

PROTÓTIPO DE RODA DE EXERCÍCIOS MICROCONTROLADA PARA ESTUDO DE COMPORTAMENTO DE ROEDORES: PROPOSTA DE OTIMIZAÇÃO E ADIÇÃO DE FUNCIONALIDADES.

Leonardo da Silva de Luca¹, Valter Blauth Júnior²

Resumo: A automação laboratorial tem se tornado cada vez mais relevante na pesquisa científica por sua capacidade de aumentar a eficiência, a precisão e a reprodutibilidade dos experimentos. Este trabalho tem por objetivo desenvolver um protótipo de sistema embarcado com foco em automação laboratorial, utilizando um microcontrolador ESP32, sensor indutivo e comunicação via API REST. O sistema será aplicado no monitoramento da atividade voluntária de roedores em rodas de exercício, buscando superar as limitações presentes em modelos atualmente disponíveis. O projeto visa melhorar a precisão na coleta de dados, otimizar funcionalidades e tornar a interface do usuário mais acessível. Para isso, foi criado um novo módulo de hardware responsável pela coleta e transmissão dos dados, além de uma aplicação *desktop* desenvolvida em Python, que atua como servidor e interface de controle. O sistema busca atender às necessidades do meio acadêmico ao proporcionar maior reprodutibilidade, eficiência e segurança nos estudos comportamentais com roedores, promovendo inovação tecnológica de baixo custo no ambiente de pesquisa. Os resultados demonstram que o protótipo apresentou desempenho satisfatório, com elevada precisão dos sensores, e melhorias significativas na interação com a interface do usuário. Além disso, o aplicativo de integração atendeu às expectativas, atuando sem perdas na importação dos dados. Embora os resultados tenham sido positivos, futuros aprimoramentos podem ser direcionados à melhoria dos processos de visualização e importação de dados, bem como à validação do sistema em diferentes cenários experimentais.

Palavras-chave: Sistemas embarcados; Internet das coisas; API REST; Automação laboratorial; Monitoramento comportamental de roedores;

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), leonardo.s.deluca@gmail.com

²Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), valterblauth@unesc.net

ABSTRACT: Laboratory automation has become increasingly relevant in scientific research due to its ability to enhance the efficiency, accuracy, and reproducibility of experiments. This work aims to develop a prototype of an embedded system focused on laboratory automation, using an ESP32 microcontroller, an inductive sensor, and communication via a REST API. The system will be applied to monitor the voluntary activity of rodents on exercise wheels, seeking to overcome the limitations present in currently available models. The project aims to improve data collection accuracy, optimize functionalities, and make the user interface more accessible. To achieve this, a new hardware module was created, responsible for data collection and transmission, along with a desktop application developed in Python, which acts as both server and control interface. The system is designed to meet the needs of academic research by providing greater reproducibility, efficiency, and safety in behavioral studies involving rodents, while promoting low-cost technological innovation in the research environment. The results show that the prototype demonstrated satisfactory performance, with high sensor accuracy and significant improvements in user interface interaction. Furthermore, the integration application met expectations, operating without data import losses. Although the results were positive, future improvements may focus on enhancing data visualization and import processes, as well as validating the system in different experimental scenarios.

Keywords: Embedded systems; Internet of things; REST API; Laboratory automation; Rodent behavior monitoring

1 INTRODUÇÃO

Desde sempre, a investigação científica se apresentou fundamental para a humanidade. Ela é essencial para o progresso da sociedade, o reforço da economia, a promoção da inovação e a resolução dos problemas difíceis e desafiantes que enfrentamos enquanto povo, lugar e planeta (Rosowsky, 2022).

Neste sentido, os sistemas embarcados emergem como instrumentos poderosos que auxiliam na investigação, especialmente ao integrar-se progressivamente nos ambientes de trabalho e processos industriais, elevando as taxas de produção, a eficiência e a qualidade dos resultados (Hitomi, 1994; Holland; Davies, 2020).

Os sistemas embarcados podem ser definidos como um computador embutido em algo que não é um computador, ou seja, são objetos

inteligentes que possuem uma combinação de hardware e software projetados para cumprir um propósito específico (Vasseur; Dunkels, 2010; Nureni, 2022).

Para a implementação de sistemas embarcados, seja em pesquisa, na indústria ou em casas inteligentes, a primeira escolha geralmente recai sobre o uso de microcontroladores. Os microcontroladores são autosuficientes, têm custo benefício e cabem perfeitamente para aplicações de sistemas embarcados em tempo real (Pang, 2015).

Um dos microcontroladores mais utilizados é o ESP32, suas principais qualidades estão relacionadas com a conectividade. O ESP32 conta com interfaces Wi-Fi e Bluetooth integradas, sendo compatível com vários protocolos Wi-Fi. Além disso, sua usabilidade é prática e versátil, ele pode ser programado utilizando várias linguagens, possuindo também compatibilidade com diversas outras placas e uma grande variedade de bibliotecas à disposição do programador. (Hercog et al., 2023).

Atualmente, quando se trata do tema de estudos laboratoriais e acadêmicos, é visto cada vez mais presente a utilização de métodos automatizados que auxiliam na análise e estudo dos objetos de pesquisa. São diversos os benefícios que a automação em laboratórios acadêmicos pode trazer, dentre eles pode-se citar a reprodutibilidade, a eficiência laboratorial, uma tradução mais rápida e fatores de segurança (Holland; Davies, 2020).

A automação laboratorial tem emergido como uma tendência significativa na pesquisa científica, trazendo avanços consideráveis na eficiência e precisão dos processos experimentais. São diversos os benefícios para os laboratórios, um dos mais pertinentes é seu uso na melhoria da reprodutibilidade da pesquisa laboratorial, que ocasiona na melhoria da qualidade dos testes laboratoriais, reduzindo a variabilidade induzida por humanos e aumentando a taxa de geração de dados (Holland; Davies, 2020).

Dentre as diversas áreas nas quais a automação laboratorial consegue operar, pode-se citar o uso para estudos de roedores. Os ratos e camundongos são animais ideais para pesquisas biomédicas devido às suas semelhanças com os humanos em termos de anatomia e fisiologia, uma vez que cerca de 95% dos seus genes são compartilhados pelas três espécies (Bryda, 2013).

