



Universidade Federal de Ouro Preto - UFOP
Escola de Minas
Colegiado do curso de Engenharia de
Controle e Automação - CECAU



Danilo Duarte Lisboa

Aperfeiçoamento do Sistema de Monitoramento de Biotérios da UFOP - Medição de ITGU e Utilização de Visão Computacional

Monografia de Graduação em Engenharia de Controle e Automação

Ouro Preto, 2017

Danilo Duarte Lisboa

**Aperfeiçoamento do Sistema de Monitoramento de Biotérios da
UFOP - Medição de ITGU e Utilização de Visão Computacional**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo

Coorientador: Prof. Dr. José Vítor Nicacio

Ouro Preto, 2017

L769a Lisboa, Danilo Duarte.
Aperfeiçoamento do sistema de monitoramento de biotérios da UFOP -
medição de ITGU e utilização de visão computacional [manuscrito] / Danilo
Duarte Lisboa. - 2017.

73f.: il.: color; tabs; algoritmos.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo.
Coorientador: Prof. Dr. José Vítor Nicacio.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de
Minas. Departamento de Engenharia de Controle e Automação e Técnicas
Fundamentais.

1. Biotérios. 2. Monitoramento ambiental. 3. Processamento de imagens
auxiliado por computador. 4. Microprocessadores. I. Segundo, Alan Kardek
Rêgo. II. Nicacio, José Vítor. III. Universidade Federal de Ouro Preto. IV.
Título.

Catálogo: ficha@sisbin.ufop.br

CDU: 681.5

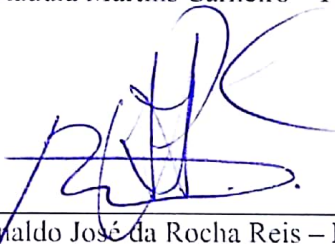
Monografia defendida e aprovada, em 11 de dezembro de 2017, pela comissão avaliadora constituída pelos professores:

A complex, stylized handwritten signature in blue ink, featuring multiple loops and a long horizontal stroke extending to the right.

Prof. Dr. Alan Kardek Rêgo Segundo – Orientador

A handwritten signature in blue ink, consisting of several loops and a long horizontal stroke.

Profa. Dra. Cláudia Martins Carneiro – Professora Convidada

A handwritten signature in blue ink, featuring a large, bold 'R' and several loops.

Prof. Dr. Agnaldo José da Rocha Reis – Professor Convidado

Resumo

Como toda pesquisa, em especial nas áreas das ciências biológicas e da saúde, deve-se obedecer normas e padrões a fim de garantir a saúde dos humanos e animais envolvidos, tornando um estudo válido ao ter referências durante os procedimentos. Além de determinar um ambiente saudável, as condições ideais de temperatura, umidade, luminosidade, ruído, nível de amônia e conforto térmico influenciam no comportamento do animal utilizado para o experimento. Comportamentos indesejados desse animal, provenientes de más condições ambientais, podem levar a falsos diagnósticos de doenças, ou à alegações incorretas sobre os efeitos colaterais de um determinado fármaco em estudo. O presente trabalho é uma continuação ao desenvolvimento de um sistema de monitoramento sem fio para biotérios, que registra valores de temperatura, umidade relativa do ar, luminosidade e níveis de ruído e de amônia. O sistema compara os valores obtidos com os de referência e os registra em um banco de dados. Para isso, foram utilizadas as plataformas Arduino para realizar a aquisição dos dados e o Visual Studio para desenvolver o programa supervisorio, em linguagem C#. A primeiro momento, foi realizado o aperfeiçoamento do sistema de monitoramento, em que foram feitos avanços na aquisição e transmissão dos valores, nos *softwares* e banco de dados, bem como no desenvolvimento de um protótipo encapsulado que pôde ser usado definitivamente dentro do biotério. O protótipo final foi replicado para o monitoramento de múltiplas salas do biotério. Como proposta de ampliação, foi incorporado ao projeto um sensor de globo negro para o cálculo do índice de temperatura de globo negro e umidade (ITGU). Também foi aplicado uma técnica de visão computacional para reconhecimento de padrões e atividade dos roedores. Somando-se essas ferramentas adicionadas, será possível, futuramente, determinar o conforto térmico dos animais estudados em uma única plataforma. O sistema integrado também poderá ser utilizado para o estudo comportamental dos ratos e camundongos do biotério com a comparação das condições ambientais.

Palavras-chaves: Biotério, Monitoramento, Arduino, Supervisorio, ITGU, Visão Computacional.

Abstract

Like all research, especially in the areas of life sciences and health, standards must be followed to ensure the health of humans and animals involved, making a valid study by having references during procedures. In addition to determining a healthy environment, ideal conditions of temperature, humidity, luminosity, noise, ammonia level and thermal comfort influence the behavior of the animal used for the experiment. Unwanted animal behaviors from poor environmental conditions may lead to false diagnosis of disease or incorrect claims about the side effects of a particular drug under study. The present work is a continuation of the development of a wireless monitoring system for animal houses, which records values of temperature, relative humidity, luminosity, noise and ammonia level. The system compares the values obtained with the reference values and registers them in a database. For this, was used the Arduino platform to perform data acquisition and Visual Studio to develop the supervisory program, in C# language. At first moment, the improvement of the monitoring system was carried out, in which progress was made in the acquisition and transmission of values, in the software and database, as well as in the development of an encapsulated prototype that could definitely be used inside the animal houses. The final prototype was replicated for the monitoring of multiple rooms of the animal houses. As a proposal for expansion, a black globe sensor for calculating the black globe temperature and humidity index (BGHI) was incorporated into the project. A computer vision technique for pattern recognition of rodent activity was also applied. Adding these tools, it will be possible, in the future, to determine the thermal comfort of the animals studied on a single platform. The integrated system may also be used for the behavioral study of rats and mice of the animal houses with the comparison of the environmental conditions.

Keywords: Animal houses, Monitoring, Arduino, Supervisory, BGHI, Computer Vision.

Lista de ilustrações

Figura 1	Arduino Nano	19
Figura 2	Exemplo Blink da IDE Arduino	20
Figura 3	Criando projeto no Microsoft Visual Studio 2017	21
Figura 4	Protótipo do projeto	24
Figura 5	Protótipo do projeto	25
Figura 6	Tela do sistema em funcionamento	26
Figura 7	Módulo emissor e receptor	27
Figura 8	Tela principal e de dados do <i>software</i> desenvolvido	27
Figura 9	Diagrama do projeto	28
Figura 10	Microfone de eletreto CZN-15E	31
Figura 11	Esquema eletrônico do projeto	31
Figura 12	Representação do sinal de ruído de máxima amplitude	32
Figura 13	Dados e curva de calibração do sensor de intensidade sonora	34
Figura 14	Sensor LDR	34
Figura 15	Circuito do LDR	35
Figura 16	Curva de comportamento do LDR	36
Figura 17	Dados e curva de calibração do LDR	36
Figura 18	Sensor de temperatura e umidade relativa DHT11	37
Figura 19	Sensor de amônia MQ135	38
Figura 20	Módulo NRF24l01+ com amplificador e antena externa	39
Figura 21	Esquema eletrônico do módulo emissor	41
Figura 22	Esquema eletrônico do módulo receptor	41
Figura 23	Layout da placa de transmissão	43
Figura 24	Layout da placa de recepção	43
Figura 25	Módulo emissor aberto	44
Figura 26	Módulo emissor	45
Figura 27	Módulo receptor	45
Figura 28	Tela de programação do projeto	47
Figura 29	Tela principal do programa	48
Figura 30	Mensagens de conexão com o banco de dados	49
Figura 31	Janela de histórico	50
Figura 32	Exemplo de busca com média horária	50
Figura 33	Módulos transmissores em funcionamento	51

Figura 34	Módulo receptor em funcionamento	52
Figura 35	Gráfico de teor de amônia(ppm) x tempo	53
Figura 36	Gráfico de Iluminância(lux) x tempo	54
Figura 37	Gráfico de temperatura(°C) x tempo	54
Figura 38	Gráfico de umidade relativa do ar(%) x tempo	55
Figura 39	Gráfico de ruído(dB) x tempo	55
Figura 40	Globo negro	57
Figura 41	Conexão do globo negro	57
Figura 42	Gráfico de média horária do ITGU calculado	58
Figura 43	Gráfico de média diária do ITGU calculado	59
Figura 44	Configuração dos Testes em Campo Aberto	61
Figura 45	Janela de análise comportamental	62
Figura 46	Software em execução	66
Figura 47	Parte da tabela gerada do teste realizado	67
Figura 48	Gráficos de ruído e velocidade do hamster em função do tempo	68

