

VentiVet: Desenvolvimento de interface gráfica em MATLAB para ventilador mecânico de pequenos animais



<https://doi.org/10.56238/sevened2023.006-159>

Amanda Fila de Lima

Grau de formação mais alto: Médio/Técnico
Instituição acadêmica: Instituto Federal do Paraná

Brayan Agio de Miranda Andrade

Grau de formação mais alto: Médio/Técnico
Instituição acadêmica: Instituto Federal do Paraná

Beatriz do Santos Pês

Grau de formação mais alto: Doutorado
Instituição acadêmica: Instituto Federal do Paraná

Diego Tefili

Grau de formação mais alto: Mestrado
Instituição acadêmica: Instituto Federal do Paraná

Marcos Santos Hara

Grau de formação mais alto: Doutorado
Instituição acadêmica: Instituto Federal do Paraná

RESUMO

Um ventilador pulmonar é um equipamento de suporte respiratório para procedimentos cirúrgicos ou clínicos, como os que necessitam de anestesia geral, por exemplo. A construção deste tipo de equipamento pode envolver conhecimentos de

diversas áreas, como sistemas de controle em malha fechada, desenvolvimento de software, sensoriamento e protocolos de comunicação. O presente texto expõe o desenvolvimento de um ventilador pulmonar voltado para animais de pequeno porte e a implementação de uma interface gráfica para monitoramento e ajuste de parâmetros. O ventilador veterinário construído conta com uma plataforma microcontrolada Arduino Uno, um sensor de pressão e válvulas de esfera com controle proporcional de fluxo, construídas com servo motores, desempenhando as funções de válvula expiratória e inspiratória, por meio de um algoritmo de controle proporcional implementado em firmware. Além disso, a interface gráfica é desenvolvida em MATLAB, possibilitando ajustes de parâmetros e visualização de dados ventilatórios e alarmes, que são essenciais para um funcionamento seguro do ventilador. Diferentemente de outros projetos apresentados na literatura, o protótipo utiliza ar comprimido e não manual, com a implementação do modo de ventilação PCV - ventilação com pressão controlada, em ciclos controlados. O protótipo possui ainda uma interface dinâmica com exibição do histórico da pressão medida em um gráfico em tempo real, além de controles e ajustes.

Palavras-chave: Ventilação mecânica, Controle proporcional, MATLAB, Interface gráfica.

1 INTRODUÇÃO

A ventilação mecânica tem como objetivo prover assistência ou substituir parte das funções respiratórias de seu usuário, podendo ser invasiva (prótese inserida diretamente nas vias aéreas) ou não invasiva (máscara como instrumento que realiza as trocas gasosas) (CARVALHO *et al.*, 2007).

O modo de funcionamento do protótipo é ventilação controlada a pressão, em inglês *Pressure Controlled Ventilation* (PCV) no qual o volume corrente (V_t) é variável. Os ciclos são pré-determinados pelo tempo e pelo objetivo de pressão inspiratória, que deve ser alcançada no início da inspiração e mantida durante toda esta fase (CARVALHO *et al.*, 2007). O protótipo construído tem uma interface digital com as informações referentes às condições do paciente e botões interativos que



permitem a personalização dos parâmetros de acordo com as características de cada paciente. O protótipo tem como objetivo uma utilização dinâmica, possibilitando facilidade em sua utilização, focando preliminarmente o uso em clínicas veterinárias.

Neste projeto, é utilizada a plataforma microcontrolada Arduino Uno e o *software* MATLAB, em sua versão R2023a, pois este permite o desenvolvimento de aplicativos que se comunicam diretamente com Arduino Uno. Com essas ferramentas, construiu-se a um protótipo que integra *hardware* eletrônico microcontrolado composto por atuadores e sensores, formando um controlador em malha fechada, além de *firmware* e *software* dedicados. Foi também definido um protocolo próprio de comunicação entre o microcontrolador e o computador para troca de dados. Os valores lidos pela placa são armazenados no computador, que os exibe na tela, na qual o operador pode monitorar o funcionamento da ventilação, receber alarmes em caso de falhas e alterar variáveis do processo, como frequência respiratória e tempo de inspiração.

As próximas seções descrevem o funcionamento do sistema respiratório, os tipos de ventilação mecânica, os componentes utilizados no desenvolvimento do protótipo e os resultados obtidos.

2 FUNDAMENTAÇÃO

Apresenta-se aqui uma compilação de alguns conhecimentos necessários para a implementação do protótipo, dispendo desde os princípios da ventilação mecânica necessários para a plena compreensão do desenvolvimento até os componentes utilizados e suas especificações relevantes.

2.1 O SISTEMA RESPIRATÓRIO

A respiração é um processo indispensável para a execução plena das funções metabólicas teciduais e essencial para o organismo. O sistema respiratório fornece os meios para que isso ocorra, através das vias aéreas superiores, compostas pelas cavidades nasais, faringe e laringe, e das vias aéreas inferiores, compostas pela traqueia, brônquios, bronquíolos e alvéolos. As vias aéreas inferiores compõem a estrutura do pulmão, onde ocorre o processo de hematose, removendo dióxido de carbono (CO₂) do sangue e introduzindo oxigênio (O₂). Isso ocorre na menor estrutura pulmonar, os alvéolos.

A inspiração se dá por meio da contração dos músculos respiratórios, que provoca uma queda de pressão alveolar em relação ao meio externo, induzindo o ar atmosférico a entrar nas vias aéreas e acessar o interior dos pulmões através de sua distensão. A expiração sucede a inspiração, que de forma contrária, provoca a saída do volume de ar adquirido pelo relaxamento da musculatura, fazendo os pulmões voltarem à posição inicial por meio da energia elástica armazenada no tórax. Assim, a respiração é realizada pela diferença de pressão criada nas vias aéreas superiores (pressão atmosférica) e os alvéolos (pressão intratorácica) (CUNNINGHAM, 2014).



O processo de respiração ocorre de forma automática e rítmica, coordenada pelo sistema nervoso central. Em casos diversos, que podem variar desde consequências de anestésias até falência mecânica do sistema respiratório, torna-se necessária a introdução da ventilação mecânica para auxiliar o organismo a realizar todo o processo de respiração (CAVALCANTE, 2020).

2.2 VENTILAÇÃO MECÂNICA

O primeiro ventilador mecânico a ser usado com êxito foi o pulmão de aço, em meio a epidemia de poliomielite, em 1926, que funcionava por meio de pressão e um método não invasivo, mas de pouco conforto ao paciente (CASTRO, M., 2011). Conforme o avanço de modelos, adotou-se o método invasivo, com a introdução de um tubo pela boca ou nariz até a traquéia, com o uso mais comum por anestesistas (ROMERO, 2006). A área de utilização de ventiladores mecânicos se estendeu e, na década de 1980, com microcontroladores, foi possível atingir um nível mais alto de sincronia entre paciente e ventilador e a escolha de diferentes modos ventilatórios (CASTRO, M., 2011).

A ventilação mecânica (VM) agora é uma maneira de oferecer suporte ventilatório para pacientes com insuficiência respiratória aguda ou crônica agudizada. Dentro dos objetivos da utilização da VM, estão: correção da hipoxemia e acidose respiratória devido ao aumento da pressão parcial de CO_2 (PCO_2) no sangue; aliviar o esforço necessário pela musculatura respiratória para manter o organismo em situações de grande demanda metabólica; reduzir o consumo de oxigênio, aumentando o conforto e permitindo a execução de terapêuticas específicas (CARVALHO; TOUFEN; FRANCA, 2007). Tendo em vista a imprescindibilidade de ventiladores mecânicos para esses tratamentos, segundo a legislação brasileira, é obrigatória a presença desses aparelhos em clínicas veterinárias. (CONSELHO NACIONAL DE MEDICINA VETERINÁRIA, 2013).

2.2.1 Princípios

Os ventiladores mecânicos insuflam as vias respiratórias com um volume corrente (V_t), onde o movimento pode ocorrer por: diferença de pressão criada nas vias aéreas superiores e os alvéolos, que se encontram sob menor pressão (ventilação por pressão negativa); aumento da pressão nas vias aéreas superiores (ventilação por pressão positiva). A ventilação por pressão positiva é mais comum.

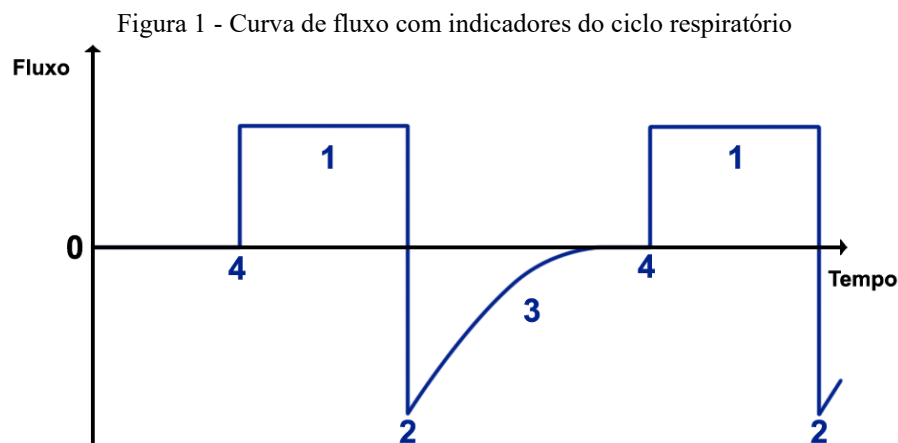
No ar fornecido ao paciente, pode ser controlada a concentração de O_2 (FIO_2 - Fração de Oxigênio Inspirada), necessária para atingir a taxa arterial de oxigênio (PaO_2 - Pressão parcial de oxigênio no sangue arterial) que é preciso. Também pode ser controlada a velocidade em que o ar é administrado (fluxo inspiratório). Pode ainda ser definida a forma de onda do fluxo (descendente, quadrada, ascendente ou sinusoidal), escolhida de acordo com a situação.