No capítulo 12 do livro “Translational Neuroscience: Toward New Therapies”, (Robbins, 2015) descreve pesquisas que exploram aspectos comportamentais e cognitivos dos roedores, evidenciando o potencial da automação para conduzir testes de percepção, atenção, aprendizagem as-

sociativa, funções executivas e memória, entre outros.

As atividades dos roedores se caracterizam por séries de corrida contínua de aproximadamente 150s seguidas de intervalos curtos de recuperação de cerca de 30s, com velocidade média de 1,5 a 3 km/h, assemelhando-se ao treinamento intervalado empregado em seres humanos (Manzanares; Silva; Gandra, 2019).

Além disso, a automação permite uma abordagem inovadora na avaliação das respostas adaptativas ao exercício. Por meio da corrida voluntária em rodas, os pesquisadores podem investigar não somente as mudanças comportamentais e as capacidades de exercício, mas também os efeitos do exercício sobre doenças e envelhecimento (Landisch et al., 2008; McMullan et al., 2016). Apesar dos benefícios apontados, há também afirmações claras na literatura de que muitos investigadores em instituições acadêmicas têm sido lentos para adotar a automação laboratorial (Sadowski; Grant; Fell, 2016; Almeida; Ferreira, 2017; Jessop-Fabre; Sonnenschein, 2019).

Na elaboração deste trabalho, foram pesquisados diversos sistemas correlatos para monitoramento da atividade voluntária em roedores. O principal sistema analisado é de autoria local da UNESC. Nele é utilizado um sensor magnético para contagem de voltas, um Arduíno para leitura do sensor e um ESP32 para conexão Wi-Fi. Esse sistema apresenta falhas de leitura constantes, devido ao sensor, além de sua utilização não ser prática, já que a todo momento que se precisa modificar algo é necessário compilar os códigos ao microcontrolador novamente.

O estudo de Chomiak et al. (2016) utilizou um encoder angular combinado com conversores A/D, breakout boxes e uma câmera de alta velocidade para gerar dados contínuos de movimento e rotação, demonstrando eficácia na análise comportamental.

O Smart-Kage, de Ho et al. (2023), apresentou uma gaiola automatizada que integra câmeras e válvulas solenóides para monitoramento e testes cognitivos, proporcionando uma avaliação padronizada e contínua dos comportamentos dos camundongos.

Já o sistema WRAQ, descrito por Zhu et al. (2021), emprega um sensor de palheta em versões offline e online com ESP32 para registrar dados de atividade em tempo real e armazenar informações localmente, oferecendo uma solução econômica e prática. É o sistema com mais benefícios dentre os pesquisados, podendo citar como melhoria o sensor de palheta que pode falhar na contagem por trabalhar com campo magnético.

Portanto, o presente trabalho tem por objetivo projetar e otimizar um protótipo de roda de exercícios microcontrolada com IoT para monitorar a atividade voluntária de roedores, assegurando maior precisão e eficiência em pesquisas acadêmicas. Com foco principal nas melhorias específicas para o sistema de roda de exercício para roedores da UNESC.

Os objetivos específicos desta pesquisa consistem em melhorar a acurácia dos dados adquiridos pelos sensores, otimizar o sistema e suas funcionalidades, implementar a comunicação entre o ESP32 e a aplicação desktop por meio de uma API REST, adicionar novas funcionalidades ao protótipo, garantindo principalmente uma melhoria na interface do usuário e, por fim, reunir, validar e avaliar os resultados com os dados obtidos do sistema.

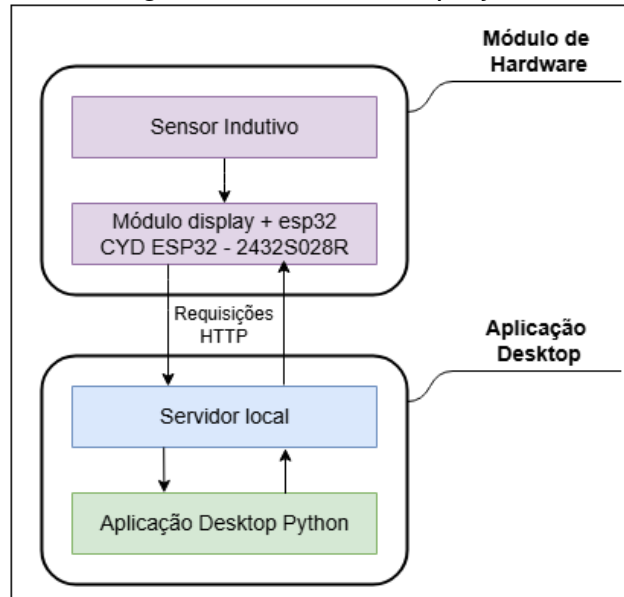
2 MATERIAIS E MÉTODOS

Esta pesquisa caracteriza-se por ser explicativa, aplicada de base tecnológica, com uma abordagem qualitativa, se inserindo nas áreas de internet das coisas (IOT) e sistemas embarcados. Quanto aos meios, a pesquisa se qualifica como bibliográfica e experimental.

O objetivo da pesquisa é aprimorar e melhorar um protótipo existente no ambiente educacional da UNESC, otimizando a interface do usuário, mobilidade e, principalmente, a precisão dos dados. Para isso, foi desenvolvido um novo protótipo de roda de exercícios para o uso em estudos comportamentais de roedores e também uma aplicação *desktop* em python para controle e acesso aos dados do protótipo.

Pode-se então dividir o projeto em duas partes. A primeira, e principal, consiste no desenvolvimento do módulo de hardware, responsável pela coleta de dados, pela interação com o usuário e pela comunicação com o servidor. A segunda parte corresponde ao desenvolvimento da aplicação *desktop*, na qual é criado o servidor responsável pela comunicação por meio de chamadas HTTP que conectam o dispositivo à aplicação. Na figura 1 tem-se a divisão do projeto em módulos.

Figura 1- Módulos do projeto.



Fonte: Elaborado pelo autor.

2.1 APLICAÇÃO DESKTOP

Para o desenvolvimento da aplicação *desktop*, foi utilizada a linguagem Python na versão 3.13.3, no ambiente do *Visual Studio Code*, versão 1.99.3 em *Windows 11*. Foram utilizados alguns pacotes Python para auxiliar no desenvolvimento, são eles: PyQt6 versão 6.9.0, Werkzeug 3.1.3, Flask versão 3.1.0 e Pandas 2.2.3.