Lista de Algoritmos

1	Função de calculo da intensidade sonora	33
2	Protocolo de envio de dados sem fio	46
3	Função de rastreo do animal	65

Sumário

1	Introdução	10
1.1	Aplicação: análise comportamental animal	11
1.2	Objetivos	12
1.2.1	Objetivos Específicos	12
1.3	Motivação	13
1.4	Organização do trabalho	13
2	Revisão Bibliográfica	15
2.1	Biotérios	15
2.2	Parâmetros ambientais	16
2.2.1	Conforto Térmico	17
2.3	Arduino	18
2.4	Sensores e transdutores	20
2.4.1	Analógico e digital	21
2.5	Visual Studio	21
2.6	OpenCV e Emgu CV	22
2.6.1	Trabalhos relacionados	23
2.7	Histórico do projeto	25
3	Aperfeiçoamento do sistema de monitoramento	28
3.1	Apresentação do Sistema	28
3.2	Materiais Utilizados	29
3.3	Sensores e calibração	30
3.3.1	Ruído - Microfone de eletreto CZN-15E	30
3.3.2	Iluminância - LDR	34
3.3.3	Temperatura e umidade - DHT11	37
3.3.4	Índice de amônia - MQ135	38
3.4	Módulo rádio - NRF24L01	39
3.5	Esquemático do circuito	40
3.6	Layout da placa	42
3.7	Prototipagem dos módulos	44
3.8	Software	45
3.8.1	Supervisório	47
4	Resultados da Primeira Etapa	51

5	Índice de Temperatura de Globo Negro e Umidade	56
5.1	Implementação do globo negro	56
5.2	Cálculo do ITGU	58
6	Visão Computacional para Estudo Comportamental Animal	60
6.1	Materiais e montagem do experimento	60
6.2	Software	62
6.3	Funcionamento e implementações	63
6.4	Realização do experimento	66
6.5	Resultados preliminares	67
7	Conclusão	69
	Referências	71

1 Introdução

Biotérios são definidos por Cardoso (2001) como áreas destinadas à criação e à manutenção de animais de laboratório em condições sanitárias dentro de padrões rigorosamente estabelecidos, respeitando as normas éticas e as leis de manipulação e vivissecção. Como toda pesquisa, em especial nas áreas das ciências biológicas e da saúde, deve-se obedecer essas normas e padrões a fim de garantir a saúde dos humanos e animais envolvidos e tornar um estudo válido, ao ter base e valores de referências durante os procedimentos.

Pensando nisso, o projeto de um sistema de monitoramento de biotérios foi criado em 2014 a partir de uma iniciativa de alunos e professores da universidade. Os biotérios em que foram realizados os estudo e desenvolvimento do projeto foram os de ratos e camundongos do Centro de Ciência Animal (CCA) da Universidade Federal de Ouro Preto (UFOP), que apoiou em sua validação e financiamento. O trabalho foi elaborado com o objetivo de criar uma plataforma que monitore os principais parâmetros ambientais de um biotério. O projeto deu início por Sanchez (2014) e Pereira (2014), que realizaram um estudo detalhado dos parâmetros relevantes a serem monitorados. A partir de pesquisas na literatura e acervos bibliográficos, determinaram as variáveis: temperatura, umidade relativa do ar, luminosidade, ruído e índices de amônia no ar. Em que os valores medidos devem ser comparados com os de referência pesquisados. Em 2015, Jesus (2017) deu continuidade ao trabalho. O autor teve como principal objetivo a implementação de comunicação *wireless* ao projeto. No mais, toda a pesquisa realizada anteriormente ao presente trabalho será melhor apresentada e referenciada no próximo capítulo.

O presente trabalho é uma continuação ao desenvolvimento de um sistema de monitoramento de biotérios iniciado pelos autores citados no parágrafo anterior. Foi tema de iniciação científica do autor, que propôs melhorias para o projeto, em segundo momento, sua expansão e aplicação. Neste trabalho foram desenvolvidos alguns aprimoramentos na rede de comunicação sem fio, na utilização e calibração dos sensores, no *software* supervisor e no banco de dados. Além disso, foi criado um circuito especial para mensurar o nível de ruído, devido ao não funcionamento do sensor empregado anteriormente. O projeto funcionava com o circuito exposto, então o desenvolvimento do primeiro protótipo encapsulado também foi abordado neste trabalho, para que possa ser definitivamente utilizado dentro do biotério.

O presente trabalho segue o conceito de *OpenSource*, ou Código Aberto, onde pode-se estudar, modificar e distribuir o software gratuitamente. O sistema é baseado em

plataforma Arduino com interface programada em C# no *Visual Studio 2017 Community*. Tem uma proposta de baixo custo, que opera com sensores e demais dispositivos eletrônicos bem acessíveis no mercado comum.

Neste trabalho também será apresentada a implementação ao projeto de um sensor de globo negro desenvolvido por Nicacio (). Utilizando o sensor de umidade relativa do ar já presente no sistema de monitoramento, mais a temperatura de globo negro, pretende-se obter o índice de temperatura de globo e umidade (ITGU). Na literatura, o ITGU é utilizado para determinar faixa de de conforto térmico dos animais no ambiente monitorado.

1.1 Aplicação: análise comportamental animal

Estudos realizados por Majerowicz et al. (2005) demonstram a importância do controle dos principais parâmetros ambientais de um biotério e a interferência que eles podem causar no comportamento animal. O estudo do comportamento de animais de laboratório é uma ferramenta muito utilizada para auxílio do diagnóstico de patologias orgânicas e psicológicas. Além disso, a partir da observação do comportamento desses animais, é possível obter os efeitos de fármacos que estão em fase de teste ao organismo. Com tais informações pode-se prever os possíveis efeitos físicos e psíquicos em uma posterior utilização em humanos.

"Nos organismos intactos, a variabilidade comportamental pode ser função de processos decorrentes da interação do organismo com seu meio ambiente"(HUNZIKER; MORENO, 2000). Isso significa que, mesmo em um indivíduo saudável, pode haver alteração em seu comportamento devido à condições ambientais inadequadas. Essa interferência podem levar a falsas deduções nos efeitos do fármaco em estudo. O método de análise comportamental do animal estudado também pode comprometer os resultados alcançados. Uma observação feita através da visão humana não é capaz de distinguir padrões com uma complexidade elevada. Se os critérios de avaliação dessa pessoa não estiverem bem definidos, pode levar a falsos diagnósticos.

Com o primeiro protótipo do sistema de monitoramento de parâmetros ambientais de biotérios desenvolvido, pretende-se introduzir o estudo de análise comportamental automatizada do roedor com comparação de parâmetros ambientais. Para isso será utilizado processamento de imagens de uma câmera comum e a biblioteca *OpenCV*. A comunicação da biblioteca com o *software* desenvolvido em C# é estabelecida através do pacote *Emgu CV*, que permite que as funções do OpenCV sejam chamadas a partir de idiomas compatíveis com o .NET, como C#, VB, VC++, IronPython, etc. Assim, o código poderá ser compilado em C# no Visual Studio. O *software* final deve ser integrado ao de monitoramento, em que será possível relacionar os parâmetros ambientais com os dados do processamento de

imagens, como distância percorrida e velocidade do animal.

Com o cálculo do ITGU e a implementação da técnica de visão computacional, será possível, futuramente, determinar o conforto térmico para uma determinada espécie animal. Na literatura há pouco estudo sobre o cálculo do ITGU adequado para os animais dos biotérios em estudo (ratos e camundongos). Isso se deve ao fato desses animais não serem criados com um propósito de maximizar a produção para consumo, como bovinos e suínos por exemplo. Porém, como abordado anteriormente, o conforto do animal em estudo pode comprometer o resultado da pesquisa.

1.2 Objetivos

O presente trabalho tem o objetivo de dar continuidade ao projeto de desenvolvimento de um sistema de monitoramento de biotério de baixo custo e com plataforma aberta. O sistema deve atender aos requisitos de monitoramento de um biotério estudados na literatura e realizar a coleta de dados automatizada dos parâmetros ambientais de temperatura de bulbo seco, temperatura de globo negro, umidade relativa do ar, luminosidade, nível de ruído e teor de amônia no ar.