A frequência respiratória (F) é dada pelo número de ciclos ventilatórios que ocorrem durante um minuto, e é consequência do tempo inspiratório (T_i), o qual depende do fluxo e do tempo expiratório (T_e) (CARVALHO; TOUFEN; FRANCA, 2007).

2.2.2 Ciclo ventilatório

É chamado de ciclo ventilatório o período do início de uma inspiração até o disparo do ventilador. De acordo com a Figura 1, há 4 períodos que descrevem o ciclo ventilatório (CARVALHO; TOUFEN; FRANCA, 2007):



Fonte: Os autores, 2023..

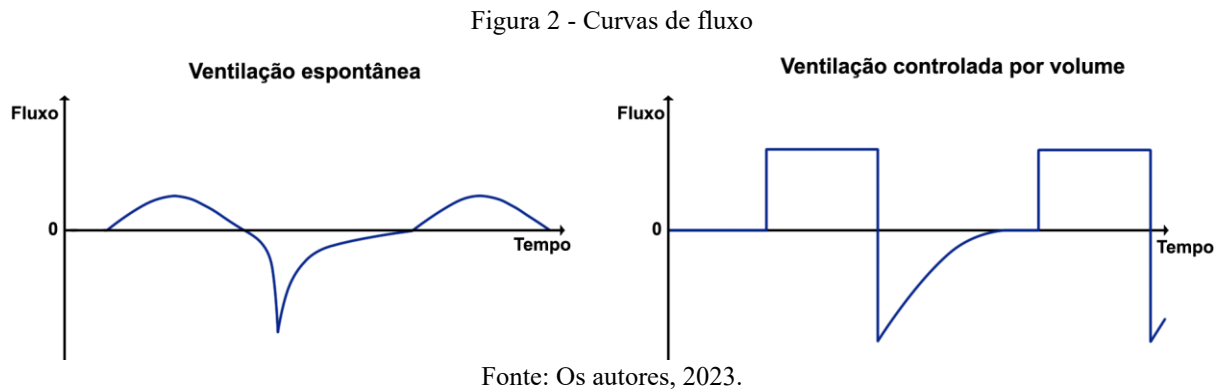
- 1 - Fase inspiratória: período que se inicia quando a válvula respiratória do ventilador mecânico é aberta, permitindo a insuflação dos pulmões de acordo com sua elasticidade.
- 2 - Mudança de Fase (ciclagem): momento em que ocorre a transição entre a fase inspiratória e a fase expiratória.
- 3 - Fase expiratória: ocorre após o fechamento da válvula respiratória, seguido da imediata abertura da válvula expiratória. Permite a saída do volume de ar introduzido no paciente de forma natural até que atinja a pressão intratorácica igual à do ambiente ou um valor determinado pelo ventilador, chamado PEEP (*Positive End Expiratory Pressure*), que impede o colapso dos alvéolos do paciente por esvaziamento completo do pulmão.
- 4 - Mudança de fase expiratória para fase inspiratória (disparo): terminada a fase expiratória, ocorre o disparo do ventilador, abrindo a válvula inspiratória e voltando a fase inspiratória, iniciando um novo ciclo.

2.2.3 Análise Gráfica

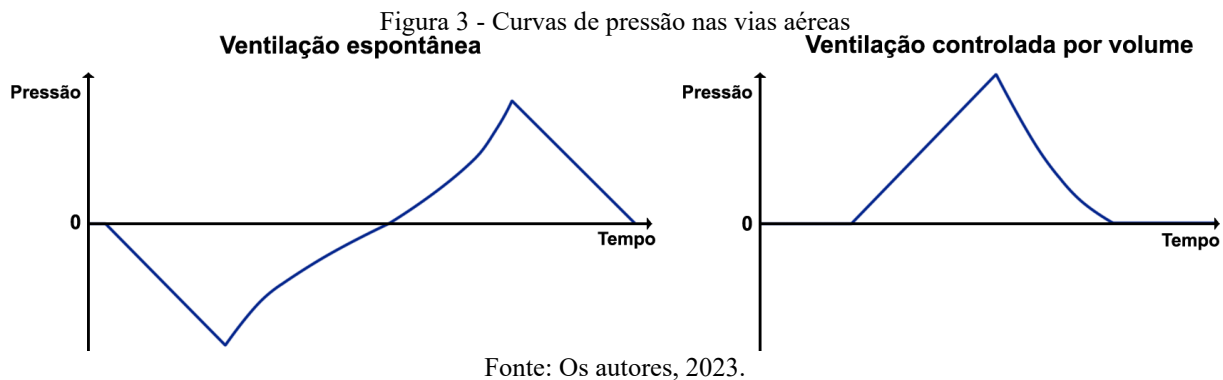
Curva de fluxo. O fluxo é medido por sensores de pressão presentes no ventilador. É iniciado de acordo com as configurações do aparelho, após um disparo, até alcançar o pico de fluxo. A forma



de onda mais utilizada é a quadrada, como mostra a Figura 2, que representa o fluxo em ventilação espontânea e controlada:



Curva de pressão. Como observado na Figura 3, à medida que o fluxo de ar adentra o sistema respiratório, a pressão inspiratória vai se elevando, pois é necessária para vencer dois componentes: um resistivo (devido à resistência ao fluxo de ar passando pelas vias aéreas) e outro elástico (decorrente da distensão dos pulmões e da parede torácica).



Na ventilação espontânea, ocorre uma queda de pressão nos alvéolos devido a contração da musculatura, o que não ocorre de forma significativa na ventilação mecânica, visto que o ventilador pode suprir a necessidade do paciente de realizar um esforço.

Curva de volume. A curva de volume mostra o volume inspirado pelo paciente de acordo com os parâmetros ajustados. Seus valores máximos e mínimos devem ser constantes, do contrário pode indicar vazamentos, aprisionamento aéreo ou desconexão do circuito.

2.3 PRINCIPAIS INDICAÇÕES PARA O USO DA VENTILAÇÃO MECÂNICA

Os critérios de aplicação da ventilação mecânica dependem completamente dos objetivos que se quer alcançar. Porém, em casos mais críticos, nos quais não há tempo para tais avaliações da função



respiratória, a impressão clínica é o que mais deve ser levado em consideração para o início do tratamento. As principais indicações são:

- Reanimação devido à parada cardiorrespiratória;
- Hipoventilação e apnéia;
- Insuficiência respiratória devido à doença pulmonar intrínseca e hipoxemia;
- Falência mecânica do aparelho respiratório (fraqueza muscular, doenças neuromusculares, paralisia, comando respiratório instável, entre outros casos);
- Prevenção de complicações respiratórias;
- Redução do trabalho muscular respiratório e fadiga muscular.

Em suma, o uso da VM é indicado quando o paciente não consegue atingir os níveis recomendados de O₂ e CO₂ no sangue de forma autônoma (CARVALHO; TOUFEN; FRANCA, 2007).

A VM também é utilizada em pacientes submetidos à anestesia geral, para que as funções respiratórias não sejam comprometidas durante uma intervenção cirúrgica. É essencial em tal caso, se forem utilizados hipnoanalgésicos, hipnóticos ou agentes anestésicos inalatórios em doses suficientemente elevadas para que haja interferência no ciclo respiratório espontâneo (DUGDALE, 2007). Da mesma forma, é imprescindível quando há indicação da administração de agentes bloqueadores neuromusculares para obter um relaxamento muscular que facilite o acesso cirúrgico.

Outras indicações incluem o hiperadrenocorticismismo, no qual os animais podem apresentar fraqueza muscular generalizada pela redução das bombas de sódio e potássio no músculo esquelético. Essa depressão respiratória pode ainda ser intensificada quando há hepatomegalia, acúmulo de gordura abdominal que se opõe aos movimentos respiratórios e aumenta a pressão diafragmática levando o animal a uma depressão respiratória (LEAL, 2008 *apud* CASTRO, M., 2011).

2.4 MODOS VENTILATÓRIOS RELEVANTES

Diferentes modos de operação do ventilador mecânico podem ser aplicados e configurados para atender da melhor forma possível as necessidades do paciente, desde fases críticas até a recuperação.

2.4.1 Ventilação mandatória contínua com pressão controlada – modo controlado

É fixada a frequência respiratória, o tempo inspiratório ou a relação inspiração:expiração (Ti/Te), e o limite de pressão inspiratória. O disparo ocorre de acordo com a frequência respiratória, mas a ciclagem ocorre em função do tempo inspiratório ou relação Ti/Te. O V_t depende da pressão inspiratória pré-estabelecida e da impedância do sistema respiratório em questão, junto do tempo inspiratório selecionado pelo operador (CARVALHO; TOUFEN; FRANCA, 2007).



2.4.2 Ventilação mandatória contínua com pressão controlada – modo assistido-controlado

Diferente do modo controlado, o modo assistido-controlado dá certa autonomia ao paciente, o qual pode determinar, até certa dimensão, o momento do disparo do ventilador. Através de um esforço para respirar, o paciente pode criar uma diferença suficiente de pressão, determinada pelo operador, para que o ventilador inicie um disparo. O V_t obtido também irá depender desse esforço. Caso não haja esforço por parte do paciente, o ventilador irá operar de acordo com a frequência respiratória determinada (CARVALHO; TOUFEN; FRANCA, 2007).

2.5 MATLAB

MATLAB é um *software* desenvolvido pela MathWorks, conhecido por sua versatilidade em diversas aplicações que envolvem cálculos numéricos e desenvolvimento de matrizes. Possuindo linguagem de programação própria, é um ambiente voltado para análises iterativas de dados, desenvolvimento de processos, visualização e desenvolvimento de algoritmos, entre outros. Através do MATLAB, é possível criar interfaces customizáveis para controle e processamento de dados recebidos em tempo real, possibilitando a plotagem e visualização de gráficos, manipulação de matrizes e diversas tarefas, que podem ser controladas por um operador ou não. A disponibilização de diversos pacotes de ferramentas e bibliotecas de códigos fornecem funções que podem resolver problemas diversos encontrados em um processo (MATHWORKS, 2023).

