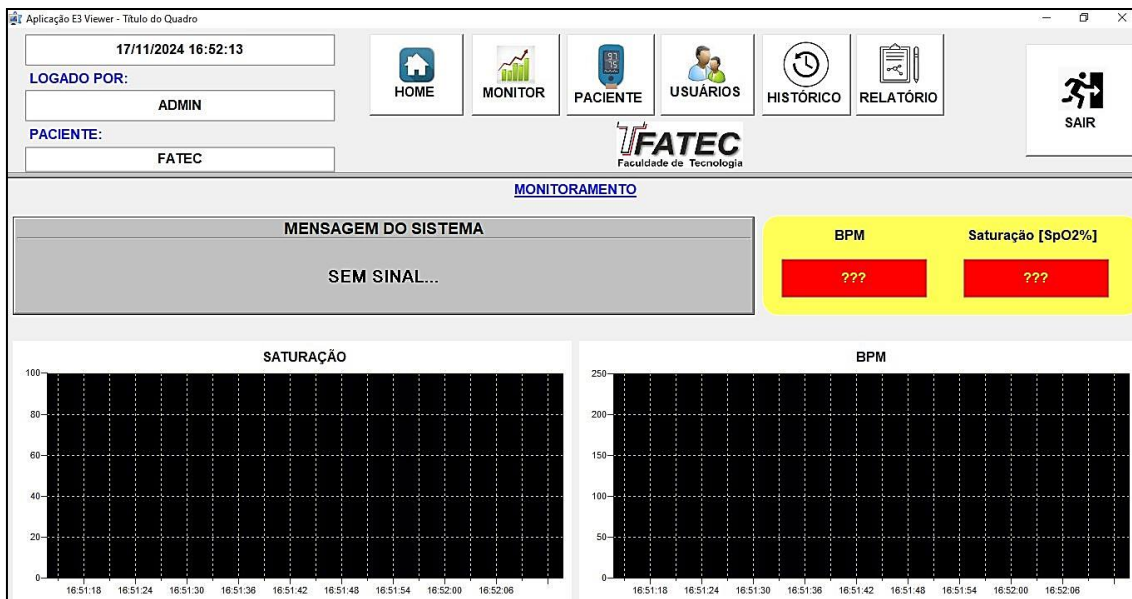
Figura 5.7 - Menu e Home



Fonte: Autoria própria, 2024

Abaixo, a Figura 5.8 está demonstrando a aparência final do Menu e da tela Monitor.

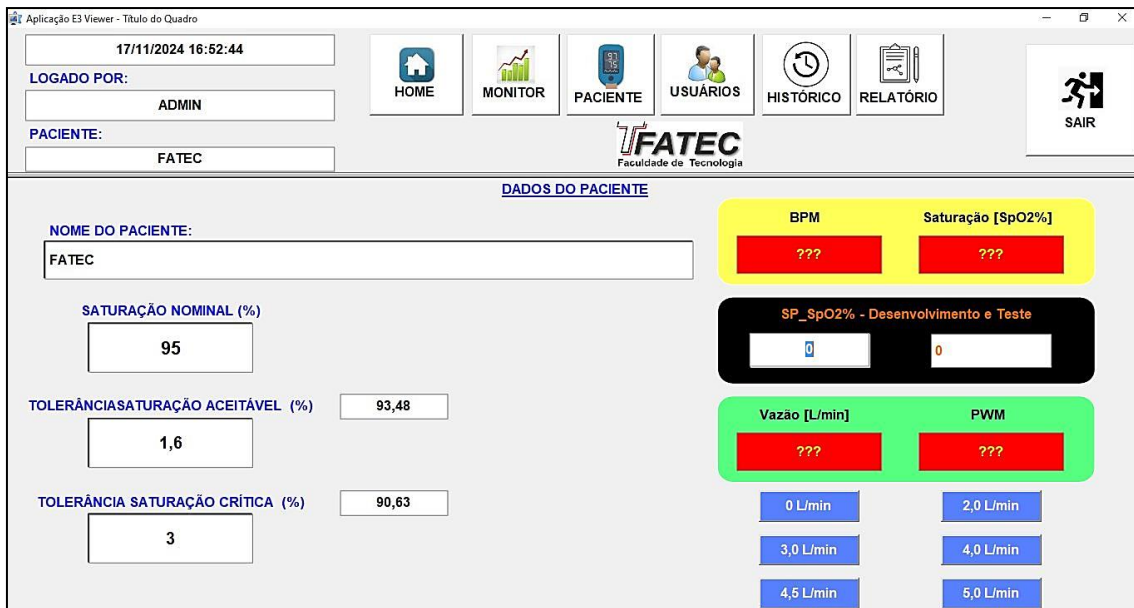
Figura 5.8 - Menu e Monitor



Fonte: Autoria própria, 2024

A seguir, a Figura 5.9 está ilustrando a aparência final do Menu e da tela Paciente.