O pacote do PyQt6 foi utilizado para a construção do front da aplicação, dentro dele existem diversos componentes pré-definidos usados para a construção das telas, pode-se citar, QWidget, QLabel e QPushButton. O pacote do Pandas foi utilizado para a conversão dos dados para excel possibilitando o download dos dados para posterior análise. E por fim, os pacotes Werkzeug e Flask são utilizados para os momentos de criação, conexão e roteamento do servidor local.

A aplicação inicia em uma tela de login, desenvolvida com usuário e senhas fixos somente para testes, pois não é um tópico de extrema importância para o funcionamento do projeto, podendo ser tratado posteriormente em trabalhos futuros.

Passando pela tela de login, tem-se o menu principal da aplicação. A partir dele é possível acessar as três funções existentes no programa. A característica principal da aplicação está na sua capacidade de abrir um servidor local para que possa ocorrer a comunicação entre o ESP e um computador. Ocorrendo essa comunicação, é possível fazer a trans-

ferência de arquivos sem a necessidade de conexão com fios entre os dispositivos, apesar de que ainda é uma conexão em rede local, sendo assim ambos os dispositivos devem estar conectados na mesma rede.

A aba "Ligar Servidor" é responsável por iniciar um servidor que permite a comunicação entre dispositivo e aplicação por meio de requisições HTTP a uma API REST. Optou-se por manter toda a API no computador host, simplificando a configuração para o usuário. Como esse servidor opera localmente no computador, quando é necessário que a aplicação se comunique com o ESP, o dispositivo realiza leituras periódicas em um endpoint da API. Isso possibilita que o ESP receba comandos do programa. Nos demais momentos, a comunicação ocorre normalmente, com o ESP enviando informações para os endpoints existentes.

Além da comunicação dos dados, foi criado também um endpoint de "ping" que permite ao ESP enviar requisições periódicas para ver a situação do servidor.

Ainda na aba de ligar o servidor, é exibido ao usuário o endereço que deve ser configurado no ESP para estabelecer a conexão. Esse endereço é composto pelo IPv4 da máquina e pela porta utilizada. Isso facilita o processo de configuração, permitindo que mesmo usuários sem conhecimentos técnicos aprofundados realizem a integração de forma simples. A figura 2 possui uma imagem da tela responsável por ligar o servidor.

Figura 2 - Tela ligar servidor.



Fonte: Elaborado pelo autor.

Após a conexão ser estabelecida, já é possível visualizar os da-

dos provenientes da roda de atividades. O usuário conta com duas opções de visualização. Na aba "Live View", será exibida a última volta registrada pela roda, o que permite verificar rapidamente se algum animal está se movimentando, sem que seja necessário fazer todo o processo de importação de dados.

A segunda opção consiste na importação dos dados salvos localmente no dispositivo. O usuário faz a requisição de importação e espera o microcontrolador enviar o arquivo. O arquivo recebido é no formato binário (.bin), escolhido por ser leve, já que cada atividade do roedor ocupa apenas 12 bytes, o que facilita o armazenamento no dispositivo e a transferência. Embora o arquivo seja binário, o usuário tem duas opções para baixá-lo em formatos mais acessíveis para visualização, seja em .txt ou .xlsx. A figura 3 é dividida entre as telas de *live view* e importação.

Figura 3 - Live view e importação.



Fonte: Elaborado pelo autor.

Tanto na visualização em tempo real quanto na importação dos dados, cada atividade contém os campos, número de voltas, tempo da atividade, e data de início da atividade. Já as informações sobre o nome da gaiola e o diâmetro da roda são enviadas apenas uma vez, no momento da importação.

Os dados de velocidade e distância são calculados diretamente na aplicação, utilizando as fórmulas de cálculo da circunferência e da velocidade média. Por fim, com todas as informações disponíveis, cabe ao pesquisador decidir a melhor forma de realizar a análise.

2.2 MÓDULO DE HARDWARE

O módulo de hardware representa o componente central deste projeto, enquanto a aplicação *desktop* pode ser considerada um complemento para a visualização dos dados, porém ainda com sua devida importância.

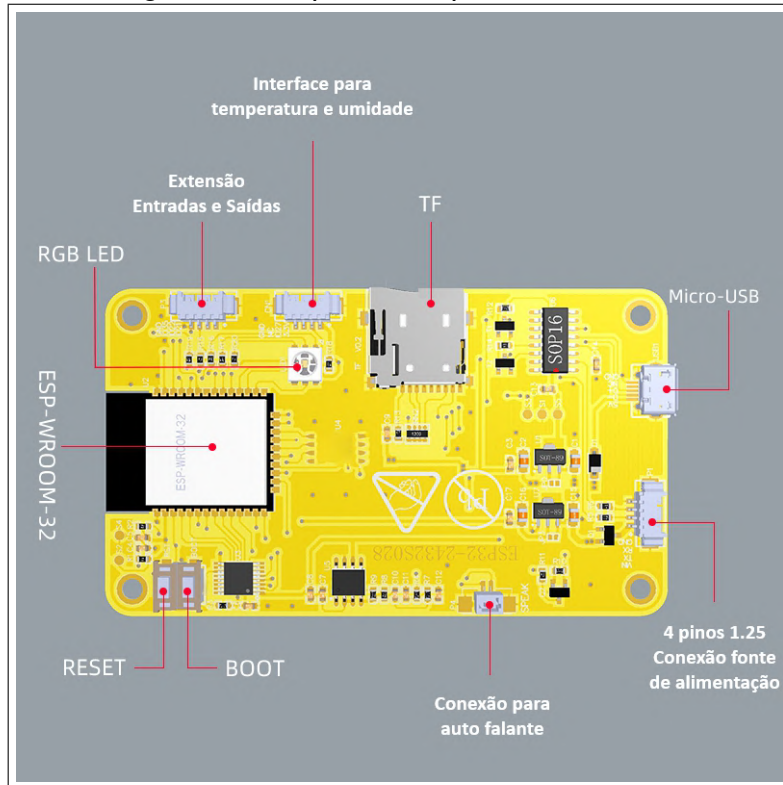
É no dispositivo onde ocorre a coleta e transmissão das informações, além de ser o local onde se configuram os parâmetros da roda. A incorporação de um módulo de *display* com *touchscreen* acoplado ao ESP32, torna o sistema mais compacto e melhora significativamente a interação usuário–máquina. Sendo assim, ao adotar o display no dispositivo, busca-se suprir uma das principais lacunas observadas em projetos semelhantes, a melhoria da facilidade de uso e da interação entre o usuário e o sistema.