2.5.1 App Designer

O App Designer é uma ferramenta integrada do MATLAB que permite criar interfaces de usuário gráficas (do inglês, *Graphical User Interface - GUI*) através de funções como “clica e arrasta” de diversos elementos visuais que podem ser exibidos numa tela que dispõe de diversas opções de edição. Essa ferramenta contribui para a criação de aplicativos interativos sem precisar de conhecimentos extensos na área de desenvolvimento de *software*. Ainda assim, as funções de cada elemento em tela, os alarmes e a comunicação precisam ser programados. Os aplicativos feitos nessa plataforma podem ser exportados e executados em dispositivos sem instalação do MATLAB, já que este requer uma licença paga, mas apenas com o MATLAB Runtime, que é gratuito. Assim, a edição e programação dependem de uma licença, mas a execução do programa não.

2.6 ARDUINO

Arduino é uma plataforma microcontrolada, que objetiva a prototipagem de circuitos de baixo custo através de uma interface simples, desenvolvida para comunicar-se facilmente com microcontroladores Atmel. Sua programação é feita numa linguagem variante de C/C++ e a plataforma possui uma IDE de código aberto que auxilia os usuários a projetar suas aplicações. Seus empregos



vão desde um simples acender e apagar de LEDs até projetos sofisticados que dependem apenas do conhecimento de Arduino e eletrônica (SILVA; ARAUJO; CAVALCANTE., 2019).

A placa Arduino Uno, uma das variantes da família Arduino, possui 14 pinos digitais, dos quais 6 são capazes de emitir sinal PWM, 6 entradas analógicas, porta USB, processador de 16 MHz e um botão de reset. Sua tensão nominal é de 5 V, com tensão de entrada que varia de 7 a 12 V, num limite de 6 a 20 V, e emite 20 mA em pinos GPIO em corrente contínua (CC). Sua massa é de 25 g (ARDUINO, 2023).

2.7 SENSOR DE PRESSÃO DIFERENCIAL MPS20N0040D-D

O sensor de pressão MPS20N0040D-D é um sensor do tipo de estado sólido de tecnologia microeletromecânica, de alta precisão, baixo custo e fácil uso. Suas leituras alcançam até 40 kPa e é alimentado por 5 V e 1 mA em CC. A impedância de entrada é de 4 a 6 Ω e sua saída varia de 50 a 100 mV (ADI; KITAGAWA, 2019).

2.8 VÁLVULA TIPO ESFERA

Válvulas são dispositivos mecânicos que controlam o fluxo de fluidos em tubulações, com aplicações industriais, sistemas hidráulicos e controle de posicionamento em robótica. Ao contrário de válvulas convencionais, a válvula de tipo esfera controla a vazão de forma proporcional a um sinal de controle através de um obturador esférico com um intervalo de operação de 90°. Isso permite ajustes precisos na pressão ou vazão do fluido. Essas válvulas são de baixo custo e excelentes para controle proporcional, com a desvantagem de não terem uma vazão linear em relação ao sinal de controle (SILVA, H., 2019).

2.9 CONTROLADOR DO MOTOR

Sua função é gerar um sinal de modulação por largura de pulso (do inglês, *Pulse Width Modulation* - PWM) para o motor. Esse sinal é proveniente da alimentação em corrente contínua (CC), e é transformado em corrente, assim acionando as fases do servo motor (SOKIRA; JAFFE, 1990 *apud* DINAU, 2009). Suas principais características são alta precisão e alto desempenho, que se devem à operação em malha fechada, que ocorre de acordo com a realimentação da posição indicada pelo sensor acoplada ao eixo (CASTRO, F., 2016).

A modulação PWM faz com que o valor médio da tensão possa ser alterado a cada enrolamento do estator, durante a sequência de comutação (SOKIRA; JAFFE, 1990 *apud* DINAU, 2009), fazendo com que a tensão máxima seja ligada e desligada para cada fase, de tal maneira que a tensão média seja correspondente à indicada no código do microcontrolador. Algumas das vantagens de se utilizar PWM para controlar a velocidade de um motor são (WIBERG, 2003 *apud* DINAU, 2009): grande



praticidade na associação com um microcontrolador, inexistência de perda de potência na bateria, caso utilizado em velocidades baixas e torque maior no motor, quando a utilização da tensão máxima ocorre em curtos intervalos de tempo.

O circuito integrado PCA9685 é uma implementação de um controlador PWM que possui 16 canais, possibilitando a conexão com diversos servo motores. A comunicação é do tipo I2C e tem alta precisão e eficiência. O controlador possui dois bornes que permitem a alimentação externa dos servos motores, para evitar prejudicar todo o circuito a qual está conectado e proteger contra inversão de polaridade.

2.10 SERVO MOTOR

Os motores CC, ao receberem tensão, produzem a força magnetomotriz, que transforma a energia elétrica em energia mecânica. Seu principal objetivo tem ligação com o controle de velocidade associado ao torque (SIMONE; NASCIMENTO JUNIOR, 2000, 2006 *apud* CASTRO, F., 2016). O controle da velocidade dos servo motores está diretamente ligado à quantidade de tensão aplicada (CASTRO, F., 2016).

O modelo MG995 opera em CC, possui escovas, alta velocidade e não depende de códigos para seu funcionamento. Ele atua num intervalo de 120°, percorrendo 60° em 0.2 s com 8.5 kgf/cm alimentado por 4.8 V e 60° em 0.16 s com 10 kgf/cm a 6 V. Suporta uma tensão máxima de 7.2 V e opera em temperaturas de 0 a 55 °C. Seus mecanismos de metal promovem mais vida útil e é resistente a choques, constando de roletes de duas carreiras (MG995, 2023).

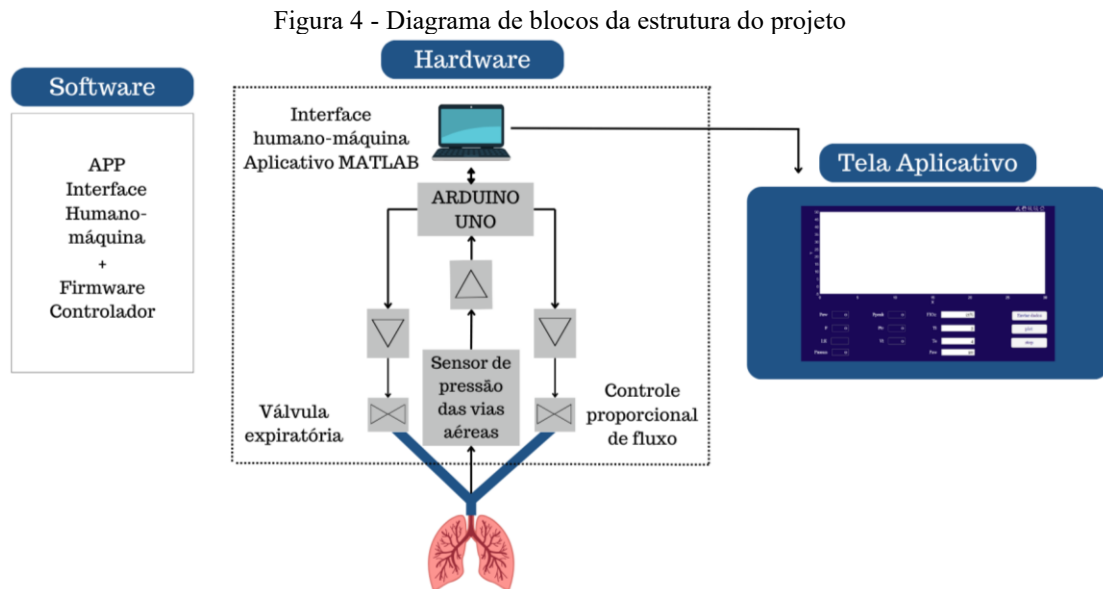
3 MATERIAIS E MÉTODOS

Após o levantamento de informações sobre ventiladores pulmonares em animais de pequeno porte, foi iniciada a construção de um protótipo de ventilador pulmonar capaz de prover ventilação por pressão controlada (PCV) em modo controlado, por ser este o modo ventilatório mais comumente utilizado e pela possibilidade de sua implementação sem a utilização de sensores de vazão, mas apenas com sensores de pressão. Sensores de pressão são mais simples, mais baratos e mais fáceis de serem adquiridos.

A Figura 4 ilustra a estrutura proposta para o protótipo. As setas indicam a direção do fluxo de informação e comandos de controle, genericamente, do sensor para o controlador e do controlador para as válvulas. A comunicação entre o controlador e o dispositivo de interface, que será abordada com mais detalhes à frente, é bidirecional, de modo que os comandos e ajustes dos parâmetros feitos pelo operador são enviados ao controlador, que, por sua vez, deve responder de acordo com a nova configuração. O controlador e a IHM trocam informações de monitoramento, diagnóstico e alarmes relacionados ao processo. Os triângulos nas portas do controlador representam condicionadores de



sinal. A válvula expiratória deve apenas assumir os estados fechado e totalmente aberto, permitindo, neste último caso, que a compressão natural da caixa torácica do paciente expulse o ar. A válvula inspiratória, entretanto, deve abrir parcialmente para que o volume a ser injetado durante o período de inspiração corresponda ao fixado pelo operador. O pico de pressão inspiratória, o nível de pressão para disparo de ciclo assistido e o tempo inspiratório, bem como a frequência respiratória, também são parâmetros ajustáveis manualmente, e o nível de abertura da válvula é calculado pelo sistema.



Fonte: Os autores, 2023.