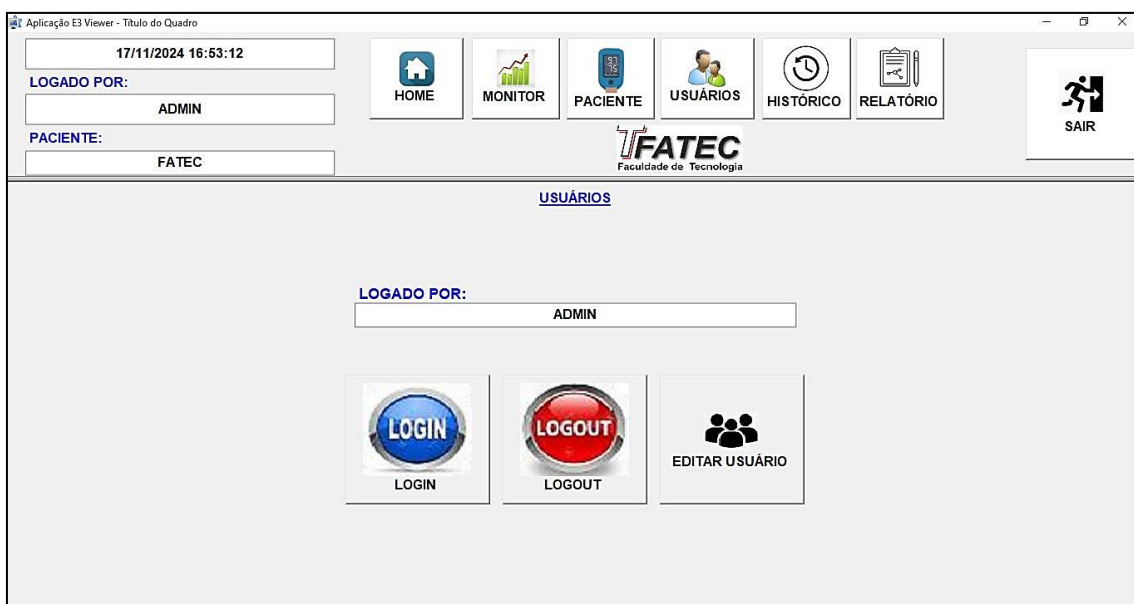
Figura 5.9 - Menu e Paciente



Fonte: Autoria própria, 2024

A seguir, a Figura 5.10 está ilustrando a aparência final do Menu e da tela Usuários.

Figura 5.10 - Menu e Usuários



Fonte: Autoria própria, 2024

Abaixo, a Figura 5.11 está demonstrando a aparência final do Menu e da tela Histórico.

Figura 5.11 - Menu e Histórico

Aplicação E3 Viewer - Título do Quadro

17/11/2024 16:53:47

LOGADO POR: ADMIN

PACIENTE: FATEC

HOME MONITOR PACIENTE USUÁRIOS HISTÓRICO RELATÓRIO SAIR

FATEC
Faculdade de Tecnologia

HISTÓRICO

DATA_HORA	PACIENTE	SATURACAO	BATIMENTO	OXIGENACAO	COD_MSG	MENSAGEM
17/11/2024 16:53:27	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:53:17	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:53:07	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:57	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:47	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:37	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:27	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:17	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:52:07	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:57	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:47	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:37	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:27	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:17	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:51:07	FATEC	%	BPM	, L/min	3	SEM SINAL
17/11/2024 16:50:57	FATEC	%	BPM	, L/min	0	
17/11/2024 16:50:47	FATEC	%	BPM	, L/min	0	
17/11/2024 16:50:37	FATEC	%	BPM	, L/min	0	

Registro: 1 de 3987

CONSULTAR PACIENTE

Fonte: Autoria própria, 2024

E abaixo, a Figura 5.12 está ilustrando a aparência final do Menu e da tela Relatório.

Figura 5.12 - Menu e Relatório

Aplicação E3 Viewer - Título do Quadro

17/11/2024 16:54:25

LOGADO POR: ADMIN

PACIENTE: FATEC

HOME MONITOR PACIENTE USUÁRIOS HISTÓRICO RELATÓRIO SAIR

FATEC
Faculdade de Tecnologia

RELATÓRIO

NOME DO PACIENTE:
FATEC

VISUALIZAR RELATÓRIO DO PACIENTE NA TELA

IMPRIMIR RELATÓRIO DO PACIENTE

EXPORTAR RELATÓRIO PACIENTE [.PDF]

EXPORTAR RELATÓRIO PACIENTE [.XLS]