2.2.1 Componentes e montagem do sistema

Para a montagem do dispositivo, foi utilizado um microcontrolador ESP32, em função de sua conectividade sem fio, baixo custo e facilidade de programação. Um diferencial da implementação do ESP é que ele vem em conjunto com uma placa de desenvolvimento com um *display*.

A placa possui nome popular de CYD2USB(ESP32 2432S028) e já integra um *display touchscreen* de 2,8 polegadas modelo TPM408-2.8, simplificando assim a conexão entre o ESP32 e o painel gráfico. Além disso, essa placa oferece slot para cartão SD, LED RGB traseiro, saída para altofalante e cabeamento dedicado para sensores de temperatura e umidade. Uma desvantagem do uso da CYD2USB é que a maior parte dos pinos do ESP32 já está reservada para o funcionamento dos componentes integrados, restando apenas uma entrada livre para a conexão de dispositivos externos. A figura 4 possui um esquema da placa CYD2USB.

Figura 4 - Esquema da placa CYD2USB.



Fonte: Lough (2024). Adaptado pelo autor.

Para a coleta dos dados de voltas, foi escolhido o sensor indutivo não faceado LM18-3008PA, um dispositivo PNP normalmente aberto de três fios. Optou-se por esse modelo pela simplicidade de uso e pela maior imunidade a falsos positivos em rotações elevadas, em comparação a sensores magnéticos. Seu funcionamento ocorre quando o sensor é aproximado de qualquer objeto metálico, seja ele magnético ou não, a uma distância de aproximadamente oito milímetros.

O sensor é alimentado por uma fonte regulada de 6 V e 2 A, proporcionando uma margem de corrente suficiente para o funcionamento estável do sistema.

2.2.2 Desenvolvimento e programação do ESP

O desenvolvimento do firmware para o ESP32 ocorreu na Arduino IDE versão 2.3.4, com a instalação do suporte à placa ESP32 na versão 3.2.0. A linguagem utilizada foi a C++ que já padrão da IDE também para desenvolvimento em Arduino.

Para implementar a interface gráfica e o tratamento de toques, foram incorporadas duas bibliotecas externas. Temos a TFT_eSPI na versão 2.5.43, responsável pela renderização de textos, gráficos e elemen-

tos na tela TFT via protocolo SPI (*Serial Peripheral Interface*), e a biblioteca XPT2046_Touchscreen na versão 1.4, que provê o suporte à detecção de toques na camada resistiva do controlador XPT2046 integrado à placa CYD2USB.

Cada interface gráfica foi construída manualmente, definindo-se explicitamente as coordenadas X e Y de cada elemento. Da mesma forma, o tratamento de toque requer a configuração manual das regiões ativas na tela. Essa abordagem, embora proporcione maior controle sobre o layout e a interação, torna o desenvolvimento mais trabalhoso e propenso a erros, pois cada modificação feita no layout deve ser feita também no toque, e um componente editado vai interferir nos demais componentes da tela. O exemplo a seguir da figura 5 demonstra como é feita a renderização em tela.

Figura 5 - Exemplo código do display.

```
// Botão OFF
if (x > 65 && x < 135 && y > 100 && y < 140) {
  setBacklightBrightness(0);
  screenOff = true;
  offTouchTime = millis();
  drawBacklightConfigScreen();
  return;
}
// Botão OK
if (x > 195 && x < 265 && y > 180 && y < 220) {
  currentScreen = HOME;
  drawHomeScreen();
  return;
}

void drawBacklightConfigScreen() {
  ...

  Button offButton = { 65, 100, 70, 40, "OFF" };
  drawButton(offButton);

  Button okButton = { 195, 180, 70, 40, "OK" };
  drawButton(okButton);

  Button nameButton = { 195, 100, 70, 40, "Nome" };
  drawButton(nameButton);

  Button diamButton = { 65, 180, 70, 40, "Diam." };
  drawButton(diamButton);
}
```

Fonte: Elaborado pelo autor.

Além disso, o firmware faz uso de sete bibliotecas internas do CORE do ESP32 para gerir funcionalidades de tempo, conexão de rede, comunicação HTTP, armazenamento em memória flash e sincronização de horário. A `Arduino.h`, fornece funções de temporização em milissegundos (por meio de `millis()`). gerencia a conexão e reconexão automática à rede sem fio, enquanto a `HTTPClient.h` viabiliza o envio e recebimento de requisições RESTful ao servidor local.

Para a persistência dos dados na memória não volátil do ESP, utilizou-se a biblioteca `Preferences.h`, que armazena parâmetros como o SSID da rede e o diâmetro da roda, e a `SPIFFS.h`, em conjunto com a `FS.h`, para gravação e leitura de arquivos de log no sistema de arquivos interno do ESP32.

Por fim a biblioteca `"Time"` sincroniza o relógio interno via proto-

colo NTP (*Network Time Protocol*) sempre que há conexão com a Internet.

2.2.3 Funcionamento e interface

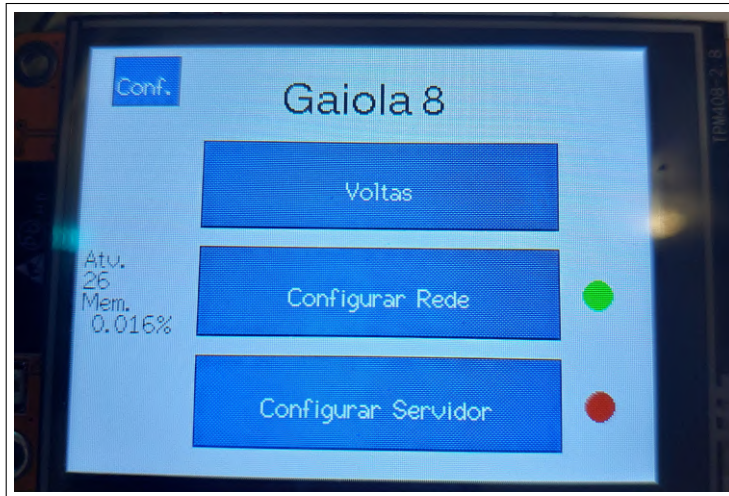
O funcionamento do dispositivo concentra-se principalmente na coleta de dados de voltas, bem como na configuração dos parâmetros da roda e dos servidores da aplicação. A visualização dos dados em tempo real na tela não é o objetivo central do projeto, apesar de possuir a opção de visualizar os dados da atividade atual.