Para o desenvolvimento do controlador em malha fechada do ventilador mecânico, foi escolhido o Arduino Uno. Seu custo acessível, vasta documentação e simplicidade de uso tornam a plataforma atrativa para projetos didáticos envolvendo projetos pedagógicos. Além disso, ele possui algumas portas GPIOs (*General Purpose Input/Output*), inclusive analógicas. Isso é essencial para a leitura dos sensores necessários para o funcionamento do protótipo, para o controle de válvulas e outros atuadores, bem como para a transmissão de dados ao aplicativo desenvolvido, que configura a interface. O aplicativo utilizado para a programação do Arduino Uno é a própria Arduino IDE.

Antes da escolha do Arduino Uno, um ESP-32 estava sendo utilizado. Mas, sua tensão de entrada e saída de 3,3 V era insuficiente para alimentar o conversor HX711 e representa problemas para a leitura dos sensores, junto da incompatibilidade com demais componentes que operam em 5 V. Além disso, seus exemplos de aplicação foram considerados bastante limitados, quando comparados aos disponíveis para placas do tipo Arduino.

Para o desenvolvimento de uma GUI, o MATLAB foi o programa escolhido. Inicialmente, o *software* Blynk foi considerado, mas funções essenciais para o VentiVet estavam fora de alcance na versão gratuita. Além do mais, a transmissão de informações ocorria por um servidor na nuvem,



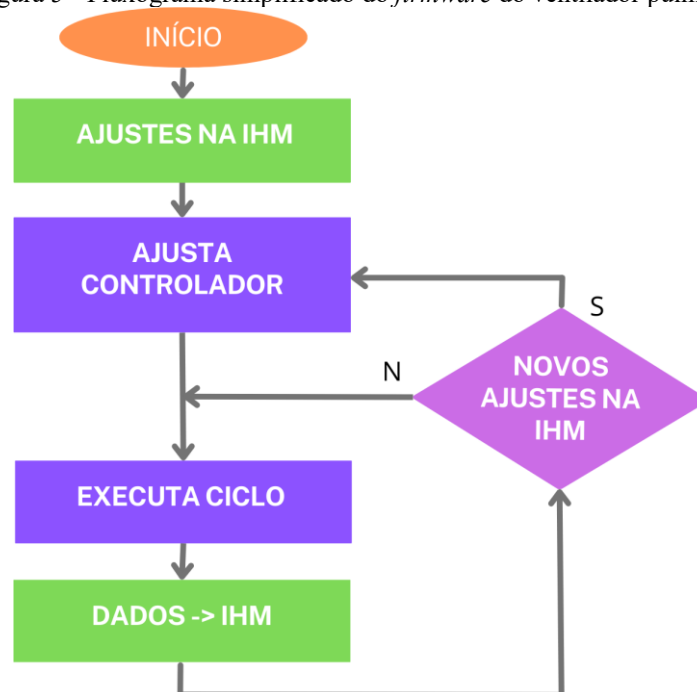
tornando o tempo de resposta uma preocupação e a dependência de conexão permanente com a Internet traria riscos de falhas críticas no processo.

Como segunda opção, através do MATLAB, foi criada uma interface para a operação do ventilador veterinário, com entradas numéricas e botões com opções de controle de diversos parâmetros, além de exibição, através de gráficos, dos dados coletados pelos sensores do ventilador. Os dados são transmitidos do Arduino Uno para o MATLAB através de conexão serial e a sincronização dos processos é feita por meio de interrupções de *software*. O aplicativo pode ser compilado e exportado para ser executado em dispositivos que não tenham o *software* de desenvolvimento do MATLAB instalado, mas apenas o pacote gratuito MATLAB Runtime.

Na Figura 5 e na Figura 6, podem ser vistos fluxogramas simplificados do *firmware* embarcado no Arduino Uno e do *software* desenvolvido para a IHM, respectivamente.

No fluxograma do controlador, é evidenciado que os parâmetros de controle são recebidos da IHM após ajustes feitos pelo usuário. Na sequência, o controlador em malha fechada implementa as fases inspiratória e expiratória do ciclo respiratório, monitorando o sensor de pressão e atuando sobre as válvulas proporcional e expiratória. Ao longo de todo o ciclo, o controlador envia dados para a IHM. O ventilador mudará suas configurações somente quando o botão “Enviar dados” for pressionado.

Figura 5 - Fluxograma simplificado do *firmware* do ventilador pulmonar

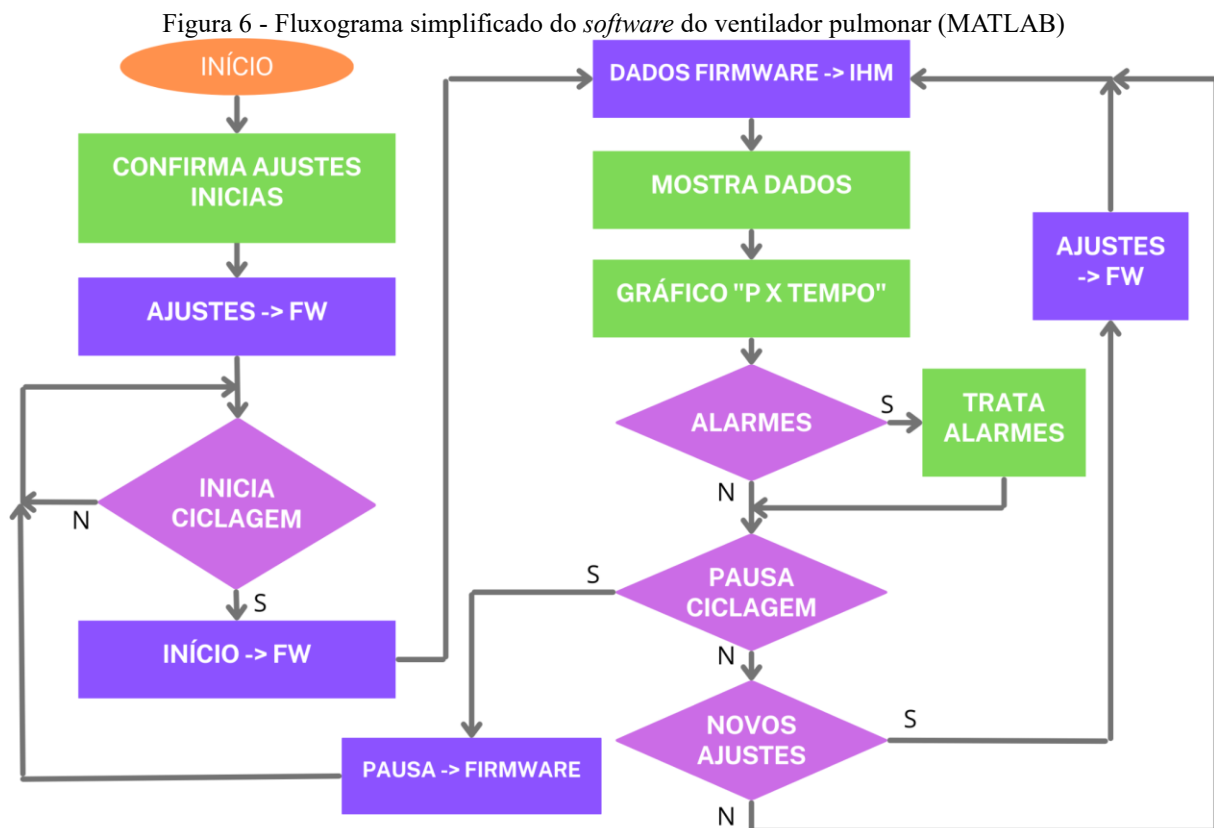


Fonte: Os autores, 2023.

Como exibido no fluxograma da IHM (Figura 6), o operador precisa confirmar os ajustes iniciais antes de iniciar a operação do controlador. Todos os parâmetros de ciclagem são enviados ao *firmware*, que inicia e controla o ciclo em si, enviando periodicamente dados para serem mostrados na IHM. Adicionalmente aos valores numéricos, os valores medidos da pressão das vias aéreas são



apresentados em um gráfico com a evolução do parâmetro nos últimos 30 segundos. Caso haja alarmes disparados no ciclo, a IHM aciona os dispositivos e mostradores correspondentes para alertar o operador. Caso o operador solicite a interrupção do ciclo respiratório, o *software* envia dados para o *firmware*, que interrompe o ciclo até que seja solicitado seu reinício. Eventuais novos parâmetros informados pelo operador são enviados ao controlador para sua adequação.



Fonte: Os autores, 2023.

Os fluxogramas ilustram como o *firmware* e o *software* operam de maneira independente, porém relacionada. Caso haja problemas na comunicação serial, o *firmware* controlador do Arduino Uno segue ventilando o paciente conforme os últimos ajustes.

A fim de garantir uma cadência constante nas operações de leitura de sensores e atuação do controle, são utilizadas interrupções do controlador, de modo a priorizar essas ações. Um temporizador que funciona paralelamente ao *firmware* do controlador acionará uma interrupção a cada 10 ms, que suspenderá estados menos prioritários, como a comunicação com a IHM e operações aritméticas de cálculos do algoritmo de controle proporcional. Este ritmo constante é importante para manter a frequência de amostragem estável, tornando o sistema de controle invariante no tempo, condição fora da qual o comportamento do controle efetivamente realizado diferiria do algoritmo calculado. Este, por sua vez, consistirá de um controlador proporcional que terá como alvo o valor de pressão das vias aéreas superiores, determinado pelo operador. Esse valor de pressão alvo é comparado com a pressão medida pelo sensor. Com base nesta diferença, é calculado o nível de abertura da válvula inspiratória.



3.1 PARÂMETROS

Nos tópicos a seguir, serão abordados os parâmetros envolvidos no processo de ventilação, que são relevantes para o operador, funcionamento do protótipo e comunicação entre dispositivos. A IHM teve inspiração no protótipo de um ventilador mecânico desenvolvido pela Universidade Tecnológica Federal do Paraná (UTFPR). A interface do mesmo, feita pela Universidade, está disponível na Figura 7.