Fonte: Autoria própria, 2024

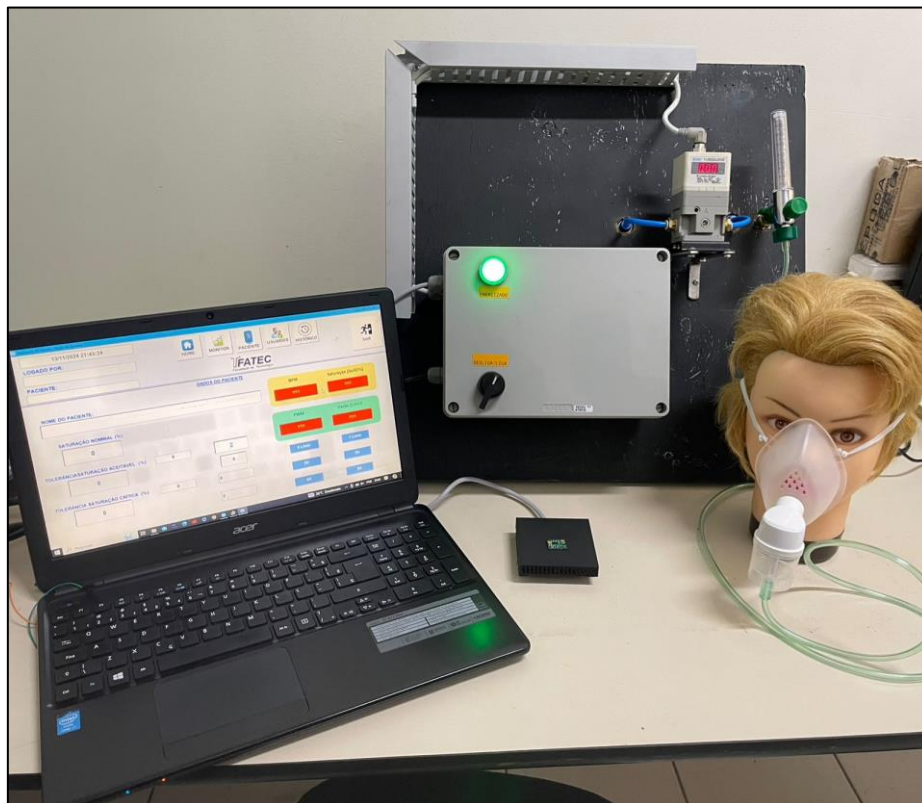
Em paralelo com a finalização da sétima etapa, foi iniciada a oitava etapa referente à montagem final e ao teste funcional, deixando a instalação mais robusta e visualmente agradável. Buscando auxiliar na visualização do projeto durante a exposição, foi comprada uma máscara de oxigênio e uma cabeça de boneca. Para a organização, também foi comprada

uma caixa plástica de montagem elétrica, uma chave de energia liga/desliga e uma *LED* verde para indicar a alimentação de energia.

Neste momento, foi definido dispor o Arduino Uno e a fonte de alimentação de 24 V dentro da caixa plástica. Para que fosse feita a fixação da caixa plástica, da válvula de controle e do fluxômetro conforme ilustrado na Figura 5.13. Foram disponibilizados madeira usada e suportes de fixação para o grupo. Um dos integrantes do grupo também disponibilizou a calha de organização dos condutores.

Concluindo a montagem da oitava etapa, foram realizados testes funcionais conforme descrito no desenvolvimento. Ao final deste teste funcional foi identificado que todas as tarefas propostas para o protótipo estão sendo realizadas e que não havia nenhuma parte do projeto que não estava funcionando corretamente. Além disso, foi confirmado através da quantidade de leituras seguidas dos mesmos integrantes do grupo que a variação da leitura do dispositivo protótipo utilizando o oxímetro MAX30100 foi de aproximadamente 2%, assim como indicado no datasheet do componente. Este também é o valor de referência nos datasheets dos oxímetros hospitalares das marcas estabelecidas

Figura 5.13 - Montagem Final e Teste Funcional



Fonte: Autoria própria, 2024

6. Considerações finais

Este projeto de pesquisa de estudo de caso que tem como tema Automação e Supervisão da Oxigenação do Paciente, indicou como objetivo principal desenvolver um dispositivo para supervisionar os dados de oxigenação do paciente, notificando os responsáveis pelo ajuste em caso de variações relevantes desta necessidade, e indicou como objetivo secundário gerar um banco de dados com registro histórico das leituras e das alterações realizadas pelos funcionários no decorrer do tempo.