Ao ligar, o *firmware* faz as configurações iniciais de tela, toque e variáveis. Ele inicia montando o sistema de arquivos SPIFFS, onde ficam armazenados o histórico de atividades e um cabeçalho contendo o nome da gaiola (32 bytes) e o diâmetro da roda (float), que será utilizado posteriormente ao exportar os dados. As credenciais de Wi-Fi, ip do servidor, o nome da gaiola e o diâmetro da roda são mantidos em memória não-volátil através da API Preferences, permitindo que essas configurações sobrevivam a desligamentos.

A tela inicial do projeto é onde o usuário inicia as interações, no topo da tela é possível ver o nome da gaiola configurado atualmente, abaixo tem-se os botões de acesso ao gráfico de voltas, configuração da rede WiFi e configuração do servidor. À esquerda dos botões, é mostrado ao usuário quantas atividades estão salvas em memória, bem como a quantidade de memória utilizada. Como as atividades são salvas em arquivos binários, a porcentagem normalmente ficará em valores baixos de uso, já que o ESP possui 4MB (4.000.000 bytes) de flash, totalizando em média mais de 499.000 atividades.

Já à direita o usuário pode verificar os indicadores de conexão de rede e servidor, representados por círculos coloridos, verde para conectado e vermelho para desconectado. Por fim, no canto superior esquerdo existe a tela de configuração. Na figura 6 é possível ver a tela inicial.

Figura 6 - Tela inicial.



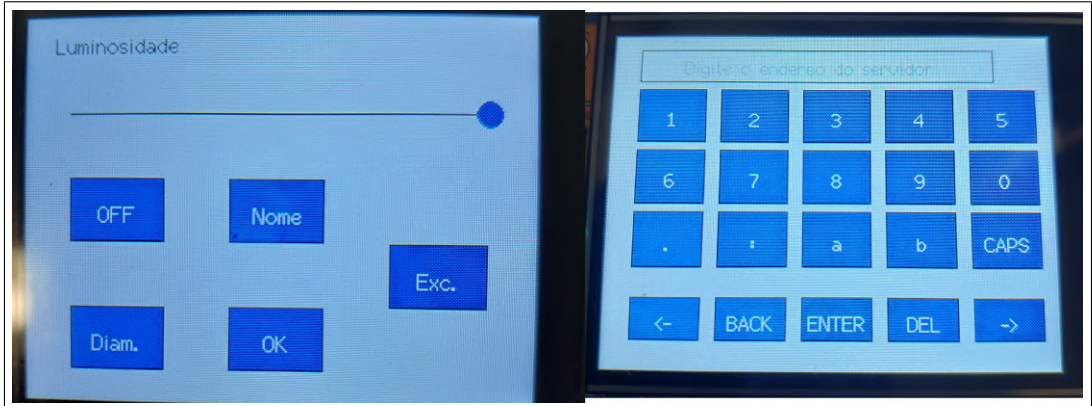
Fonte: Elaborado pelo autor.

A tela de configuração é composta por cinco botões e um controle deslizante. O botão "Nome" e o botão "Diam." configuram respectivamente o nome da roda e o seu diâmetro. A informação do nome é somente para organização, mas o diâmetro da gaiola vai influenciar nos cálculos de distância percorrida e velocidade do roedor.

Ao acessar as opções de configuração de nome, rede WiFi ou servidor, é apresentado ao usuário um teclado alfanumérico na tela. Esse teclado permite a inserção de letras maiúsculas e minúsculas, números e pontos, possibilitando assim a configuração do nome, rede WiFi e endereço do servidor. Outro botão presente é o "Exc." ele é responsável pela exclusão dos dados do dispositivo caso seja de interesse do usuário.

Por fim, há a configuração de brilho da tela. Ela pode ser ajustada através do controle deslizante ou ser completamente desligada se apertar em "OFF". Essa funcionalidade é extremamente relevante, pois a luz emitida pelo *display* pode interferir no comportamento natural dos roedores, prejudicando a coleta de dados confiáveis durante os experimentos. Na figura 7 tem-se à esquerda a tela de configurações e à direita a tela de teclado, no qual é utilizada em diferentes áreas do sistema.

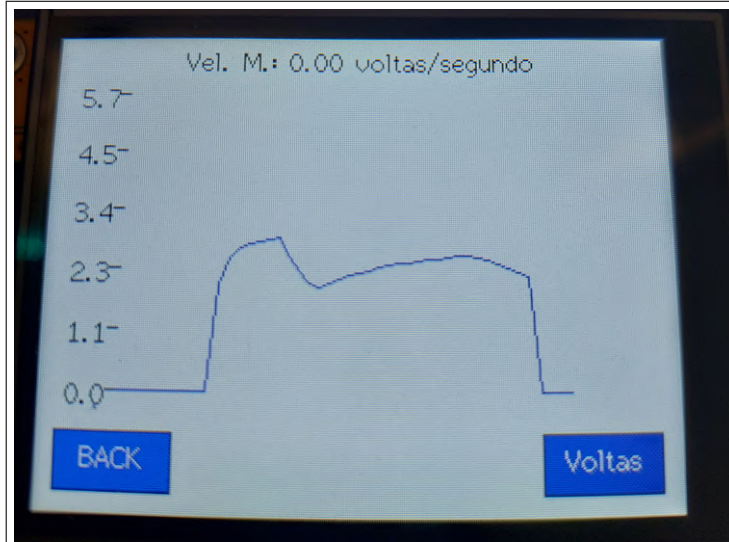
Figura 7 - Tela configuração e teclado.



Fonte: Elaborado pelo autor.

Partindo da tela de início para a tela de voltas, estão disponíveis dois tipos de gráficos para a visualização em tempo real da atividade atual. O primeiro gráfico exibe o número de voltas realizadas pelo roedor ao longo do tempo, já o segundo apresenta a velocidade média do roedor, como é visto na figura 8.

Figura 8 - Tela gráfico de voltas.



Fonte: Elaborado pelo autor.

O ESP32 executa um *loop* que monitora continuamente o sensor de voltas e atualiza imediatamente tanto o contador interno quanto a interface gráfica. Assim, o registro de voltas não é interrompido por mudanças de tela.