Este trabalho também tem como objetivo o estudo de uma aplicação de visão computacional para o sistema de monitoramento. Será desenvolvido no *software*, um módulo de processamento de imagens para o estudo comportamental dos roedores. O sistema de análise de imagens deverá ser integrado, de forma que compare os dados ambientais com os de comportamento.

1.2.1 Objetivos Específicos

São listados aqui, os objetivos específicos para o sistema de monitoramento:

- Prototipar os módulos emissor e receptor com novas adaptações.
- Recalibração dos sensores.
- Aprimorar os *softwares* do sistemas.
- Realizar testes e validação do sistema de monitoramento

Concluídos os objetivos listados acima, o presente trabalho teve como objetivo:

- Calcular o ITGU e índice de conforto térmico para o ambiente.
- Rastrear posição e movimentação do roedor utilizando processamento de imagem.
- Implementar interface de rastreamento ao sistema de monitoramento.

- Realizar teste preliminares
- Buscar correlação entre comportamento do roedor e parâmetro ambiental.

1.3 Motivação

Atualmente no biotério da UFOP, não há controle eficaz das variáveis ambientais que este trabalho estuda. O único controle de valores que é feito é da temperatura, onde os técnicos e pesquisadores entram periodicamente em cada sala do biotério para checar e anotar a temperatura de um termômetro analógico de mercúrio em um caderneta. Depois ainda, os valores são repassados para uma planilha no computador, tudo feito manualmente. Mas fique claro que o biotério segue cuidadosamente os procedimentos para manter os níveis de todos parâmetros adequados, como por exemplo o uso de gaiolas ventiladas e sua higienização regularmente para manter o nível de amônia no ar abaixo dos limites prejudiciais. Porém, os valores não são mensurados ou medidos de forma ineficiente como o da temperatura.

Existem no mercado sistemas de monitoramento de parâmetros ambientais que podem ser utilizados em biotérios, mas além de serem muito caro, não há um único produto que monitore todas as variáveis que são estudadas neste trabalho ao mesmo tempo, poupando investimento e espaço no biotério. Isso justifica o principal objetivos do projeto, que é desenvolver um sistema de monitoramento sem fio, de baixo custo e com plataforma aberta. Dessa forma, além do produto ser mais acessível financeiramente, favorece a pesquisa na área da saúde e tecnologia. Por ser de plataforma aberta, pode ser constantemente melhorado como vem sendo pelos alunos e professores da UFOP.

A expansão do trabalho com o cálculo do ITGU e implementação de técnicas de visão computacional além de trazerem os benefícios citados, gera motivação à continuação da pesquisa. O que justifica que presente trabalho aborda novos assuntos e técnicas, mas não chega ao grande objetivo final, pois demanda um longo tempo de pesquisa.

1.4 Organização do trabalho

Este trabalho está organizado de forma a distinguir duas grandes etapas do projeto:

- Etapa 1: Aperfeiçoamento do sistema de monitoramento de parâmetros ambientais.
- Etapa 2: Adição do sensor de globo negro com cálculo do ITGU e aplicação de visão computacional para análise comportamental animal.

No Capítulo 1 foi feita uma breve introdução ao tema abordado e apresentado o projeto como uma continuação de trabalho anteriores.

No Capítulo 2 é feita uma revisão bibliográfica. O capítulo inicia com uma breve estudo sobre biotérios, os parâmetros ambientais e suas influências ao organismo. Segue com os assuntos abordados na monografia, trabalhos relacionados e histórico do projeto.

O Capítulo 3 apresenta como foi feito o aperfeiçoamento do sistema de monitoramento, que foi definido como primeira etapa do projeto.

O Capítulo 4 apresenta os resultados da primeira etapa, ou seja, do sistema de monitoramento de biotérios aperfeiçoado.

No Capítulo 5 é apresentado a implementação do sensor de globo negro para os cálculos do ITGU.

No Capítulo 6, a aplicação do processamento de imagens de uma câmera para o estudo comportamental animal.

No Capítulo 7 são esclarecidas considerações finais e as conclusões sobre o trabalho.

2 Revisão Bibliográfica

2.1 Biotérios

Biotérios são definidos como áreas destinadas à criação e à manutenção de animais de laboratório em condições sanitárias, dentro de padrões rigorosamente estabelecidos, respeitando as normas éticas e as leis de manipulação e vivisseção. Eles podem ser separados segundo a finalidade a que se destinam. Os biotérios de criação são aqueles onde são produzidas e mantidas as matrizes das linhagens, com controle rigoroso da saúde dos animais e esquemas especiais de cruzamentos para manutenção das características genéticas, assegurando os padrões de qualidade. Já os biotérios de produção são aqueles onde se criam grandes quantidades de animais para atender às pesquisas, recebendo matrizes dos biotérios de criação e, por fim, os biotérios de experimentação que se destinam a receber os animais dos biotérios de produção para utilizá-los na experimentação (CARDOSO, 2001).

A utilização dos biotérios é feita por grandes indústrias, universidades e outros centros de pesquisa que também requerem animais de boa procedência para a condução de seus experimentos. Segundo Politi et al. (2008), a condição sanitária é determinada por fatores como a origem dos animais e meio ambiente em que estão vivendo, os quais afetam diretamente suas respostas biológicas, fisiológicas e comportamentais. Assim, instalações adequadas para os padrões necessários à criação e manutenção de animais de laboratório devem possuir programas de gerenciamento das condições físicas e ambientais, garantindo cuidados que favoreçam o desenvolvimento e a reprodução dos animais, mantendo o bem-estar, bem como minimizando as variações que podem interferir nos resultados dos ensaios.

Em biotérios, é mais comum estudos com os ratos e camundongos. O motivo por qual esses roedores são populares no campo de pesquisa, são por que eles são mamíferos assim como nós e são pequenos e de fácil de procriação. Além disso, a sequência de genoma de ratos e camundongos são bem determinadas (MAJEROWICZ et al., 2005).

O presente trabalho estuda as condições ambientais adequadas à esses animais. Quando é realizado um monitoramento, deve-se ter valores de referência a serem comparados para a validação de uma pesquisa, especialmente com animais e na área da saúde. Este estudo então, se torna de grande importância.

2.2 Parâmetros ambientais

Tomando base nos métodos que este trabalho tem a apresentar, estudos dirigidos por Politi et al. (2008) determinaram que os fatores físicos de um ambiente, tais como temperatura, umidade relativa do ar, gases e substâncias particuladas, iluminação e ruídos interferem nas respostas biológicas dos animais selecionados para pesquisas laboratoriais. São apresentados a seguir, a influência que tais variáveis ambientais podem causar ao organismo dos animais em estudo e aos pesquisadores quando expostos a condições inapropriadas.

Para evitar situações de estresse ou desconforto térmico, que levariam à invalidação do experimento científico com animais, a temperatura para camundongos e ratos deve ser de 19 a 23°C. Essa faixa de temperatura também é ideal para os profissionais que trabalham nos biotérios, que devem sempre munidos de equipamentos de proteção individual (MAJEROWICZ et al., 2005).

A taquipneia (aceleração do ritmo respiratório) é um mecanismo empregado pela maioria dos animais de laboratório para compensar a falta ou baixa sudoração. Porém, o processo de taquipneia com a umidade relativa inadequada interfere na troca de calor e água com o ambiente. Recomenda-se então, uma faixa de umidade relativa do ar de $(55 \pm 15)\%$, especialmente nos casos dos ratos e camundongos, sendo que a umidade nas salas é mantida por meio de aparelhos de ar condicionado. Variações e extremos na umidade relativa do ar podem favorecer o aparecimento de doenças e alterar o consumo de ração e água (POLITI et al., 2008).

O ambiente deve ser bem ventilado e higienizado, caso contrário, produtos metabólicos começarão a se acumular nas gaiolas e no ar, prejudicando a saúde dos animais e pessoas envolvidas. O acúmulo de amônia, por exemplo, produto nitrogenado oriundo da ação de bactérias urease positivas sobre as excretas dos ratos, pode afetar o sistema respiratório destes animais (MAJEROWICZ et al., 2005). O número de trocas e higienizações realizadas nas gaiolas influencia na concentração de gases e na umidade relativa do ambiente. Entretanto, ainda não foi estabelecido o número de trocas ideal para as espécies de roedores de laboratório. Se faz necessário a higienização regular do ambiente e gaiolas.