Figura 7 - Interface do ventilador mecânico desenvolvido pela UTFPR



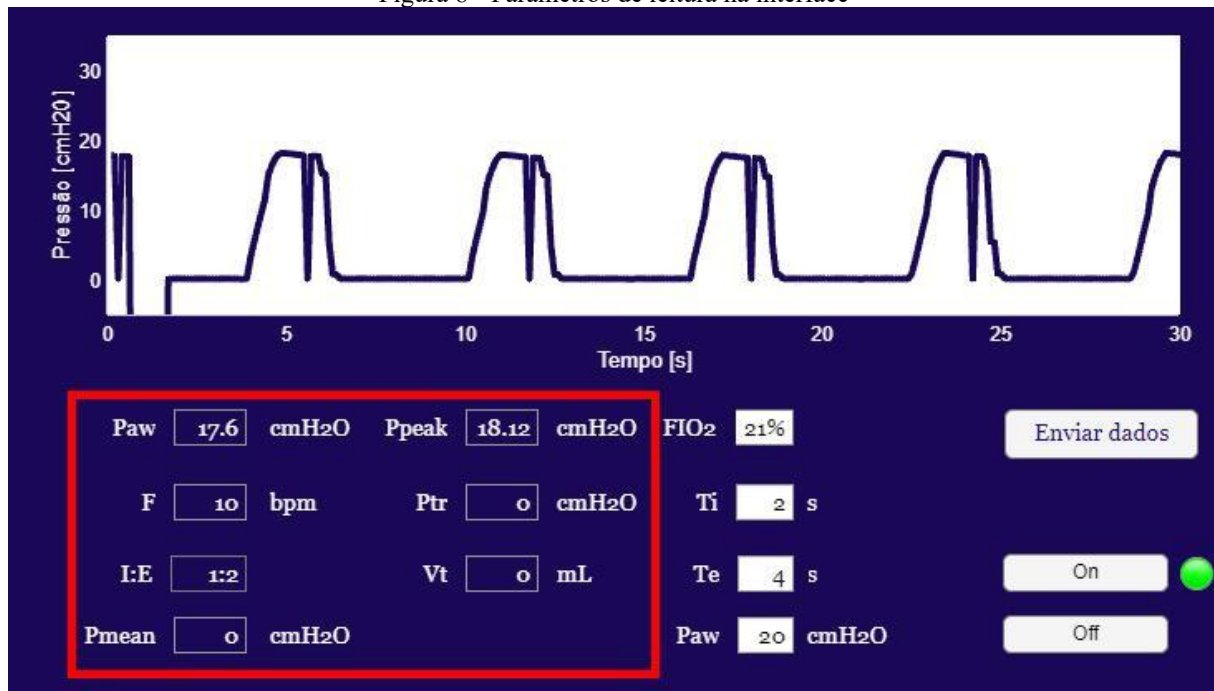
Fonte: UTFPR, 2020.

3.1.1 Parâmetros de leitura

Os parâmetros não-editáveis consistem em caixas numéricas, que não estão relacionadas a qualquer alteração de configuração do processo respiratório, mas sim à leitura de suas grandezas físicas. A Figura 8 demonstra um exemplo de como estes parâmetros podem ser visualizados com o protótipo em funcionamento.



Figura 8 - Parâmetros de leitura na interface



Fonte: Os autores, 2023

Existem quatro parâmetros de leitura funcionais presentes na interface. São eles:

Paw (Airway Pressure - Pressão nas vias aéreas). Esse parâmetro está ligado à leitura do sensor de pressão, sendo alterado em tempo real, e expresso em centímetros de água [cmH₂O], de acordo com a variação da pressão dentro do pulmão-teste, o qual traz grande similaridade com um pulmão de um ser vivo, já que a sua pressão interna também seria alterada proporcionalmente ao funcionamento do ventilador.

F (frequência respiratória). a frequência corresponde ao número de ciclos respiratórios enviados do VentiVet ao paciente por minuto, e está diretamente ligada ao valor de Ti e Te, valores estes que podem ser editados pelo usuário. A frequência é expressa em bpm - *breaths per minute* (respirações por minuto).

Como mencionado anteriormente, a frequência não pode ser desassociada do tempo inspiratório e expiratório. Com isso, o cálculo das respirações por minuto é feito da seguinte forma:

- Leitura de Ti[s] e Te[s] após o pressionamento do botão “enviar dados”;
- soma dos valores lidos (pois, um ciclo respiratório corresponde ao início do tempo inspiratório até o final do tempo expiratório);
- divisão de 60 (número de segundos em um minuto) pelo valor gerado após a soma;
- disposição do valor gerado na caixa numérica.

I:E (relação inspiração-expiração). assim como a frequência, a I:E também está intrinsecamente ligada aos valores de Ti e Te. O funcionamento dessa leitura se dá da seguinte maneira: primeiramente, é necessária uma alteração nos valores de Ti e Te, pois estes iniciam-se zerados. Após isso, há uma leitura destes valores e o maior divisor comum dos mesmos é calculado pela função *gcd*

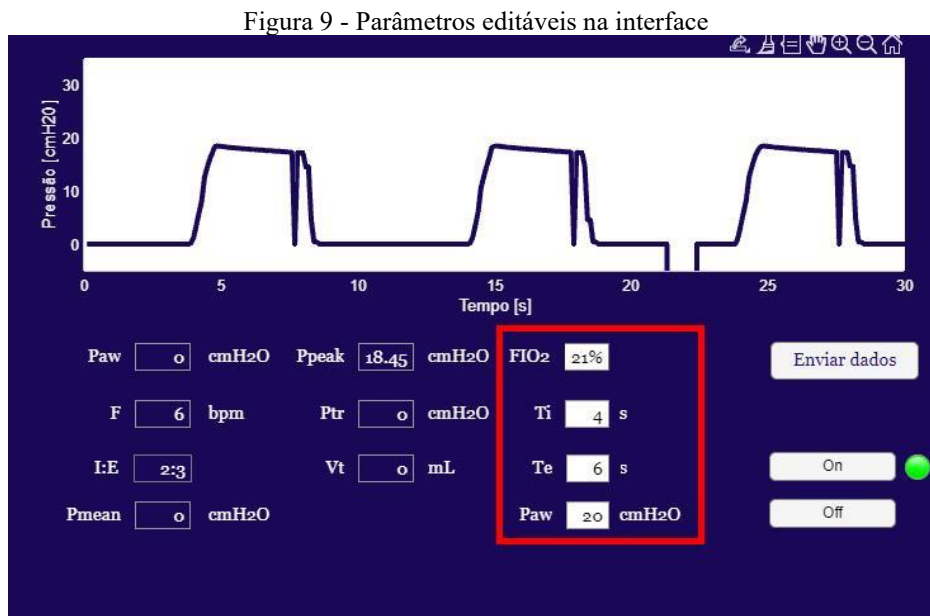


(*greatest common divisor*), para que a razão esteja em sua forma mais simplificada. Desta forma, T_i e T_e são divididos, separadamente, pelo maior divisor comum, e as variáveis são transformadas em cadeias de caracteres (em inglês, *strings*). Assim, a razão I:E pode finalmente ser exibida na IHM.

Ppeak (pressão de pico). o valor expresso nesse parâmetro é referente ao maior nível de pressão atingido naquele ciclo respiratório e é resetado a cada início de ciclo. Para realizar a leitura no momento correto e desenvolver uma lógica ideal no algoritmo, é preciso da identificação imediata de em que fase o ciclo respiratório está. Para isso, foi adicionada uma informação a mais nas mensagens enviadas pelo Arduino: se o processo se encontra na fase expiratória ou inspiratória. Esta grandeza é expressa em centímetros de água (cmH₂O).

3.1.2 Parâmetros editáveis

Os parâmetros editáveis, como os de leitura, também são caixas numéricas, com o diferencial de que seus valores podem ser editados pelo operador. A alteração desses valores permite alterar diretamente a ciclagem. O retângulo vermelho na Figura 9 destaca a localização desses elementos na interface:



Fonte: Os autores, 2023

T_i e T_e (tempo expiratório e tempo inspiratório). A determinação dos valores inspiratório e expiratório, em segundos [s], juntamente com o objetivo de pressão, são base para o funcionamento do protótipo. O funcionamento de tais parâmetros ocorre de acordo com o código do Arduino e sua alteração é feita diretamente pelo MATLAB, que faz a comunicação com o microcontrolador.

Paw (versão editável). É utilizado para determinar o objetivo de pressão nas vias aéreas, em centímetros de água [cmH₂O]. O servo motor que controla a válvula inspiratória fará ajustes na



abertura da mesma para que o objetivo de pressão seja atingido, conforme a programação do microcontrolador.

3.2 COMUNICAÇÃO ARDUINO-MATLAB

A comunicação entre o Arduino Uno e a interface em MATLAB é realizada por comunicação serial pela porta USB do dispositivo que exibe a interface e o microcontrolador.

Apesar de ser possível escrever e ler informações das portas analógicas e digitais do Arduino através do MATLAB por algoritmos, a comunicação era limitada a isso, sem possibilidades de alterar o *firmware*, sendo insuficiente para realizar as aplicações desejadas. Para eliminar esse obstáculo, foi desenvolvido um protocolo de comunicação inspirado no protocolo Modbus, comum na integração de redes industriais e amplamente utilizado na automação industrial.

A comunicação pode ser considerada a realização mais desafiadora desenvolvida neste projeto, pois não foram encontrados exemplos que utilizassem métodos simples e havia dúvidas sobre a possibilidade de alcance de uma comunicação ideal para o protótipo. Uma das formas de envio e recebimento de informações entre o MATLAB e o Arduino é através da criação de um arquivo de texto com a utilização da função “fopen”, no código executado no MATLAB, que escreve as informações num arquivo de texto que precisa ser acessado constante. Porém, isso não apresenta a mesma eficiência que o protocolo de comunicação desenvolvido.