O programa trabalha com atividades dos roedores e não dados contínuos. Para definir o término de uma atividade, é adotado um intervalo

de cinco segundos sem novas voltas. Se esse intervalo expirar, entende-se que a atividade chegou ao fim e seus dados são gravados no arquivo binário e também são enviados ao modo *Live View* do aplicativo via requisição POST.

Algumas outras requisições são enviadas paralelamente, como GETs aos endpoints `/ping` e `/upload-status`, verificando se o servidor está acessível e se houve solicitação para importar todo o histórico. Essas sondagens são feitas com tempo de espera prolongado para evitar travamentos em redes instáveis e não bloqueiam a navegação do usuário.

2.2.4 Metodologia dos Testes

Para avaliar a precisão do sensor indutivo na contagem de voltas, foram realizados testes simulados práticos em uma roda de 15 centímetros de diâmetro. Cada teste durou cerca de 30 segundos e com velocidades manuais mantidas na faixa média observada em camundongos (1,5–3,0 km/h), de modo a evitar regimes extremos de giro.

Para os testes piloto e testes finais, os giros foram realizados manualmente. Cada teste iniciava-se com a ativação de um cronômetro de 30 segundos, a partir do qual eram iniciados tanto os giros manuais quanto a contagem pelo tacômetro. Buscou-se manter a mesma distância entre a roda e o tacômetro, além de uma velocidade aproximada entre os testes. No entanto, por se tratarem de testes manuais, pode haver variações na velocidade. Para compensar essas flutuações, adotou-se como métrica de desempenho o erro relativo. Em cada sessão, o sensor registrava o número de voltas (N_{sens}), enquanto um tacômetro digital fornecia o número de voltas de referência (N_{ref}). Esses valores foram aplicados na fórmula de erro relativo, definida como a razão entre o erro absoluto e o valor de referência, conforme descrito por Zhang et al. (2014) na seguinte fórmula:

$$\text{Erro relativo(\%)} = \left| \frac{N_{sens} - N_{ref}}{N_{ref}} \right| \times 100\%$$

Inicialmente, dez testes piloto sob as mesmas condições permitiram estimar o desvio-padrão σ do erro relativo. Com esse valor, calculou-se o tamanho da amostra necessário n para garantir nível de confiança de 95% ($Z=1,96$) e margem de erro máxima de 2% ($E=2\%$), conforme Saprà (2022) na seguinte fórmula:

$$\epsilon = \left(\frac{Z * \sigma}{E} \right)^2$$

Esses indicadores foram usados para validar a precisão do sistema e confirmar sua adequação para os estudos acadêmicos.

Além disso, foram feitos testes manuais de conectividade entre o ESP32 e a aplicação desktop, nos quais se monitorou a estabilidade da comunicação HTTP, a latência média de resposta e a integridade dos dados manuseados via API REST.

3 DISCUSSÃO E RESULTADOS

O sistema de roda de exercício com microcontrolador desenvolvido demonstrou-se eficaz tanto na coleta dos dados quanto na transmissão para o aplicativo. Além disso, apresentou melhorias significativas na precisão das medições e no design da interface do usuário, evidenciando o seu potencial para a aplicação em pesquisas acadêmicas.

O aplicativo *desktop* se demonstrou eficiente na sua proposta. Durante os testes práticos, foi capaz de configurar a rede corretamente a partir do endereço IP informado, sendo testado em três localidades distintas, utilizando três redes Wi-Fi diferentes. Esse desempenho evidencia sua capacidade de mobilidade e adaptabilidade, permitindo seu uso em diferentes ambientes. As chamadas http funcionaram corretamente e os arquivos gerados pela importação tiveram seus dados verificados corretamente, sem qualquer perda de informação, comprovando a confiabilidade do sistema no armazenamento e manipulação dos dados recebidos. São opções válidas de melhoria a possibilidade de usar em nuvem e outros métodos de expor os dados.

Com relação aos testes no dispositivo, primeiramente foram feitos os cálculos de desvio padrão e taxa amostral para descobrir a quantidade necessária de testes de precisão. Para o desvio padrão nos 5 testes piloto, foi encontrado o valor de $\sigma = 4,34\%$ com os dados e a fórmula. A seguir é apresentada a fórmula e a tabela 1 demonstra os resultados pilotos.

$$\sigma = \sqrt{\frac{\sum(\text{valor} - \text{mdia})^2}{\text{tam.populaaao}}}$$

Tabela 1 - Dados testes piloto.

Número de voltas	Duração(s)	Distância(m)	Vel.Média(m/s)	Valor tacômetro (rpm)	Voltas tacômetro	Erro relativo
41	30	12,88	0,39	78,7	39	5%
34	29	10,68	0,37	67,5	33	3%
36	29	11,31	0,39	72,8	35	3%
34	29	10,68	0,37	71,3	34	0%
40	28	12,57	0,45	76,5	36	11%

Fonte: Elaborado pelo autor.

A partir do desvio padrão, foi feito o cálculo da taxa amostral, levando em consideração uma margem de erro de 2%. O resultado indicou a necessidade de aproximadamente 18 ensaios para alcançar o nível de precisão estatística desejado, os dados da tabela vieram diretamente do importar do aplicativo desenvolvido, os tempos variaram de 29 a 31 segundos, no qual foi medido pelo dispositivo. A tabela 2 a seguir demonstra os resultados obtidos da comparação do valor medido pelo sensor e do valor do tacômetro:

Tabela 2 - Dados testes finais.

Número de voltas	Duração(s)	Distância(m)	Vel.Média(m/s)	Valor tacômetro (rpm)	Voltas tacômetro	Erro relativo
37	29	17,44	0,60	70,5	34	9%
29	30	13,67	0,46	59,7	30	3%
31	30	14,61	0,49	64,1	32	3%
34	30	16,02	0,53	67,3	34	0%
31	30	14,61	0,49	59,2	30	3%
32	30	15,08	0,50	58	29	10%
32	29	15,08	0,52	59,2	29	10%
31	30	14,61	0,49	62,6	31	0%
33	31	15,55	0,50	59,4	31	6%
32	30	15,08	0,50	66,2	33	3%
32	29	15,08	0,52	60,2	29	10%
32	29	15,08	0,52	71,8	35	9%
34	30	16,02	0,53	60,7	30	13%
35	30	16,49	0,55	65,2	33	6%
35	30	16,49	0,55	65	33	6%
32	30	15,08	0,50	58,9	29	10%
36	31	16,96	0,55	70,9	37	3%
33	30	15,55	0,52	61,9	31	6%