Os roedores em geral possuem hábitos noturnos e são bem sensíveis a altas intensidades luminosas. Segundo Majerowicz et al. (2005), uma iluminação que propicie boa visibilidade, seja uniforme e o mais próxima possível da luz natural, é adequada à reprodução e assegura um comportamento normal para a maioria dos roedores. A iluminação dos biotérios são com lâmpadas fluorescentes com temperatura de cor que representa a luz solar. A intensidade de iluminação recomendada é de 250 lux a um metro do piso, com um período de 12 horas/dia. Nas outras 12 horas do dia é recomendado a ausência de luz.

O ruído proveniente do ambiente também pode afetar os animais. Assim, a exposição a sons acima de 85 dB pode causar tanto efeitos auditivos quanto sistêmicos, como: eosinopenia, aumento de peso das supra-renais, fertilidade reduzida em roedores (CLOUGH, 1982).

Ratos submetidos a ruídos entre 55 e 95 dBA, apresentam alteração do comportamento alimentar devido a aumento da sinalização de alerta. "Eles poderão apresentar desde fome exagerada, até mesmo perda de apetite. Supõe-se que a perda de apetite esteja relacionada ao sistema de alerta simpato-adrenal adaptando o animal para o perigo" (CRUZ et al., 2010). Assim, o excesso de ruídos relacionadas às atividades humanas e de máquinas nos biotérios impedem a normalidade fisiológica e comportamental dos animais.

2.2.1 Conforto Térmico

A zona de conforto térmico varia de acordo com a espécie, idade, sexo, tamanho corporal, peso, estado fisiológico, estado produtivo, amplitude de temperatura ambiente, radiação, umidade e velocidade do ar (AMARAL et al., 2012). A primeira condição de conforto térmico dentro de uma instalação é que o balanço térmico seja nulo, ou seja, o calor produzido pelo organismo animal somado ao calor ganho do ambiente seja igual ao calor perdido pelos animais através da radiação, da convecção, da condução, da evaporação e do calor contido nas substâncias corporais eliminadas.

Para se avaliar o bem estar do animal em relação a sensação térmica, utiliza-se o índice de conforto térmico. Os índices mais usados são o de temperatura e umidade (ITU), que associa a temperatura de bulbo seco e a temperatura de bulbo úmido e o índice de temperatura de globo e umidade (ITGU), que considera a temperatura de globo negro e umidade do ar. O ITGU é um indicador mais preciso do conforto térmico, já que leva em considerações fatores como ventos e incidência solar (ESMAY et al., 1969).

O globo negro consiste de um sensor de temperatura em um interior de uma esfera, que geralmente é pintada com tinta preta fosca. O sensor é colocado próximo ao animal, simulando sua troca de calor ocorrida por ventilação e radiação (AMARAL et al., 2012).

Em seu projeto, Nicacio () desenvolveu um globo negro utilizando um sensor de temperatura analógico - LM35 e uma esfera plástica pintada de preto fosco. Seu sensor foi usado no presente trabalho juntamente com a umidade relativa do ar mensurada pelo DHT11 para calcular o ITGU do ambiente interno do biotério.

Vale observar que, calcular o ITGU não significa determinar o conforto térmico do animal no ambiente. Cada espécie possui uma reação diferente às condições ambientais, ou seja, o conforto térmico é estabelecido em diferentes índices de temperatura de globo e umidade - ITGU. Não foi encontrado na literatura a determinação do conforto térmico para os ratos e camundongos de laboratório. Esse parâmetro é mais difundido na área da

agronomia, onde se estudam os índices de conforto térmico para animais de corte ou de produção. Mas esse fato não desqualifica a importância deste estudo para animais voltados a pesquisa.

O propósito do presente trabalho é apenas implementar o cálculo do ITGU para futuras determinações de conforto térmico. Isso, devido a grande carga de trabalho que envolve a determinação da zona térmica de conforto, onde várias amostras de grupos de animais devem ser levadas em consideração.

2.3 Arduino

Segundo McRoberts (2011), o Arduino é um pequeno computador que pode ser programado para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele, é um plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. Atualmente existem diversos modelos oficiais e não oficiais da placa Arduino, isso é possível graças à sua plataforma aberta, o que significa que seu *software* e *hardware* podem ser utilizados livremente por qualquer pessoa e com qualquer propósito. O modelo do Arduino é basicamente definido pelo microcontrolador que está embarcado nele e suas conexões. Os principais modelos e seus respectivos microcontroladores são listados a seguir:

- Arduino UNO: ATmega328P;
- Arduino Leonardo: ATmega32u4;
- Arduino Nano: ATmega328 - SMD;
- Arduino Micro: ATmega32U4 - SMD;
- Arduino Mega: ATmega2560.

De acordo com Angnes (2003), microcontroladores são computadores que operam em sequência, sem qualquer intervenção humana, sendo essas operações previamente programadas por alguém com conhecimentos em linguagem de programação, como por exemplo: linguagem C, C++, Java ou Assembly. São conhecidos como computadores embutidos em CI's (circuitos integrados), contendo periféricos de entrada e saída, CPU e memória. Alguns ainda possuem conversores analógico/digital (ADC) e/ou digital/analógico (DAC). Ou seja, o microcontrolador é responsável basicamente por todo o processamento do Arduino.

Além do microcontrolador AVR da fabricante ATMEL, o Arduino compõe também de conectores, regulador de tensão 5V, um chip da Future Technology Devices International(FTDI) responsável pela comunicação USB/Serial, LEDs e outros periféricos que fazem do embarcado uma placa simples de ser utilizada por qualquer pessoa, bastando

conectá-lo ao computador e fazer a programação em seu ambiente de desenvolvimento integrado (IDE) nativa. O modelo utilizado no presente trabalho é o Arduino Nano (Figura 1), que já atende todos os requisitos do projeto, possui fácil conexão com placas de circuito impresso (PCI) e é a versão mais barata que pode ser encontrado facilmente no mercado.

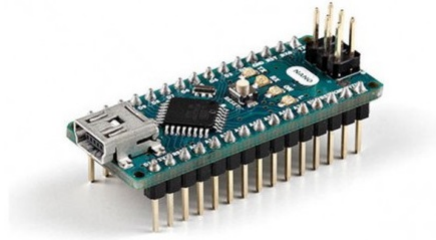


Figura 1 – Arduino Nano

Fonte: (ARDUINO, 2017)

O Arduino Nano tem várias facilidades para se comunicar com um computador, outro Arduino ou outros microcontroladores. O ATmega328 fornece a comunicação serial, que está disponível nos pinos digitais 0 (RX) e 1 (TX) e/ou com o computador por uma porta serial virtual. Os LEDs RX e TX na placa piscarão quando os dados estiverem sendo transmitidos através do chip FTDI e conexão USB para o computador. O ATmega328 também suporta comunicações I2C e SPI. Mais informações podem ser consultadas na pagina oficial do Arduino e em sua folha de dados (ARDUINO, 2017).

Como dito, o Arduino possui uma IDE de programação própria, um *software* livre com linguagem de programação baseada em C. A IDE permite escrever, compilar e carregar o programa para o Arduino. Os programas são conhecidos como *sketches* (rascunho, ou esboço). Uma das maiores facilidades fornecidas pelo Arduino é a disponibilidade de diversas bibliotecas e função pré-programadas, onde o programador não precisa ter um conhecimento específico dos registradores do microcontrolador. Esse tipo de programação é conhecida como programação de alto nível (MCROBERTS, 2011).

O exemplo de *sketch* mais simples disponível em sua IDE é o *Blink*, que faz com que o Arduino pisque seu LED incluso com um período de um segundo. Esse exemplo pode ser observado na Figura 2.