O tema tem como principais pontos fortes a relevância social em relação à melhoria do atendimento e da atividade de trabalho dos profissionais de saúde envolvidos, bem como a oportunidade comercial em relação a produção de um dispositivo que pode ser vendido. Por outro lado, o tema tem como principais pontos fracos a amplitude de conhecimento específico necessário ao processo de oxigenação, a falta de informação em relação aos protocolos dos fabricantes hospitalares de oxímetro de dedo, a dificuldade e a responsabilidade de realizar testes hospitalares, além da impossibilidade de automatização em relação à administração do oxigênio uma vez que o oxigênio hospitalar é considerado um medicamento que depende de prescrição adequada.

Ressalta-se que mesmo sem automatizar a administração do oxigênio, tornar o processo de oxigenação suplementar menos manual, mais preciso e remotamente ajustável é uma proposta relevante, havendo a oportunidade de comprovação da relevância percebida de acordo com pesquisa de opinião na escala *Likert*. Deste modo, em acordo com os objetivos propostos e com os resultados obtidos, é possível afirmar que o protótipo está validado para fins acadêmicos e que este dispositivo atendeu aos objetivos, executando todas as tarefas e as atividades que foram propostas, embora não necessariamente estas estejam sendo executadas do modo como foram planejadas inicialmente conforme descrito.

O conteúdo exposto no trabalho e apresentado através do protótipo construído contribui para trazer a atenção à aplicação do avanço tecnológico a este processo no ambiente hospitalar, uma vez que o hospital é um ambiente em que as novas tecnologias são amplamente aplicadas e este processo permanece sendo executado de forma manual. Então, deixamos como sugestões para futuras pesquisas e melhorias em relação ao nosso projeto:

- a aplicação de outra placa microcontrolada mais robusta, como por exemplo, o ESP32. Para este estudo, acreditamos que seria necessário alterar o foco em relação ao custo e a eficiência da placa microcontrolada, focando mais especificamente na robustez e na segurança do processo que está diretamente envolvido com a vida;

- a alteração do supervisor para ajuste decimal da vazão de oxigênio e, se necessário, o desenvolvimento de uma correlação geral para a vazão de oxigênio administrada no

paciente em L/min e a pressão de entrada da válvula de controle na unidade disponível;

- a realização de testes fora do ambiente acadêmico como um equipamento auxiliar e de forma supervisionada para levantamento de dados em relação a vida-útil e a confiabilidade em relação ao tempo de uso.

REFERÊNCIAS

- AMARAL, J. L. G. D. et al. Monitorização da Respiração: Oximetria e Capnografia. **Revista Brasileira de Anestesiologia**, v. 42, p. 51-58, Janeiro-Fevereiro 1992. Disponível em: <<https://ebm.ufabc.edu.br/wp-content/uploads/2014/09/Amaral-1992-Monitoriza%C3%A7%C3%A3o-da-respira%C3%A7%C3%A3o-oximetria-e-capnografia.pdf>>. Acesso em: 30 Maio 2024.
- ARAUJO, W. M. et al. Visão geral sobre Microcontroladores e Prototipagem com Arduino. **Tecnologias em Projeção**, Agosto 2019. 36-46. Disponível em: <<https://revista.projecao.br/index.php/Projecao4/article/view/1357>>. Acesso em: 06 Junho 2024.
- BRAGA, Rosana. **História das Linguagens de Programação**, 2006. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/3439710/mod_resource/content/1/Aula08-HistoriaLinguagensProgramacao.pdf>. Acesso em 06 Junho 2024.
- BOJORGE, N. Departamento de Engenharia Química e de Petróleo – UFF. **Aula 05 - Sistemas de Medição de Vazão**, 2016. Disponível em: <<https://acrobat.adobe.com/id/urn:aaid:sc:VA6C2:3df2a9ff-b9c3-4f1e-8caf-acbda5d2a344>>. Acesso em: 28 Abril 2024.
- COELHO, Marcelo N. Instituto Federal São Paulo Campus Cubatão. **Apostila de Sistemas Supervisórios**, 2010. Disponível em: <https://professorcesarcosta.com.br/upload/imagens_upload/Apostila_%20Sistema%20Supervis%C3%B3rio.pdf>. Acesso em: 01 Novembro 2024.
- FILHO, N. D. A. **O que é saúde?** [online]. Rio de Janeiro: Fiocruz, 2011. 160 p. ISBN 978-85-7541-220-6. Visualização das páginas 01-39. Disponível em: <https://books.google.com.br/books?id=4-jrAgAAQBAJ&lpg=PA7&dq=o%20que%20%C3%A9%20sa%C3%BAde&lr&hl=pt-BR&pg=PP1#v=onepage&q&f=false>. Acesso em: 25 Março 2024.
- GOMES, F. CNN. **Saturação de oxigênio**: entenda o que é, níveis e como fazer a medição, 2023. Disponível em: <<https://www.cnnbrasil.com.br/saude/correspondente-medico-o-que-e-saturacao-de-oxigenio-e-qual-o-nivel-normal/>>. Acesso em: 25 Março 2024. Entrevista com o Neurocientista Fernando Gomes.
- GOV.BR. GOV.BR. **Sistema Único de Saúde**. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/sus>>. Acesso em: 25 Março 2024.
- HORTA, R. L. et al. O estresse e a saúde mental de profissionais da linha de frente da COVID-19 em hospital geral. **Jornal Brasileiro De Psiquiatria**, Janeiro 2021. 30-38. Disponível em: <<https://www.scielo.br/j/jbpsiq/a/3wN8kZGYJVd3B4tF6Wcctgs/?lang=pt>>. Acesso em: 20 Abril 2024.
- KERSCHBAUMER, R. Engenharia de Controle e Automação. Microcontroladores., 2018. Disponível em: <<https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2018/02/Apostila-Microcontroladores.pdf>>. Acesso em: 06 Junho 2024.