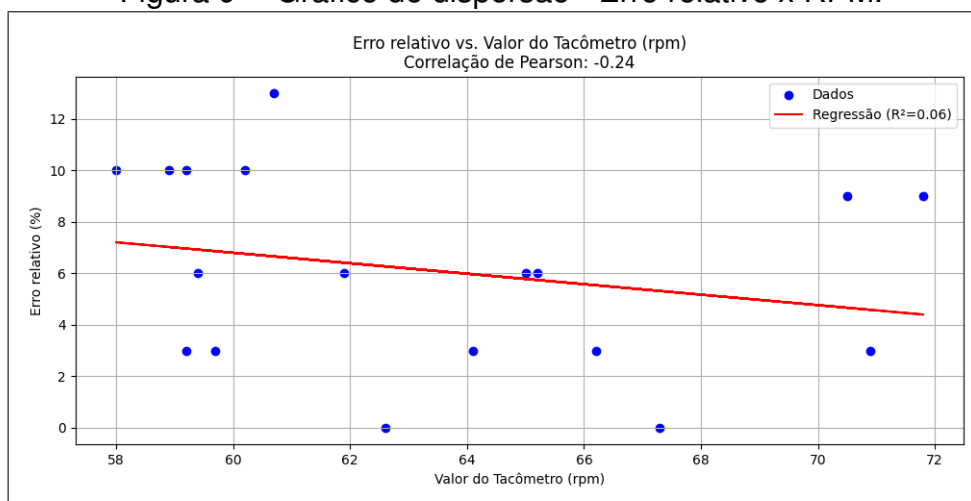
Fonte: Elaborado pelo autor.

Analisando as 18 amostras, revela que o sensor indutivo registra, em média, 32,8 voltas por sessão, com erro relativo médio de 6,11 %. Em termos absolutos, isso representa uma variação de mais ou menos 2 voltas, gerando um desvio de aproximadamente 0,94 metros na estimativa da distância percorrida. Apesar da variação observada, é necessário salientar que a medição com o tacômetro a laser está sujeita a interferências por causa da variação na refletividade do marcador, mudanças no ângulo e na distância do feixe, além de luzes do ambiente que podem causar leituras erradas ou instáveis.

Por outro lado, o sensor indutivo empregado realiza a detecção das voltas por meio do contato com uma superfície metálica de forma imune a esses fatores, resultando em menor variação nas medições e maior reprodutibilidade nos resultados.

O gráfico de dispersão a seguir mostra como o erro relativo (%) varia em função do valor indicado pelo tacômetro (rpm). Observa-se uma leve tendência de queda do erro à medida que o rpm aumenta, mas a correlação de Pearson de $-0,24$ e o baixo valor do coeficiente de determinação R^2 ($0,06$) indicam que essa relação é fraca e explica apenas 6% da variabilidade dos dados.

Figura 9 - Gráfico de dispersão - Erro relativo x RPM.



Fonte: Elaborado pelo autor.

Em futuros testes, pode-se optar pelo uso de um *encoder* incremental de alta resolução acoplado diretamente ao eixo, o que reduziria ou eliminaria as incertezas relacionadas à contagem das voltas, embora a um custo mais elevado.

Como parte da pesquisa, foram identificados alguns trabalhos semelhantes, cada um com suas particularidades. No entanto, esses sistemas apresentam algumas desvantagens ou simplesmente uma abordagem diferente do trabalho desenvolvido.

O artigo de Chomiak et al. (2016) descreve o desenvolvimento de um sistema para avaliação do comportamento locomotor de roedores por meio de uma roda de corrida, utilizando um notebook conectado a uma placa A/D de alta amostragem (NI DAC-Card 6024E), uma breakout box (NI BNC-2090), uma porta IEEE 1394a de alta velocidade e um encoder angular (modelo 6639S-1-103; Digi-Key, EUA). Os resultados demonstram que

o sistema permite uma análise precisa e detalhada de diversos parâmetros locomotores; no entanto, sua complexidade operacional e elevado custo representam limitações importantes, especialmente em comparação com a proposta apresentada neste trabalho.

O artigo de Ho et al. (2023) apresenta o Smart-Kage, uma gaiola automatizada para fenotipagem contínua da cognição e comportamento de camundongos. O sistema é composto por microcontroladores que controlam válvulas solenoides, câmeras superiores para gravação contínua, sensores para detecção de movimento, demonstrando alta precisão na diferenciação entre grupos controle, modelos de Alzheimer e animais lesionados, contudo, os dispositivos empregados podem gerar vibrações, ruídos e emissões de calor que interferem no comportamento natural dos camundongos. Em contraste, o sistema desenvolvido neste trabalho adota medidas para minimizar esses impactos, como o controle da luminosidade do ambiente e o uso de sensores de baixa emissão de ruído, buscando preservar o comportamento espontâneo dos roedores durante a coleta de dados.

O artigo de Zhu et al. (2021) apresenta o WRAQ (Wheel-Running Activity Acquisition), um sistema de baixo custo para monitoramento contínuo da atividade de corrida e ritmo circadiano em camundongos, disponível em versões offline, com adalogger e cartão microSD, e online, WRAQ-WIFI, que usa ESP32 e transmissão via Wi-Fi. Ambos os modelos usam uma roda com sensor de palheta (reed switch) e ímã para contagem de voltas, alimentados por bateria de polímero de lítio com hibernação. Embora semelhante ao sistema proposto, o WRAQ tem desvantagens, como desgaste mecânico do sensor, baixa frequência de resposta e, especialmente, a emissão de cliques audíveis, que podem interferir no comportamento dos camundongos.

4 CONCLUSÃO

O presente trabalho teve como objetivo projetar melhorias nos sistemas de exercícios para roedores já existentes. Para isso, foi desenvolvido um protótipo de dispositivo de monitoramento da atividade voluntária de roedores em rodas de exercício, aplicando conceitos de tecnologias de Internet das Coisas (IoT) e microcontroladores. O sistema apresentou desempenho satisfatório, com alta precisão na coleta de dados e interface aprimorada, facilitando a operação e visualização dos resultados.