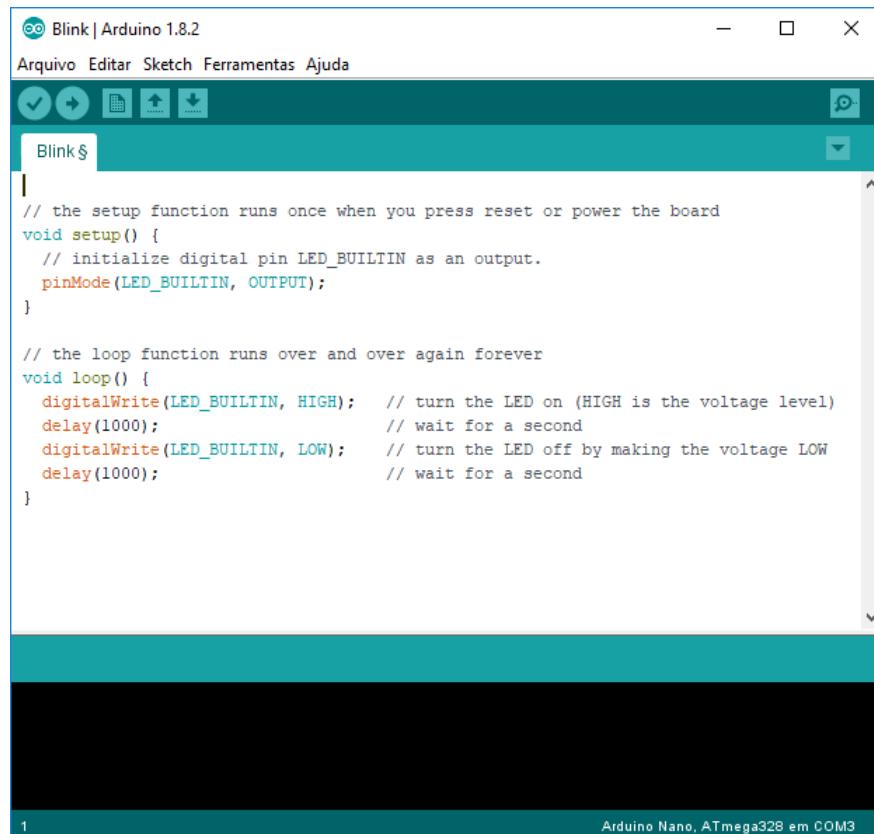


Figura 2 – Exemplo Blink da IDE Arduino

Fonte: Autor

2.4 Sensores e transdutores

Sensor é um dispositivo sensível à alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza física que precisa ser mensurada, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc. Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado por um circuito de interface antes da sua leitura no sistema de controle (WENDLING, 2010).

Sobre os transdutores, Wendling (2010) afirma que eles já transformam uma grandeza física em um sinal elétrico que pode ser interpretado por um sistema de controle. Muitas vezes os termos sensor e transdutor não são usados corretamente. O transdutor é o instrumento completo que engloba sensor e todos os circuitos de interface capazes de serem utilizados numa aplicação industrial. O amplificador operacional é um exemplo clássico para o tratamento de sinal dos sensores.

2.4.1 Analógico e digital

São distinguidos aqui as definições de sensores analógicos e sensores digitais. Os conceitos são abordados por Wendling (2010).

O sensor analógico pode assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que esteja dentro da sua faixa de operação. Essas variáveis são mensuradas por elementos sensíveis com circuitos eletrônicos não digitais.

O sensor digital por sua vez, pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, nível alto e baixo, que podem ser interpretados como zero ou um. Esse tipo de sensor não engloba apenas os que indicam um estado binário de uma variável (ligado ou desligado), mas pode comunicar com um microcontrolador ou computador através de uma sequência de estados lógicos, também conhecido como bits.

2.5 Visual Studio

O Microsoft Visual Studio em sua versão atual 2017 é um plataforma com interface amigável de programação com ferramentas de depuração, testes, compartilhamento online e muito mais (Figura 3). O pacote Visual Studio Community 2017 é de acesso gratuito e pode ser baixado pelo site da Microsoft. Ser uma plataforma integrada significa que em sua IDE é possível utilizar diversas linguagens de programação e com todos os recursos que tem direito (MICROSOFT, 2017).

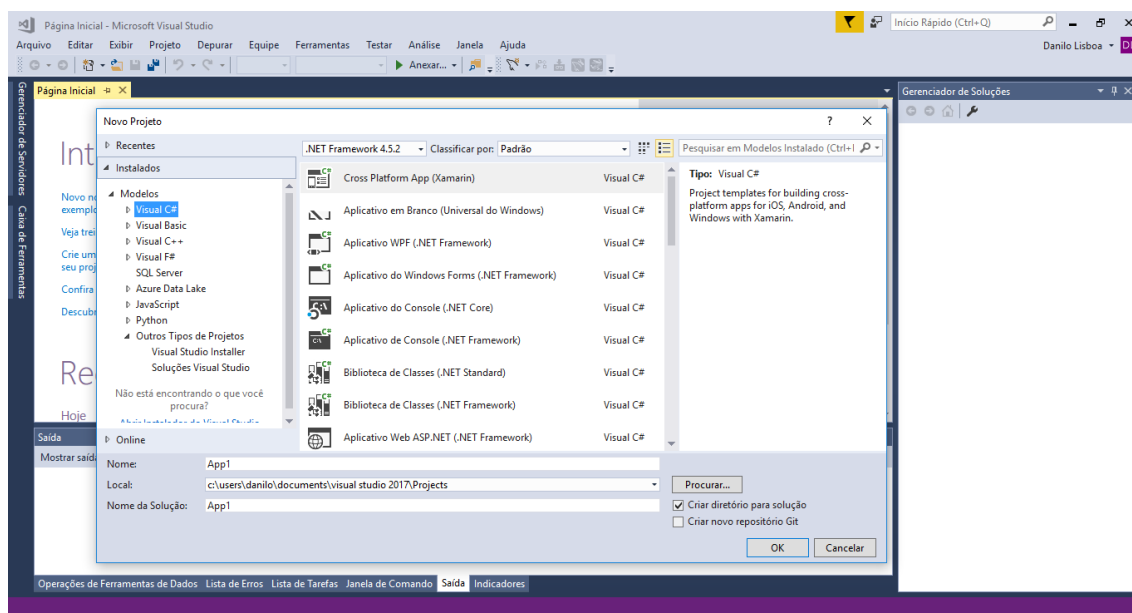


Figura 3 – Criando projeto no Microsoft Visual Studio 2017

Fonte: Autor

Na Figura 3 pode-se ver um dos modelos de programação do Visual C# (*C sharp*), uma linguagem de programação de alto nível e fácil compreensão voltada para plataformas *.NET Framework*. O C# também é conhecido como linguagem orientada a eventos, onde no projeto para aplicações para Windows, a programação é feita com janelas padrão do sistema operacional, podendo programar condições baseadas em eventos visuais, como cliques de botões ou troca de janelas, por exemplo.

No gerenciador de soluções, no canto direito da Figura 3 é possível criar múltiplos projetos e/ou biblioteca de classes que comunicam entre si. Essa técnica é muito utilizada na programação para organizar e estruturar códigos, de forma que o mesmo possa ser utilizado por mais de um programador. No final, o projeto pode ser publicado como um pacote de instalação contendo todos os recursos usados no programa.

A *Cross Platform App* ou plataforma cruzada, destacada na Figura 3, é uma grande tendência da programação. Esse recurso permite o compartilhamento de códigos para diversos ambientes como Windows, Linux, iOS e outros. A utilização do ambiente mobile está cada vez mais ampla, sendo a plataforma cruzada um ótimo recurso para desenvolvimento de aplicações mobile para Android, iOS e Windows (PELLIZZONI, 2017).

2.6 OpenCV e Emgu CV

O OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de *software* de aprendizagem e visão computacional de código aberto. Foi inicialmente desenvolvida pela Intel em 2000 e vem sendo melhorada a cada ano pela comunidade OpenCV. A biblioteca foi construída para fornecer uma infra-estrutura comum para aplicações de visão por computador e acelerar o uso da percepção da máquina nos produtos comerciais. É um produto com licença BSD, que permite que o *software* distribuído sob a licença, seja incorporado a produtos proprietários, facilitando a utilização e modificação do código pelas empresas (OPENCV, 2017).

O OpenCV é escrito de forma nativa em C++ e possui interfaces também em C, Python, Java e MATLAB. Suporta Windows, Linux, Android e Mac OS. A biblioteca possui mais de 2500 algoritmos otimizados. Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de câmera, rastrear objetos em movimento, extrair modelos de objetos 3D, juntar imagens para produzir uma alta resolução imagem de uma cena inteira, encontrar imagens semelhantes de um banco de dados de imagens, seguir os movimentos dos olhos e muitos outros recursos. A biblioteca é amplamente utilizada em empresas, grupos de pesquisa e órgãos governamentais.

Em 2008, é desenvolvido a Emgu CV, conhecido como uma *wrapper*, ou envoltório da biblioteca OpenCV para .Net C#. Permite que as funções OpenCV sejam chamadas a

partir de idiomas compatíveis com o .NET, como C#, VB, VC++ e outras. O *wrapper* pode ser compilado pelo Visual Studio, Xamarin Studio e Unity. Pode ser executado no Windows, Linux, Mac OS X, iOS, Android e Windows Phone (EMGUCV, 2017). Além disso, é uma plataforma cruzada, como foi definido no último parágrafo da seção 2.5.