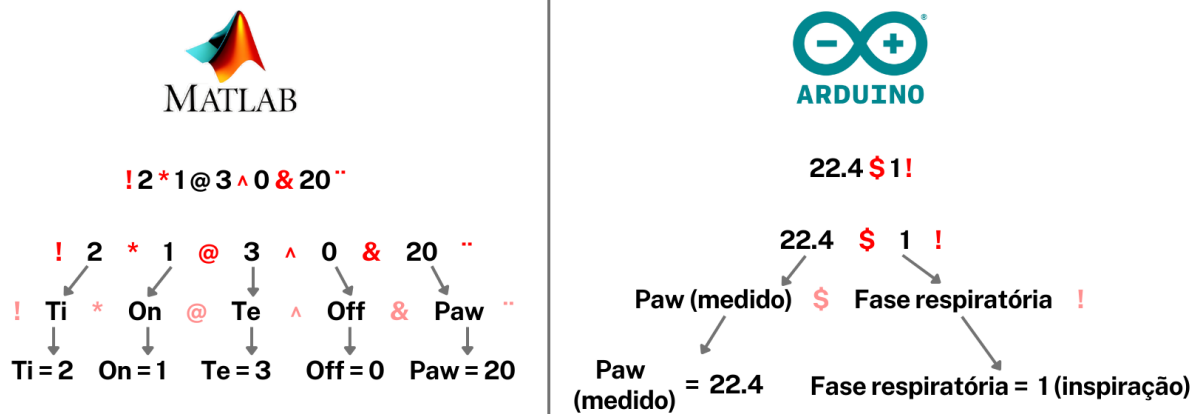
O protocolo criado consiste no envio e recebimento de mensagens entre o algoritmo em MATLAB e o Arduino. Essas mensagens são puramente *strings* contendo as informações e parâmetros necessários para controlar e monitorar o processo de respiração. Segue um exemplo real de mensagens enviadas do MATLAB para o Arduino e o contrário, respectivamente:

- MATLAB para Arduino Uno: !2*1@3^0&20"
- Arduino Uno para MATLAB: 22.4\$1!

Na Figura 10, são demonstradas visualmente a separação das mensagens e a identificação dos parâmetros presentes em cada uma.



Figura 10 - Diagrama exemplificando a tradução das mensagens



O princípio para ambos os algoritmos é o mesmo: concatenar informações recolhidas do processo pelo qual é responsável e enviar como um *string* para o outro. Nesse *string*, as informações são separadas por caracteres especiais.

Apesar de alcançar o mesmo resultado e ter o mesmo início, os processos de interpretação em cada algoritmo são diferentes. No Arduino, há funções predefinidas que facilitaram o trabalho significativamente, permitindo a separação das informações no *string* a partir da delimitação de caracteres especiais diferentes¹, como exemplificado na linha única de código a seguir:

```
New_InspP_Tg = Received_Info.substring(Received_Info.indexOf('&') + 1,  
Received_Info.indexOf("")).toInt();
```

A informação que carrega o objetivo de pressão, definido pela interface, que está entre os caracteres “&” e “””, é extraída do *string* e salva numa nova variável que será comparada com o objetivo de pressão atual e então atualizada. A decodificação dos dados não precisa ser feita em uma ordem específica.

No MATLAB, as mensagens são separadas com apenas um único caractere especial, e um segundo caractere especial indica o fim da mensagem². O *string* é quebrado em partes a partir da localização desses caracteres, e as informações são armazenadas numa matriz. Os valores referentes a cada informação enviada são acessados através da indexação pré-definida dessa matriz, a qual é atualizada a cada 100ms, que é a taxa de envio configurada na interrupção de *software* do *timer* (temporizador) do Arduino.

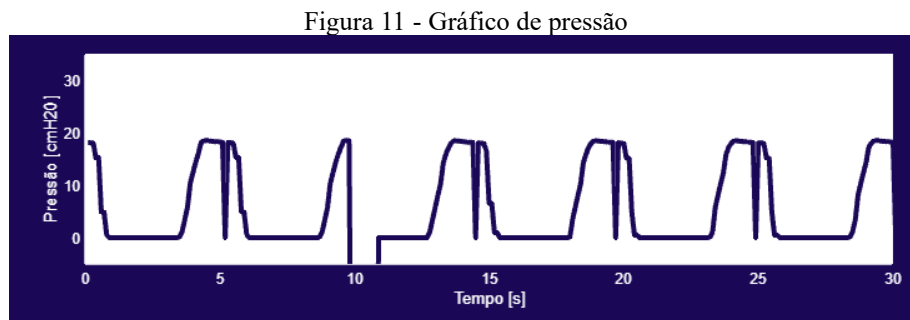
¹ Os caracteres especiais podem ser alterados livremente, desde que ambos os programas estejam configurados de acordo. Na versão descrita neste texto, a única limitação encontrada é a utilização de caracteres iguais.

² No caso do código em MATLAB apresentado, o \$ (cifrão) foi escolhido para separar as informações, e o caractere ! (ponto de exclamação) para sinalizar o fim da mensagem enviada.



3.3 GRÁFICO DE PRESSÃO

A construção do gráfico (Figura 11) na interface é feita a partir da leitura dos valores armazenados numa matriz que é utilizada para plotar o gráfico.



Fonte: Os autores (2023)

O gráfico é atualizado a cada 100ms e seu eixo horizontal está numa escala de 30s, com o eixo vertical exibindo a pressão tomada no momento em centímetros de água (cmH₂O). A tomada de pressão é realizada pelo sensor de pressão HX711, o qual é calibrado automaticamente para ter sempre a pressão inicial como referencial nulo.

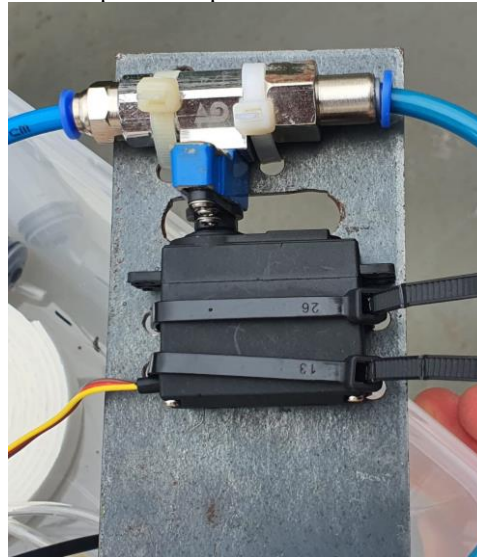
3.4 ABERTURA E FECHAMENTO DE VÁLVULAS

As válvulas inspiratória e expiratória são ambas do tipo esfera. Apenas a válvula inspiratória consta de um controle proporcional, realizado pelo algoritmo PID configurado no *firmware* do microcontrolador. Não é necessário o mesmo tipo de controle para a válvula expiratória, uma vez que esta irá assumir apenas dois estados: aberta ou fechada, considerando que ainda não há PEEP funcional no protótipo.

As válvulas são acopladas a servo motores do modelo MG995, como exibido na Figura 12. O controle desses servos deve ser preciso, visto que, se seus movimentos forem insuficientes, as válvulas permanecem abertas e demonstram grande risco ao paciente. E, caso seus movimentos seguirem além do necessário, o eixo dos servos irá travar na alavanca da válvula e seus mecanismos internos serão danificados, comprometendo todo o processo.



Figura 12 - Exemplo de acoplamento do servo motor à válvula



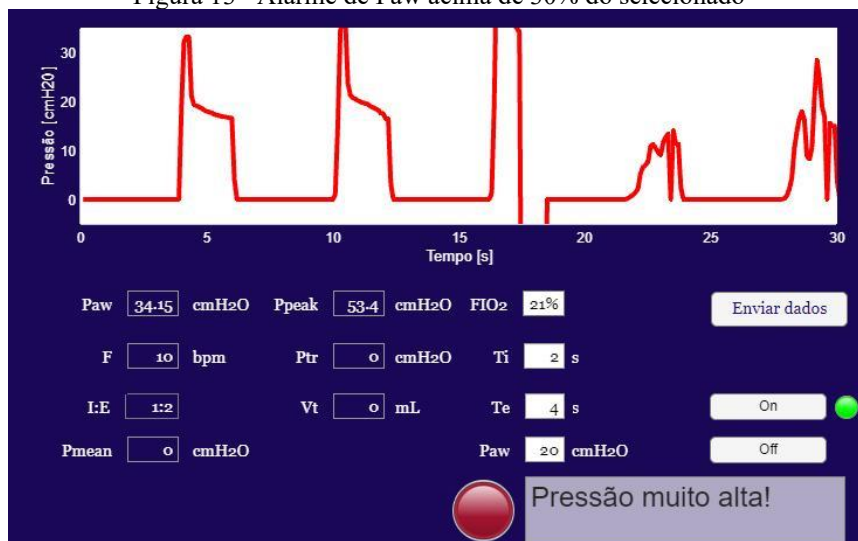
Fonte: Os autores, 2023

Os servos recebem sinal PWM a partir do driver PWM PCA9685, que por sua vez é controlado pelo *firmware*, configurado de acordo com os ajustes na IHM. Esse controlador é alimentado através de uma fonte externa de 5 V.

3.5 ALARMES

O protótipo inclui alarmes para identificar irregularidades no processo de respiração e eventualmente solucioná-las. Na versão aqui descrita do protótipo, há a identificação de dois eventos: pressão das vias aéreas muito alta ou muito baixa. Quando essas irregularidades são identificadas, a linha do gráfico assume a cor vermelha e um aviso é exibido no canto da interface, como pode ser observado nas simulações ilustradas nas Figuras 13 e 14. Esses alarmes são ativados quando a Paw está 30% acima ou 30% abaixo do nível de pressão selecionado.