Os resultados demonstram que o sensor indutivo registra as voltas da roda com elevada consistência e confiabilidade, o que o torna ade-

quado à medição das atividades voluntárias em roedores. Simultaneamente, o aplicativo *desktop* provou-se estável e intuitivo, configurando-se corretamente em diferentes redes Wi-Fi, executando requisições HTTP sem interrupções e importando todos os registros sem qualquer perda de informação. Esses resultados reforçam que o sistema desenvolvido atende de forma satisfatória às exigências de estudos comportamentais, garantindo consistência nos dados, estabilidade operacional e praticidade no uso.

O protótipo oferece uma solução eficiente para os processos de pesquisa acadêmica, promovendo maior reprodutibilidade e eficiência nos estudos comportamentais. Com isso, contribui positivamente para o avanço da pesquisa científica em universidades, ao proporcionar dados confiáveis que apoiam a análise e interpretação de experimentos.

Para trabalhos futuros sugere-se: submeter o projeto a testes com animais, implementar uma bateria externa para o dispositivo, possibilitando o salvamento de data e hora, ampliação das possibilidades de importação de dados e visualização no aplicativo, além de possibilitar o envio para a nuvem para acessar remotamente.

De modo geral, o protótipo oferece uma solução acessível e confiável para investigações comportamentais com roedores, aprimorando a reprodutibilidade e a eficiência das investigações científicas.

5 *

REFERÊNCIAS

ALMEIDA, M. D.; FERREIRA, R. **Taking Biotech to the Next Level with Laboratory Automation**. [S.l.]: Labiotech, 2017.

BRYDA, E. C. **The Mighty Mouse: the impact of rodents on advances in biomedical research**. United States: Mo Med, 2013. v. 110. 207–211 p.

CHOMIAK, T. et al. **Development and testing of a new system for assessing wheel-running behaviour in rodents**. BMC Research Notes, 2016. v. 9. ISSN 1756-0500. Disponível em: <<http://dx.doi.org/10.1186/s13104-016-2059-6>>.

HERCOG, D. et al. **Design and Implementation of ESP32-Based IoT Devices**. Sensors, 2023. v. 23. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/23/15/6739>>.

HITOMI, K. **Automation—its concept and a short history**. [S.l.]: Technovation, 1994. v. 14. 121–128 p.

HO, H. et al. **A fully automated home cage for long-term continuous phenotyping of mouse cognition and behavior**. Cell Reports

Methods, 2023. v. 3. 100532 p. ISSN 2667-2375. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2667237523001686>.

HOLLAND, I.; DAVIES, J. A. **Automation in the Life Science Research Laboratory**. Frontiers in Bioengineering and Biotechnology, 2020. v. 8. ISSN 2296-4185. Disponível em: <https://www.frontiersin.org/articles/10.3389/fbioe.2020.571777>. Acesso em: 06-04-2024.

JESSOP-FABRE, M. M.; SONNENSCHNEIN, N. **Improving Reproducibility in Synthetic Biology**. Frontiers in Bioengineering and Biotechnology, 2019. v. 7. ISSN 2296-4185. Disponível em: <https://www.frontiersin.org/articles/10.3389/fbioe.2019.00018>. Acesso em: 14-04-2024.

LANDISCH, R. M. et al. **Adaptive and nonadaptive responses to voluntary wheel running by mdx mice**. Muscle & Nerve, 2008. v. 38. 1290-1293 p. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mus.21141>.

LOUGH, B. **ESP32-Cheap-Yellow-Display**. 2024. <https://github.com/witnessmenow/ESP32-Cheap-Yellow-Display>. Acesso em: 26 abr. 2025.

MANZANARES, G.; SILVA, G. Brito-da; GANDRA, P. **Voluntary wheel running: patterns and physiological effects in mice**. Brazilian Journal of Medical and Biological Research, Associação Brasileira de Divulgação Científica, v. 52, n. 1. ISSN 0100-879X. Disponível em: <https://doi.org/10.1590/1414-431X20187830>.

MCMULLAN, R. C. et al. **Long-term exercise in mice has sex-dependent benefits on body composition and metabolism during aging**. [S.l.]: Physiol. Rep., 2016. v. 4. e13011 p.

NURENI, Y. **OVERVIEW OF EMBEDDED SYSTEM ITS APPLICATION**. [S.l.]: 3RD INTERNATIONAL ACADEMIC CONFERENCE ORGANIZED BY ACADEMIC STAFF UNION OF POLYTECHNICS IREE CHAPTER, 2022.

PANG, J. **Review of microcontroller based intelligent traffic light control**. [S.l.]: 2015 12th International Conference Expo on Emerging Technologies for a Smarter World (CEWIT), 2015. 1-5 p.

ROBBINS, T. W. **From rodent behavioral models to human disorders**. London, England: Translational Neuroscience, 2015. 183–208 p.

ROSOWSKY, D. **The Role Of Research At Universities: Why It Matters**. Forbes, 2022. Disponível em: <https://www.forbes.com/sites/davidrosowsky/2022/03/02/the-role-of-research-at-universities-why-it-matters/>. Acesso em: 14-04-2024.

SADOWSKI, M. I.; GRANT, C.; FELL, T. S. **Harnessing QbD, Programming Languages, and Automation for Reproducible Biology**. Trends in Biotechnology, 2016. v. 34. 214-227 p. Special Issue: Industrial Biotechnology. ISSN 0167-7799. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167779915002462>>.

SAPRA, R. L. How to calculate an adequate sample size? In: _____. **How to Practice Academic Medicine and Publish from Developing Countries? A Practical Guide**. Singapore: Springer Nature Singapore, 2022. p. 81–93. ISBN 978-981-16-5248-6. Disponível em: <https://doi.org/10.1007/978-981-16-5248-6_9>.

VASSEUR, J.-P.; DUNKELS, A. **Chapter 1 - What Are Smart Objects?** Boston: Interconnecting Smart Objects with IP, 2010. 3-20 p. ISBN 978-0-12-375165-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780123751652000016>>.

ZHANG, Y. et al. **Relative accuracy evaluation**. PLoS ONE, Public Library of Science, v. 9, n. 8, p. e103853. ISSN 1932-6203. ECollection 2014. Disponível em: <<https://doi.org/10.1371/journal.pone.0103853>>.

ZHU, M. et al. **A Novel Microcontroller-Based System for the Wheel-Running Activity in Mice**. eNeuro, 2021. v. 8. ENEURO.0260-21.2021 p. EPublished: 2021 Nov 22; Print 2021 Nov-Dec. Disponível em: <<https://doi.org/10.1523/ENEURO.0260-21.2021>>.