LISSA, C. B. H. Suplementação de Oxigênio no Atendimento Pré-Hospitalar (APH). **Corpo de Bombeiros Militar do Distrito Federal - CBMDF**, Distrito Federal, 17 Novembro 2022. 42. Disponível em: <<https://biblioteca.cbm.df.gov.br/jspui/bitstream/123456789/381/1/Hanna%20Lissa%20-%201213.pdf>>. Acesso em: 05 Junho 2024.

MANCUZO, E. V. et al. Função pulmonar de pacientes hospitalizados com COVID-19, 45 dias após a alta hospitalar: primeiro relato de um estudo multicêntrico prospectivo no Brasil. **Jornal Brasileiro de Pneumologia**, Minas Gerais, n. 20210162, p. 47 (6), Agosto 2021. ISSN 1806-3756. Disponível em: <<https://www.scielo.br/j/bpneu/a/zqdZPHpqHFJYKKB3ntnHwLM/?format=pdf&lang=pt>>. Acesso em: 30 Maio 2024.

MATHIAS, A. C. Fluid Controls do Brasil. **Apostila Válvulas Industriais [online]**. Disponível em: <<https://acrobat.adobe.com/id/urn:aaid:sc:VA6C2:7eaea317-7353-43c3-a1b2-d9b7fe1cfb83>>. Acesso em: 28 Abril 2024.

MCROBERTS, M. **Arduino Básico**. 2ª Edição. ed. São Paulo: Novatec, 2015. ISBN 978-85-7522-404-5. Visualização das páginas 01-25. Disponível em: <<https://s3.novatec.com.br/capitulos/capitulo-9788575224045.pdf>>. Acesso em: 05 Maio 2024.

NISE, N. S. **Engenharia de Sistemas de Controle**. 6ª Edição. ed. Rio de Janeiro: LTC, 2013. ISBN 978-85-216-2136-2 e ISBN 978-0470-54756-4. Disponível em: <https://docs.ufpr.br/~pettres/EXATAS/Fun%C3%A7%C3%B5es%20de%20transfer%C3%A7%C3%A3o/Engenharia_de_Sistemas_de_Control_6_Edi.pdf>. Acesso em 30 Maio 2024.

NISHIOKA, S. D. A. UNASUS. **O que é importante saber sobre o uso do oxímetro de pulso para monitorar a COVID-19 em casa**, 2022. Disponível em: <<https://www.unasus.gov.br/especial/covid19/markdown/502>>. Acesso em: 25 Março 2024.

OLIVEIRA, S. D. **Internet das Coisas com ESP8266, Arduino e Raspberry PI**. 1ª Edição. ed. São Paulo: Novatec, 2017. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=E8gmDwAAQBAJ&oi=fnd&pg=PA17&dq=microcontroladores+historia&ots=5RMc9qb_3y&sig=Zu9CdborOrqdg08cOs43bK_9org#v=onepage&q&f=true>. Acesso em: 30 Maio 2024.

PAULA, P. D. Biblioteca Virtual em Saúde. **71% dos brasileiros têm os serviços públicos de saúde como referência**. Disponível em: <<https://bvsmis.saude.gov.br/71-dos-brasileiros-tem-os-servicos-publicos-de-saude-como-referencia/>>. Acesso em: 25 Março 2024. Patrícia de Paula participa da Agência de Saúde.

PEREIRA, F. **Microcontroladores PIC - Programação em C**. 7ª Edição. Ed. São Paulo: Érica - Saraiva, 2013. ISBN 85-7194-935-2. Disponível em: <http://www.adjutojunior.com.br/microcontroladores/PIC_Programa%C3%A7%C3%A3o_em_C.pdf>. Acesso em: 30 Maio 2024.