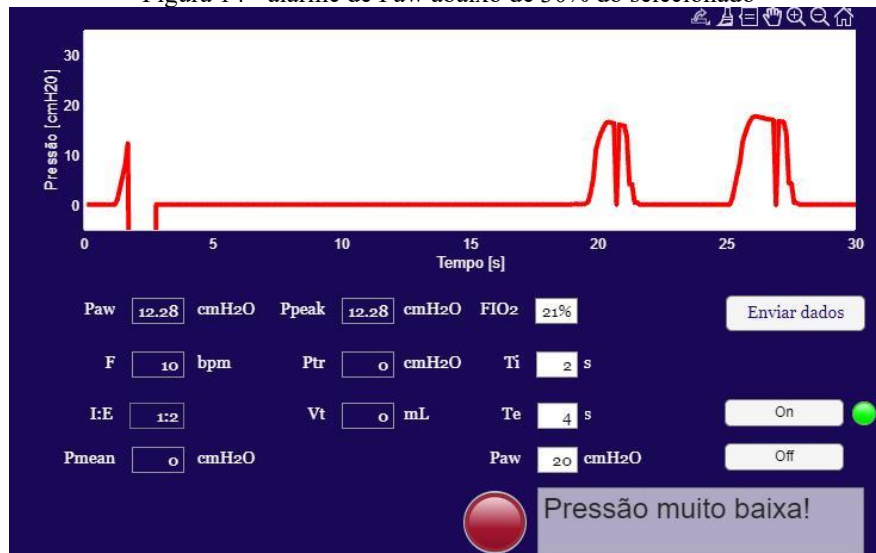
Figura 13 - Alarme de Paw acima de 30% do selecionado



Fonte: Os autores, 2023.



Figura 14 - alarme de Paw abaixo de 30% do selecionado

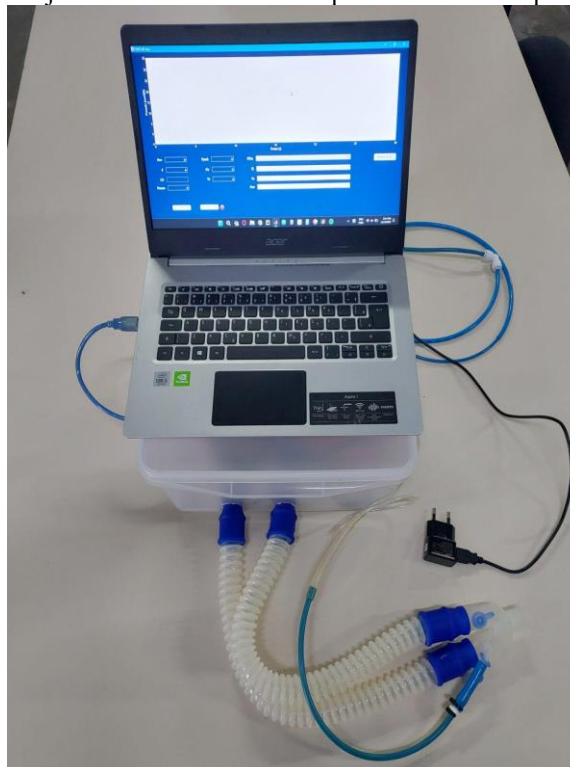


Fonte: Os autores, 2023.

4 RESULTADOS E DISCUSSÃO

O protótipo desenvolvido, exibido na Figura 15, destaca-se pela portabilidade e acessibilidade, principalmente em relação ao custo. Pois, diferente dos aparelhos disponíveis no mercado, o VentiVet pode ter sua interface exibida em qualquer computador, incluindo os que não possuem licença do *software* MATLAB. O ventilador também pode ser transportado de maneira fácil e segura, contando que todos os seus componentes eletrônicos estão aderidos à caixa que os carrega, e o aparelho que exibe a interface é, grande parte das vezes, bastante compacto.

Figura 15 - Conjunto do VentiVet sem os aparelhos de ar comprimido



Fonte: Os autores, 2023



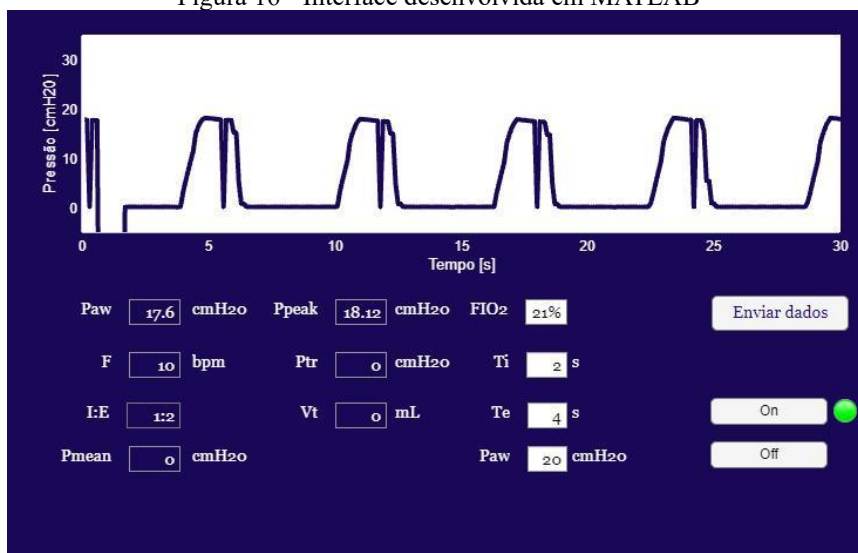
O protocolo de comunicação desenvolvido com a utilização de caracteres especiais como chave de identificação de cada parâmetro se mostrou muito satisfatório na aplicação. Pois associa ambos os códigos utilizados de forma bastante suficiente e rápida, trazendo mais segurança na operação do ventilador.

A configuração das interrupções de *software* no *firmware* inicialmente representavam um problema, uma vez que os *timers* que o Arduino Uno possui são responsáveis por funções além das programadas, interferindo com partes do processo, como a operação de portas analógicas e comunicação serial se alterados.

Para contornar esses problemas, o temporizador *Timer 1* foi utilizado. Um temporizador confiável é indispensável, uma vez que é por meio deste que o gráfico se mantém atualizado em intervalos constantes.

O objetivo de construção de uma interface gráfica para a utilização do protótipo foi concluído com sucesso, como ilustra a Figura 16. O conjunto de botões, parâmetros editáveis, de leitura e gráfico possuem uma dinâmica de interação descomplicada, permitindo que qualquer profissional da área veterinária possa operar o VentiVet, mesmo sem nenhum conhecimento prévio da área técnica que abrange a construção do aparelho. Isso se deve à disposição dos componentes da interface, que visa ser intuitiva. Por fim, a Figura 17 apresenta a montagem física do circuito.

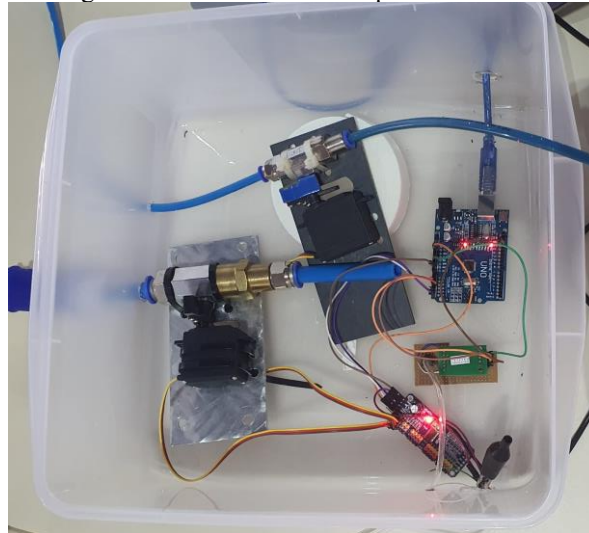
Figura 16 - Interface desenvolvida em MATLAB



Fonte: os autores, 2023.



Figura 17 - Conexão dos componentes físicos



Fonte: Os autores, 2023.

5 CONCLUSÕES

A realização da construção do protótipo demonstra a importância de uma interface associada a um aparelho de suporte à vida como este, permitindo a alteração das configurações do funcionamento do mesmo para adequar-se a cada caso. A junção do *hardware*, que abrange os servo motores, sensor de pressão, válvulas pneumáticas e demais componentes, juntamente com o *software*, que se relaciona à programação do MATLAB, e *firmware*, que corresponde ao código do microcontrolador Arduino Uno, permitiram a criação de um ventilador mecânico de uso intuitivo, eficiente e, especialmente, de baixo custo.

O uso da válvula tipo esfera se destacou como uma alternativa estratégica que não apenas proporcionou um controle satisfatório do fluxo de ar, como também contribuiu para a redução dos custos da construção do protótipo. Esse aspecto econômico é de extrema relevância, considerando que o VentiVet pode ser utilizado em ambientes onde recursos financeiros são limitados, como clínicas veterinárias filantrópicas, por exemplo.

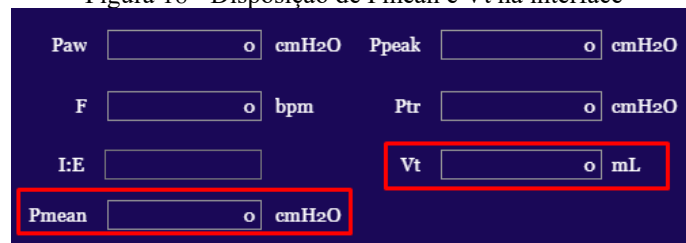
Além disso, a interface gráfica construída no MATLAB mostrou-se um elemento fundamental para o uso do dispositivo, pois facilita consideravelmente a interação entre o operador e o ventilador mecânico. A simplicidade e eficácia dessa interface tornam o equipamento acessível até mesmo para profissionais da medicina veterinária que não possuem grande experiência na área tecnológica. Isso amplia o potencial de aplicação do VentiVet em diferentes contextos.