Tendo como base de que o *software* de monitoramento desenvolvido para o presente trabalho foi todo estruturado em linguagem C# e de que o *software* de rastreamento automático proposto neste trabalho deve ser integrado ao primeiro, se faz conveniente o uso do Emgu CV. Dessa forma será possível fazer toda a programação em uma única interface e solução de projeto.

O Emgu CV pode ser baixado gratuitamente através do site na referência (EMGUCV, 2017). Seu pacote de instalação já inclui a biblioteca OpenCV, não precisando ser baixado separadamente. Para integrar o Emgu CV ao projeto, basta que, após ser instalado no computador, incluir suas dlls nas referências da solução de programação. A versão utilizada no presente trabalho foi a 3.2.0 que já dá suporte ao Microsoft Visual Studio 2017.

2.6.1 Trabalhos relacionados

Os autores Charoenpong et al. (2012) desenvolveram um experimento que analisa a distância e duração de determinado comportamento do rato em um labirinto elevado e em um modelo de tabuleiro. O procedimento do autor é feito utilizando apenas uma câmera comum e programa desenvolvido utilizando a biblioteca de visão computacional OpenCV, com intuito de retornar bons resultados, mas sem custo considerável.

Já em Zhang et al. (2016), além da câmera e utilização do OpenCV como processamento de imagem, é colocado nas costas do rato uma pequena mochila com o sensor MPU6050 que sincroniza os dados de orientação e aceleração do roedor com o *software* de rastreamento. Apesar do método ser considerado não evasivo, esse tipo de intervenção ao animal pode comprometer o seu comportamento segundo Hunziker e Moreno (2000).

No trabalho de Ishii et al. (2007), o estudo do comportamento do roedor é realizado em uma arena quadrada de 1100mm em campo aberto. Os autores, como o próprio título do artigo sugere, montam uma configuração experimental autônoma para análise de comportamento de ratos mais elaborada. O rastreamento do animal é feito por uma câmera acoplada a um braço robótico de três graus de liberdade, que junto ao *software* segue a locomoção do rato com uma certa previsão de movimento. Células de carga são colocadas no solo em locais estratégicos para detectar comportamentos do tipo levantar, roer e cheirar, que produzem vibrações específicas. Além disso, é utilizado também LEDs infravermelhos e fototransistores que detectam a altura referencial do rato com o solo.

No trabalho de Shi et al. (2010), o teste é também realizado em uma arena

quadrada de campo aberto, porém um pouco menor (700mm). Além do rato, é colocado na arena de testes um robô com o intuito de interagir com o animal, livre de intervenção humana. É demonstrado alguns algoritmos de reconhecimento do rato e de determinação de orientação utilizando a imagem da câmera apenas. Esse trabalho mostra que é possível tais processamentos de imagem sem a utilização de sensores como nos trabalhos de Zhang et al. (2016) e Ishii et al. (2007).

Para dar início ao processamento de uma imagem, primeiramente é selecionado a região de interesse, excluindo bordas ou áreas indesejadas. O segundo passo seria determinar outra variável a ser manipulada, seguindo a lógica do primeiro passo. Como exemplo, no rastreamento do animal realizado por Charoenpong et al. (2012), se determina um limite de cores que devem ser consideradas ou não. A imagem então é binarizada de acordo com essa seleção. Em seguida, é aplicado filtros para eliminar os demais ruídos da imagem. Na Figura 4 demonstra o resultado dessa técnica.

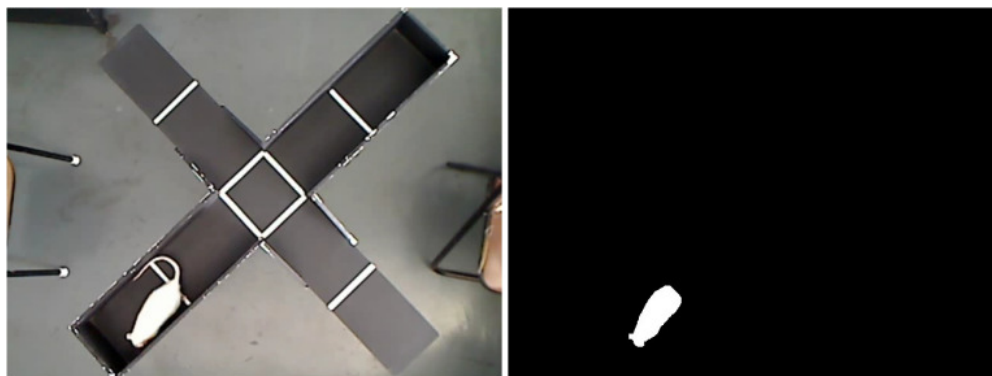


Figura 4 – Protótipo do projeto

Fonte: (CHAROENPONG et al., 2012)

As possibilidades de processamento de imagens são infinitas. A biblioteca OpenCV possui funções intuitivas que ajudam na experiência do programador. Sem essa facilidade, seria necessário o conhecimento específico em manipulações matriciais com equações complexas. Claro que, um bom programador deve saber ao menos o que ocorre por trás dessas funções. Assim é possível resolver problemas mais elaborados e criar soluções inovadoras. De qualquer forma, usufruir conscientemente dos recursos que simplificam o projeto, sempre têm ajudado no desenvolvimento da pesquisa.

2.7 Histórico do projeto

O projeto de monitoramento de biotérios da UFOP foi iniciado em 2014, com a iniciativa de uma aluna e professor do curso de Engenharia de Controle e Automação em parceria ao Centro de Ciência Animal da Universidade. Em sua monografia, Sanchez (2014) expõem com mais detalhes como foram escolhidos temperatura, umidade relativa do ar, luminosidade, ruído e teor de amônia para compor o monitoramento do ambiente. Em seguida descreve o funcionamento dos sensores que irão medir tais parâmetros, sendo que foram escolhidos com o intuito de serem de baixo custo e utilizados com plataforma Arduino.

Sanchez (2014) utilizou o Arduino Uno para dar início a pesquisa, uma placa que possui entradas e saídas de fácil acesso e diversas opções de *Shields* no mercado, que permitem expandir seus recursos e conexões. Utilizou os sensores: DHT11 para a temperatura e umidade relativa do ar; fotorresistor (LDR) para calcular a iluminância; MQ-135 para o teor de amônia. Houve uma tentativa de utilizar um módulo de microfone de eletreto comercial para medir a intensidade sonora, mas o sensor não funcionou como deveria. Seu protótipo final foi elaborado em uma *Shild com protoboard* para o Arduino Uno, que está representado na Figura 5.

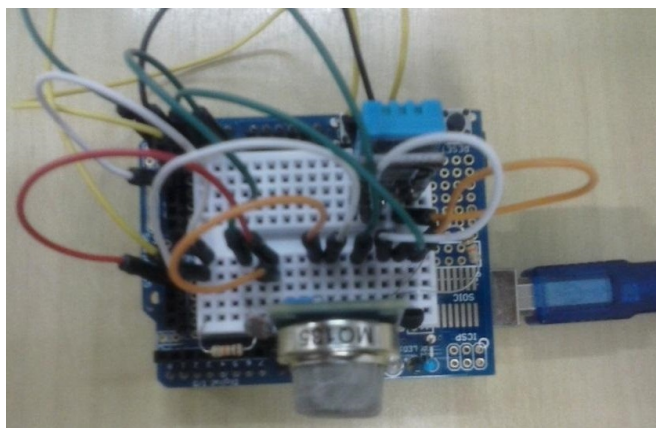


Figura 5 – Protótipo do projeto

Fonte: (SANCHEZ, 2014)

O *software* foi desenvolvido pela autora na plataforma *Visual Studio 2012*, ainda em uma versão simples. Foram dispostos quatro gauges para representar os valores de temperatura, umidade relativa do ar, luminosidade e teor de amônia. O parâmetro de ruído foi desconsiderado pela autora, pois não foi possível medir a intensidade sonora com o sensor escolhido. Em sua interface também é possível salvar os valores adquiridos em um arquivo de texto, mas não foi implementado um banco de dados que assegure as informações em tempo real. Na Figura 6 está representado sua versão final da interface desenvolvida.