SALERNO, P. et al. **Manual de Oxigenoterapia**. Centro Universitário São Camilo. São Paulo, p. 35. 2023. (978-65-86702-38-5). Disponível em: <https://saocamilosp.br/_app/views/publicacoes/outraspublicacoes/Manual%20de%20Oxigenoterapia.pdf>. Acesso em: 25 Março 2024.

SILVA, V. Z. M. D. et al. Recomendações para a utilização de oxigênio suplementar (oxigenoterapia) em pacientes com COVID-19*. **ASSOBRAFIR Ciência**, v. 11, n. Suplemento 1, p. 87-91, 11 Agosto 2020. ISSN 2177-9333. Disponível em: <<https://assobrafirciencia.org/article/doi/10.47066/2177-9333.AC20.covid19.008#:~:text=Recomenda%2Dse%20a%20utiliza%C3%A7%C3%A3o%20de,com%20m%C3%A1scara%20reservat%C3%B3rio%20n%C3%A3o%20reinalante>>. Acesso em: 30 Maio 2024.

SILVEIRA, S. R. S. et al. **Paradigmas de Programação: Uma Introdução**. Belo Horizonte: Synapse, 2021. ISBN 978-65-88890-08-0. Disponível em: <<https://acrobat.adobe.com/id/urn:aaid:sc:VA6C2:d1aac0e2-6ea9-41a0-939e-98ba04cf7d14>>. Acesso em: 30 Maio 2024.

SPDM. SPDM. **A importância social do Hospital vai muito além do atendimento médico**, 2 Julho 2015. Disponível em: <<https://spdm.org.br/noticias/mais-noticias/a-importancia-social-do-hospital-vai-muito-alem-do-atendimento-medico/>>. Acesso em: 25 Março 2024.

ANEXOS

Anexo 01 – Programa do módulo MAX30100

```
#include <Wire.h> //Biblioteca para comunicação I2C
#include "MAX30100_PulseOximeter.h" //Biblioteca para uso do sensor de frequência
cardíaca e oxímetro MAX30100

#define REPORTING_PERIOD_MS    1000 // Intervalo entre a medição de cada
amostra

uint32_t tsLastReport = 0; // Tempo da última amostra detectada

PulseOximeter pox; //Declaração objeto pox

void onBeatDetected() { //Função executada quando um pulso é detectado
    Serial.println("Batimento detectado");
}

void setup() {
    Serial.begin(9600); //Inicialização da comunicação serial
    Serial.print("Iniciando funcionamento do sensor ");

    if (!pox.begin()) //Testa a comunicação com o sensor
    {
        Serial.println("Falhou ! "); //Se a comunicação não foi realizada corretamente exibe
a mensagem "Falhou"
        for (;;)
    } else
    {
        Serial.println("Sucesso ! "); //Se a comunicação foi realizada com sucesso exibe a
mensagem "Sucesso !"
    }

    pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA); // Define que o sensor use
7,6 mA para o led

    pox.setOnBeatDetectedCallback(onBeatDetected); // Registra todas as vezes em que
um batimento for detectado
```

```

}

void loop() {
  pox.update(); //Atualiza a leitura do sensor

  if (millis() - tsLastReport > REPORTING_PERIOD_MS) { //Imprime a leitura em
intervalos de 1000ms
    Serial.print("Taxa de Batimento : ");
    Serial.print(pox.getHeartRate()); // Faz a leitura de batimento cardíaco
    Serial.print("bpm / Saturacao sangue(SP02) : ");
    Serial.print(pox.getSpO2()); //Realiza a leitura da saturação do sangue
    Serial.println("%");

    tsLastReport = millis();
  }
}

```

Anexo 02 – Programa para teste da válvula e levantamento de dados

```

int CtrlValv = 9; // Associa a saída PWM de controle da a válvula ao pino 09

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //Inicialização da comunicação serial
  pinMode(CtrlValv,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available()){
    delay(5000);
    Serial.println("Escreva um numero de 0 a 255: ");
    char num=Serial.parseInt();
    analogWrite(CtrlValv,num);
  }
}