Em síntese, este trabalho não apenas trouxe uma nova maneira de produzir um ventilador mecânico veterinário, como também ressaltou a importância em considerar a acessibilidade financeira e a facilidade no uso e desenvolvimento de tecnologias voltadas para a área da saúde animal. Apesar de seu funcionamento satisfatório, o equipamento ainda pode ser aprimorado pelo uso de:



- Alarmes: como mencionado anteriormente, já existem alarmes e protocolos de segurança implementados ao protótipo. Porém, estes podem ser aprimorados. Por exemplo, com acionamento de sinalizadores sonoros caso o funcionamento ou alguma leitura feita pelo ventilador esteja fora de conformidade. Dessa forma, o funcionamento estaria ainda mais seguro;
- Modos ventilatórios: atualmente, o único modo ventilatório em que o VentiVet opera é o controlado, que não permite nenhum tipo de autonomia ao paciente. Visando a melhoria do protótipo, planeja-se implementar o modo assistido-controlado, que abrange uma amplitude maior de tratamentos, tornando-o mais inclusivo;
- Adição de diferentes tipos de sensores: o sensor de pressão utilizado no momento não consegue fazer a leitura de alguns parâmetros que necessitam ser implementados, como Vt e Pmean (que já estão dispostos na interface, como ilustra a Figura 18, porém, sem nenhum código associado ao seu funcionamento). A implementação desses sensores também permitirá incluir gráficos para monitoramento de fluxo e volume. Para a resolução desse problema, é necessário instalar no protótipo um sensor de fluxo. Tal associação ainda se encontra pendente, pois, sensores como esses possuem alto custo para a faixa de leitura necessária.

Figura 18 - Disposição de Pmean e Vt na interface

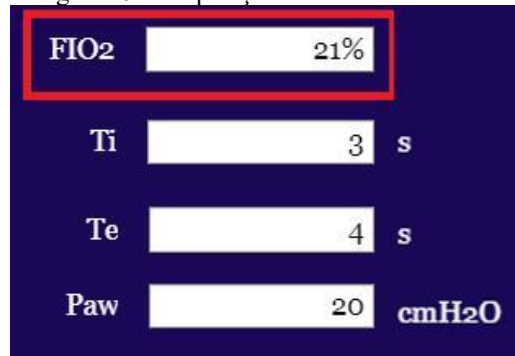


Fonte: Os autores, 2023.

- Adição de um cilindro de oxigênio: para um funcionamento satisfatório do protótipo, é necessário que um cilindro de oxigênio seja associado ao compressor. Dessa maneira, a FIO₂ pode ser ajustada, de acordo com a condição do paciente. A caixa de ajuste desse parâmetro também já está disposta na IHM, como exibido na Figura 19, porém, sem nenhum código associado. Sua marcação está constantemente em 21%, pois é equivalente à fração de oxigênio presente no ar atmosférico utilizado pelo compressor. Com a implementação dessa mudança, tornar-se-á necessária a adição de uma segunda válvula inspiratória proporcional, de forma a controlar a admissão de ar comprimido e gás oxigênio separadamente;



Figura 19 - Disposição de FIO₂ na interface



Fonte: Os autores, 2023.

- Controle do protótipo via Wi-fi ou Bluetooth: inicialmente, ao idealizar o projeto, foi planejado utilizar uma placa microcontroladora ESP-32, devido à possibilidade de integração Wi-fi e/ou Bluetooth, que se mostraria útil principalmente na exibição dos alarmes, mesmo que à distância. Devido a complicações técnicas anteriormente mencionadas, e considerando que a placa Arduino possui uma gama muito maior de exemplos de aplicações, não foi possível trazer essa idealização à realidade. Portanto, para a aprimoração do projeto, faz-se necessária essa implementação;
- Reorganização do *hardware* para tornar-se mais compacto: a caixa utilizada para manter a organização do *hardware*, atualmente, traz algumas limitações quanto à economia de espaço utilizada pelo protótipo. Para uma utilização mais conveniente, faz-se necessária uma reorganização dos componentes, utilizando-se majoritariamente do espaço vertical, com a substituição da caixa originalmente utilizada.
- Adição da PEEP: o controle da PEEP é essencial para um uso seguro do ventilador mecânico, considerando que sua atuação está ligada à preservação ou recuperação da saúde dos alvéolos. A PEEP funcionará como qualquer outro parâmetro editável, em que o valor numérico digitado deverá ser correspondente à pressão positiva mantida no pulmão ao fim da expiração.



REFERÊNCIAS

ADI, Puput Dani Prasetyo; KITAGAWA, Akio. ZigBee Radio Frequency (RF) Performance on Raspberry Pi 3 for Internet of Things (IoT) based Blood Pressure Sensors Monitoring. *International Journal of Advanced Computer Science and Applications*, v. 10, n. 5, p. 18-27, 2019.

ARDUINO. Arduino Uno Rev3, 2023. Disponível em: <https://store.arduino.cc/products/arduino-uno-rev3>. Acesso em: 11 nov. 2023.

CARVALHO, Carlos Roberto Ribeiro de; TOUFEN, Carlos Junior; FRANCA, Suelene Aires. Ventilação mecânica: princípios, análise gráfica e modalidades ventilatórias. *Jornal brasileiro de pneumologia*, v. 33, p. 54-70, 2007.

CASA DA ROBÓTICA. Módulo PWM PCA9685 Servo Motor Driver 16 Canais I2c, 2023. Disponível em: <https://www.casadarobotica.com/sensores-e-modulos/modulos/outros/modulo-pwm-pca9685-servo-motor-driver-16-canais-i2c>. Acesso em: 29 nov. 2023.

CASTRO, Fabiano Pimentel de. Servo motores: visão geral, controle e aplicação. 2016. 38 f. Trabalho de Conclusão de Curso (Especialização em Automação Industrial) - Universidade Tecnológica Federal do Paraná, Curitiba, 2016.

CASTRO, Marina Lopes. Princípios básicos da ventilação mecânica em cães. Monografia (Especialista no Curso de Pós-graduação *Latu sensu* em Residência em Medicina Veterinária) - Universidade Federal de Minas Gerais, 2011.

CAVALCANTE, Andressa Vallery Setubal de Oliveira Nunes. Análise do drive respiratório neural em indivíduos hipertensos durante a ventilação máxima. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, 2020.

CONSELHO FEDERAL DE MEDICINA VETERINÁRIA. Resolução N°1015, de 9 de novembro de 2012. [Conceitua e estabelece condições para o funcionamento de estabelecimentos médico-veterinários de atendimento a pequenos animais e dá outras providências.]. *Diário Oficial da União*: Seção 1, Brasília, DF, ano 150, n. 22, p.172-173, 31 jan. 2013.

CONSULTA DOG VET. Sistema respiratório, 2017. Disponível em: <https://consultadogvet.wordpress.com/2017/02/23/sistema-respiratorio/>. Acesso em: 7 dez. 2023.

DINAU, Priscilla Caroline Moutinho. Projeto de implementação de controle de posição e velocidade de servomotores DC brushless e proposta de posicionamento dinâmico de um modelo em escala de plataforma semi-submersível. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação). Universidade Federal do Rio de Janeiro, 2009.

DUGDALE, Alex. The ins and outs of ventilation 1. Basic principles. In *Practice*, v. 29, n. 4, p. 186-193, 2007.

HOBBY GOIÁS. Kit 4 Servo Mg996r Full Engrenagens Metal 12kg Mg996 Mg995, 2023. Disponível em: https://www.hobbygoias.com.br/MLB-2140585040-kit-4-servo-mg996r-full-engrenagens-metal-12kg-mg996-mg995-_JM. Acesso em: 30 nov. 2023.

HX711. *In*: AVIA Semiconductor. Disponível em: https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf. Acesso em: 26 nov. 2023.



MATHWORKS. MATLAB, 2023. Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 23 nov. 2023.

MELLEMA, Matthew S. Ventilator waveforms. *Topics in companion animal medicine*, v. 28, n. 3, p. 112-123, 2013.

MG995 Datasheet (PDF) - List of Unclassified Manufacturers, Alldatasheet, 2023. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132435/ETC2/MG995.html>. Acesso em: 29 nov. 2023.

MULTILÓGICA-SHOP, Placa Uno R3 SMD com cabo USB. Disponível em: <https://multilogica-shop.com/produtos/placa-uno-r3-smd-arduino-compativel/>. Acesso em: 29 nov. 2023.

ROMERO, Jorge Calderón. Confiabilidade Metrológica de Ventiladores Pulmonares. 2006. Tese de Doutorado. Dissertação (Mestrado em Metrologia para Qualidade e Inovação), Pontifícia Universidade Católica–PUC, Rio de Janeiro, 2006.

SILVA, Heuler Andrade. et al. DESENVOLVIMENTO DE PROTÓTIPO PARA CONTROLE DE TEMPERATURA PARA AQUECEDORES HÍDRICOS. *In: IX CONGRESSO DE ENGENHARIAS DA UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI*, 2019, São João del-Rei. Anais eletrônicos - Campinas, Galoá, 2019. Disponível em: <https://proceedings.science/coen-2019/trabalhos/desenvolvimento-de-prototipo-para-controle-de-temperatura-para-aquecedores-hidri?lang=pt-br>. Acesso em: 23 nov. 2023.

SILVA, Rogério Oliveira da; ARAUJO, Warley Monteiro; CAVALCANTE, Maxwell Machado. Visão geral sobre microcontroladores e prototipagem com arduino. *Tecnologias em Projeção*, v. 10, n. 1, p. 36-46, 2019.

UTFPR. VENT-U² Ventilador Pulmonar concebido na Universidade para você, 2020. Disponível em: <https://acao.utfpr.edu.br/acao/13/>. Acesso em: 7 dez. 2023.