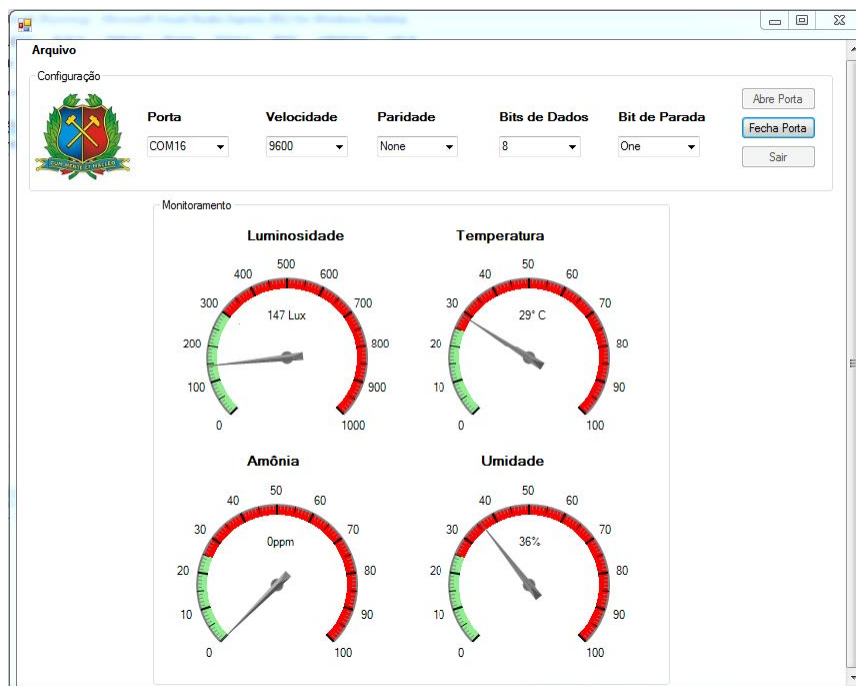


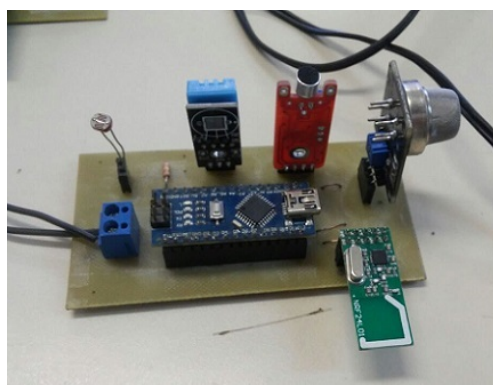
Figura 6 – Tela do sistema em funcionamento

Fonte: (SANCHEZ, 2014)

Jesus (2017) deu continuidade ao trabalho em 2015. O autor propôs melhorias no *software e hardware* do projeto de monitoramento, com o objetivo principal de implementar a comunicação *wireless*. Para isso, aplicou o NRF24L01, um módulo rádio com comunicação SPI que segue a proposta geral do projeto de ser de baixo custo e *OpenSource*.

O autor utilizou o Arduino Nano com o propósito de reduzir o tamanho do protótipo final e facilitar seu acoplamento em uma placa de circuito impresso especificamente confeccionada. Nesta versão, já são dois módulos para o sistema de monitoramento. Um emissor, com os sensores acoplados para realizar as medições e transmitir ao módulo receptor. Esse último módulo recebe os valores de temperatura, umidade, luminosidade, amônia e ruído através do NRF24L01 e transmite ao *software* por uma porta USB do computador utilizando a comunicação Serial/USB presente no Arduino. O módulo emissor é alimentado por uma fonte externa de 5V. O protótipo final é representado na Figura 7.

Em seu projeto, Jesus (2017) desenvolveu o *Software* na plataforma C# no *Visual Studio 2015* (Figura 8). A versão final possui um gráfico que permite acompanhar em tempo real o parâmetro ambiental escolhido. Um banco de dados foi implementado, permitindo que o usuário acesse através da janela "Historic" os dados coletados anteriormente com um filtro de busca. A seleção de salas é uma futura expansão do projeto, onde será possível receber a informação de vários módulos emissores em salas diferentes.



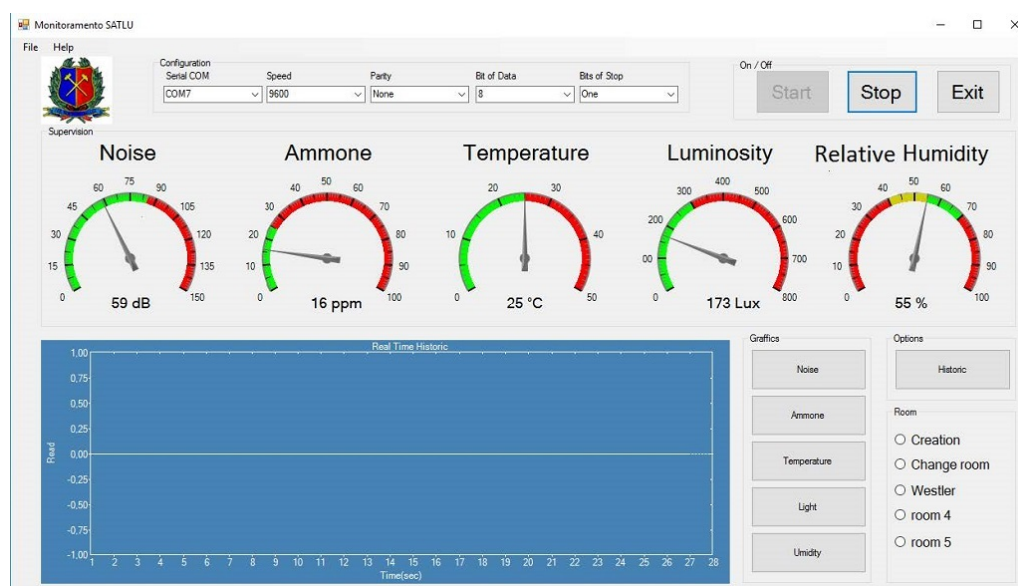
(a) Emissor



(b) Receptor

Figura 7 – Módulo emissor e receptor

Fonte: (JESUS, 2017)



(a) Tela principal

ID	Date	Time	Lux	Temperature	Humidity	Noise	Ammonia
211	06/02/2017	14:32:20	23	27	64	45	17
212	06/02/2017	14:32:22	23	27	64	45	16
213	06/02/2017	14:32:25	23	27	64	45	13
214	06/02/2017	14:32:27	23	27	64	45	12
215	06/02/2017	14:32:29	23	27	64	45	11
216	06/02/2017	14:32:31	23	27	64	45	10
217	06/02/2017	14:32:34	23	27	64	45	10
218	06/02/2017	14:32:36	23	27	64	45	9
219	06/02/2017	14:32:38	23	27	63	45	9
220	06/02/2017	14:32:40	23	27	63	45	9
221	06/02/2017	14:32:43	23	27	63	45	8

(b) Tela de dados

Figura 8 – Tela principal e de dados do *software* desenvolvido

Fonte: (JESUS, 2017)

3 Aperfeiçoamento do sistema de monitoramento

3.1 Apresentação do Sistema

Para descrever a primeira etapa do projeto, primeiramente será abordado os materiais utilizados, tratado do *hardware*, que compõe dos dispositivos eletrônicos e a construção do protótipo de monitoramento. Em seguida, será apresentado com mais detalhes os *softwares* utilizados e desenvolvidos.

O sistema é formado por módulos transmissores e um receptor. Os transmissores fazem as leituras dos parâmetros ambientais em salas distintas do biotério, podendo ser de criação, produção ou experimentação. O módulo receptor está conectado ao computador do escritório do biotério, que recebe as informações dos cinco módulos transmissores e repassa ao supervisor instalado na máquina via comunicação serial simulada por uma porta USB. O sistema é representado em forma de diagrama na Figura 9.