```

Anexo 03 – Programa final do Arduino Uno utilizando o protocolo Modbus RTU

```
#include "Modbusino.h"
#include "EEPROM.h"
#include <Wire.h> //Biblioteca para comunicação I2C
#include "MAX30100_PulseOximeter.h" //Biblioteca para uso do sensor de frequência
cardíaca e oxímetro MAX30100

//-----

// Sub-rotinas e programas específicos do Oxímetro

#define REPORTING_PERIOD_MS 3000 // Intervalo entre a medição de cada
amostra - original era 1000
uint32_t tsLastReport = 0; // Tempo da última amostra detectada
PulseOximeter pox; //Declaração objeto pox

void onBeatDetected() { //Função executada quando um pulso é detectado
  Serial.println("Batimento detectado");
}

//-----

/*****
  ARDUINO - COMUNICAÇÃO SERIAL (MODBUS)COM ELIPSE E3
  Estrutura do Prof. Ms. Marco Antonio Baptista de Sousa
  *****/

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

/* Aloca o Número de Registradores*/
uint16_t tab_reg[10]; // AQUI PODE-SE USAR ATÉ 10 REGISTROS

/* Variaveis do Projeto */
```

```

const int Trimpot = 0;

int Botao;
int Leitura_Trimpot = 0;

int Comando = 0;

// ARDUINO -> ELIPSE E3
unsigned int Status1 = 0;
unsigned int Status2 = 0; //valor
unsigned int Status3 = 0; //Saturação
unsigned int Status4 = 0; //BPM

//ELIPSE E3-> ARDUINO
unsigned int Controle1 = 0; //5 Botões
unsigned int Controle2= 0;
unsigned int Controle3 = 0;

/* PINS
Add more registers if needed
Digital input pins 2,3,4,5,6,x7
Digital output pins 7,8,9,12,13
Analog output pins 10,11 (PWM)
Analog input pins 0,1
*/

unsigned int DI_2 = 2;
unsigned int DI_3 = 3;
unsigned int DI_4 = 4;
unsigned int DI_5 = 5;
unsigned int DI_6 = 6;
unsigned int DI_7 = 7;
unsigned int DO_8 = 8;
unsigned int DO_9 = 9;
unsigned int DO_12 = 12;

```

```

unsigned int DO_13 = 13;
unsigned int valor = 0;

unsigned int AI_0 = 0;
unsigned int AI_1 = 1;
unsigned int AO_0 = 10;
unsigned int AO_1 = 11;

int Estado_port= 0;
int parte1 = 0;
int parte2 = 0;
int Contagem = 0;
int Lido = 0;

//FUNÇÃO DE BIT

#define testBit(valor,bit)  (valor & (1<<bit))
#define setBit(valor,bit)  (valor |= (1<<bit))
#define clearBit(valor,bit)  (valor &= ~(1<<bit))

void setup() {
  // SE QUISER MONITORAR PELA SERIAL HABILITE LINHA ABAIXO
  Serial.begin(9600);
  Serial.print("Iniciando funcionamento do sensor ");

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

  /* Definindo o comportamento dos pinos */
  pinMode(DI_2,INPUT_PULLUP);
  pinMode(DI_3,INPUT_PULLUP);
  pinMode(DI_4,INPUT_PULLUP);
  pinMode(DI_5,INPUT_PULLUP);
  pinMode(DI_6,INPUT_PULLUP);
  pinMode(DI_7,INPUT_PULLUP);

```

```

pinMode(DO_8,OUTPUT);
pinMode(DO_9,OUTPUT);
pinMode(DO_12,OUTPUT);
pinMode(DO_13,OUTPUT);

if (!pox.begin()) //Testa a comunicação com o sensor
{
  Serial.println("Falhou ! "); //Se a comunicação não foi realizada corretamente exibe
a mensagem "Falhou"
  for (;);
} else
{
  Serial.println("Sucesso ! "); //Se a comunicação foi realizada com sucesso exibe a
mensagem "Sucesso !"
}
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA); // Define que o sensor use
7,6 mA para o led
pox.setOnBeatDetectedCallback(onBeatDetected); // Registra todas as vezes em que
um batimento for detectado
}

//-----
void loop() {
//

  while(1)
  {
//=====
// ATUALIZA PEGANDO CONTROLE2 DO SUPERVISÓRIO E ESCRIVE NOS
BITS (SÓ OS DE SAIDA OBEDECERÃO)
//=====
// LÊ A PALAVRA Controle1 E ACIONA AS SAÍDAS CONFORME O BIT
ACIONADO
//

```

```

// if(!testBit(Controle1,8)){ //CHECA SE BIT ESTÁ ATIVADO
//   digitalWrite(DO_8,LOW);
// }
// else{           //se não
//   digitalWrite(DO_8,HIGH);
// }
//
// if(!testBit(Controle1,9)){ //CHECA SE BIT ESTÁ ATIVADO
//   digitalWrite(DO_9,LOW);
// }
// else{           //se não
//   digitalWrite(DO_9,HIGH);
// }
//
// if(!testBit(Controle1,12)){ //CHECA SE BIT ESTÁ ATIVADO
//   digitalWrite(DO_12,LOW);
// }
// else{           //se não
//   digitalWrite(DO_12,HIGH);
// }
//
// if(!testBit(Controle1,13)){ //CHECA SE BIT ESTÁ ATIVADO
//   digitalWrite(DO_13,LOW);
// }
// else{           //se não
//   digitalWrite(DO_13,HIGH);
// }

if (Controle1==0){
  valor=0;
}

if(!testBit(Controle1,0)){ //CHECA SE O BIT ESTÁ ATIVADO
} else{
  valor=51; //valor equivale a 1/5 (1V no Arduino UNO)
}

```

```

    }

    if(!testBit(Controle1,1)){ //CHECA SE O BIT ESTÁ ATIVADO
        } else{
            valor=102; //valor equivale a 2/5 (2V no Arduino UNO)
        }

    if(!testBit(Controle1,2)){ //CHECA SE O BIT ESTÁ ATIVADO
        } else{
            valor=153; //valor equivale a 3/5 (3V no Arduino UNO)
        }

    if(!testBit(Controle1,3)){ //CHECA SE O BIT ESTÁ ATIVADO
        } else{
            valor=204; //valor equivale a 4/5 (4V no Arduino UNO)
        }

    if(!testBit(Controle1,4)){ //CHECA SE O BIT ESTÁ ATIVADO
        } else{
            valor=255; //valor equivale a 5/5 (5V no Arduino UNO)
        }

    pox.update(); //Atualiza a leitura do sensor

    if (millis() - tsLastReport > REPORTING_PERIOD_MS) { //Imprime a leitura em
    intervalos do período determinado em ms
        Serial.print("Taxa de Batimento : ");
        Serial.print(pox.getHeartRate()); // Faz a leitura de batimento cardíaco
        Serial.print("bpm / Saturacao sangue(SP02) : ");
        Serial.print(pox.getSpO2()); //Realiza a leitura da saturação do sangue
        Serial.println("%");

        tsLastReport = millis();
    }

```



```

//=====
// ATUALIZA STATUS1 LENDO TODAS AS ENTRADAS E MOSTRA NO
// SUPERVISÓRIO (SÓ OS DE ENTRADA OBEDECERÃO)
//=====
// ESTÁ LENDO DOS BITS DI_2 A DI_7 E ACIONANDO BITS EM Status1

//// for (Contagem =2; Contagem<=13;Contagem++){
// int Lido = digitalRead(Contagem);
// if (Lido==0){ //CHECA SE BIT ESTÁ ATIVADO
// clearBit(Status1,Contagem);
// }
// else{
// setBit(Status1,Contagem); //configura pino PB5 como saída
// setBit(Status1,15); //configura pino PB5 como saída
// }
// }
// for (Contagem =2; Contagem<=13;Contagem++){

// LER ENTRADAS E COLOCAR NO BIT CORRESPONDENTE DE "Status1"
if (digitalRead(DI_2)==0){ //CHECA SE BIT ESTÁ ATIVADO
clearBit(Status1,DI_2);
//Serial.println(Status1);
}
else{
setBit(Status1,DI_2);
// Serial.println(Status1);
}
if (digitalRead(DI_3)==0){ //CHECA SE BIT ESTÁ ATIVADO
clearBit(Status1,DI_3);
}
else{
setBit(Status1,DI_3);
}
}

```

```

if (digitalRead(DI_4)==0){ //CHECA SE BIT ESTÁ ATIVADO
    clearBit(Status1,DI_4);
}
else{
    setBit(Status1,DI_4);
}

if (digitalRead(DI_5)==0){ //CHECA SE BIT ESTÁ ATIVADO
    clearBit(Status1,DI_5);
}
else{
    setBit(Status1,DI_5);
}

if (digitalRead(DI_6)==0){ //CHECA SE BIT ESTÁ ATIVADO
    clearBit(Status1,DI_6);
}
else{
    setBit(Status1,DI_6);
}

if (digitalRead(DI_7)==0){ //CHECA SE BIT ESTÁ ATIVADO
    clearBit(Status1,DI_7);
}
else{
    setBit(Status1,DI_7);
}

//=====
// COLOCA VALORES LIDOS DAS ANALÓGICAS EM Status2 E Status3 E
Status4
    Status2 = valor;
    Status3 = pox.getHeartRate();
    Status4 = pox.getSpO2();
// COLOCA VALORES DE Controle2 E Controle3 NAS SAÍDAS ANALÓGICAS
    analogWrite(AO_0,valor);
    //analogWrite(AO_1,Controle3);
//=====

```

```
// FAZ ATUALIZAÇÃO DAS VARIÁVEIS DO PROGRAMA
// DO ARDUÍNO -> ELIPSE
tab_reg[2] = Status1;
tab_reg[3] = Status2;
tab_reg[4] = Status3;
tab_reg[8] = Status4;
// DO ELIPSE->ARDUINO
Controle1 = tab_reg[5];
Controle2 = tab_reg[6];
Controle3 = tab_reg[7];
//=====
//Atualizando o Modbus
modbusino_slave.loop(tab_reg, 10);
}
}
```