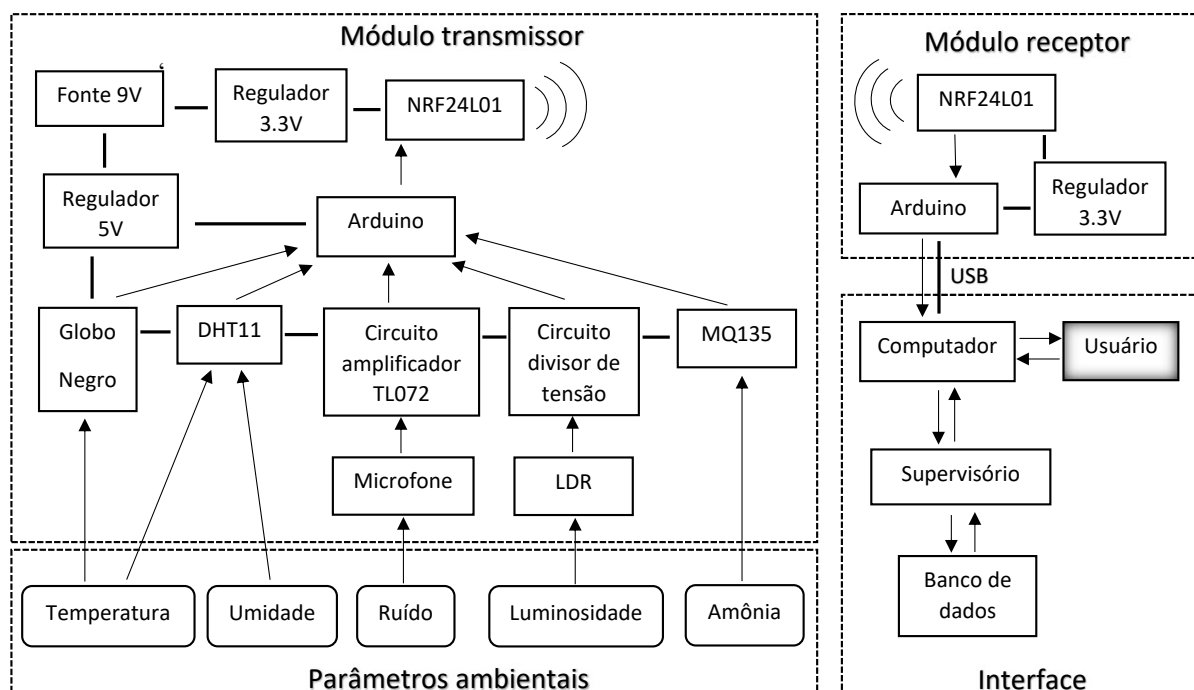


Figura 9 – Diagrama do projeto

Fonte: Autor

O sensor de globo negro foi uma expansão para o sistema de monitoramento. Ele é

considerado aqui neste capítulo, pois junto a sua implementação foram realizadas outras atualizações no *software* e no *hardware*. O sensor e o cálculo do ITGU serão amplamente abordados no Capítulo 5.

3.2 Materiais Utilizados

Durante o período de pesquisa, o projeto sofreu algumas alterações, pois a medida que as versões foram testadas, percebeu-se a necessidade de fazer pequenas mudanças para se obter a versão final. São listados a seguir os materiais utilizados para a confecção de um módulo transmissor em sua ultima versão:

- Um Arduino Nano;
- Um Módulo rádio NRF24L01+ com antena e amplificador;
- Uma Fonte de alimentação DC 9V;
- Um Sensores de temperatura e umidade relativa do ar - DHT11;
- Um Sensores de gás - MQ135;
- Uma Fotorresistência - LDR 10mm;
- Um Sensor de globo negro;
- Um Microfone de Eletreto - CZN-15E;
- Um Amplificador operacional - TL072;
- Um Regulador de tensão DC L7805CV (5V);
- Um Regulador de tensão DC Ams1117 (3.3V);
- Um Led Vermelho 3mm;
- Uma placa de cobre/fenolite de face simples (10cm x 10cm);
- Uma Caixa para projetos PATOLA PB-107 (73mm x 99mm x 41mm);
- Um Conector jack de alimentação (5,5mm , 2,1mm);
- Um Conector P10 fêmea para globo negro;
- Soquetes fêmea para pinos;
- Jumpers de alimentação;
- Resistores;

- Capacitores;
- Parafusos 3mm;

Para o módulo receptor foram utilizados os seguintes materiais:

- Um Arduino Nano;
- Um Módulo rádio NRF24L01+ com antena e amplificador;
- Um Regulador de tensão DC Ams1117 (3.3V);
- Um Led Vermelho 3mm;
- Uma placa de cobre/fenolite de face simples (10cm x 10cm);
- Uma Caixa para projetos PATOLA PB-075 (60mm x 75mm x 36mm);
- Soquetes fêmea para pinos;
- Jumpers de alimentação;
- Capacitores;
- Parafusos 3mm;

Os componentes eletrônicos como capacitores e resistores estão melhor detalhados no circuito esquemático do projeto, representado na Seção 3.5 deste capítulo.

3.3 Sensores e calibração

Os sensores de temperatura e umidade relativa do ar - DHT11, teor de amônia - MQ135 e luminosidade - LDR foram herdados do trabalho anterior. Para medir a intensidade de ruído nos biotérios foi desenvolvido um circuito com um microfone de eletreto e amplificador operacional. São apresentados a seguir as propriedades, calibração e aplicação de cada sensor.

3.3.1 Ruído - Microfone de eletreto CZN-15E

O primeiro sensor trabalhado foi o de ruído, visto que nos trabalhos anteriores a este foi usado um sensor modular que não retornava a intensidade de ruído, mas sim, apenas a presença de algum ruído alto, como uma batida de palmas ou de porta.

Para mensurar a intensidade de ruído no biotério, foi desenvolvido um circuito específico com o microfone de eletreto CZN-15E (Figura 10). O microfone capta as vibrações

sonoras do ambiente e as convertem em uma tensão alternada com frequência equivalente e amplitudes baixas, de no máximo 10mV. Para obter um sinal adequado para o Arduino, foi desenvolvido um circuito amplificador com o circuito integrado TL072, que pode ser observado na Figura 11.



Figura 10 – Microfone de eletreto CZN-15E

Fonte: (ELECTRONICS, 2017)

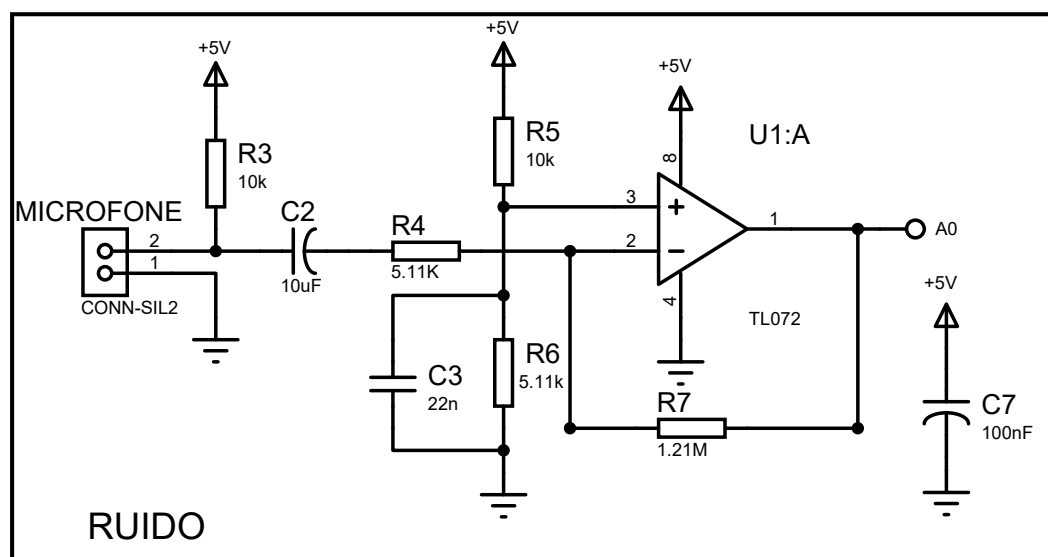


Figura 11 – Esquema eletrônico do projeto

Fonte: Autor

O circuito é alimentado com 5V, provenientes do regulador de tensão LM7805. Ele amplifica a saída do microfone em aproximadamente 238 vezes e reposiciona o 'zero' do sinal para um valor resultante do divisor de tensão formado por R5 e R6, que se aproxima de 1,69V. Na Figura 12 está representado o sinal resultante de um ruído com intensidade máxima (para a sensibilidade do sensor) e período de 1ms, ou seja, frequência de 1000Hz.

Essa metodologia foi necessária, pois o conversor analógico/digital do Arduino não consegue ler tensões negativas e possui por padrão referência de 5V. O valor de 3,4V foi obtido através de testes, onde o CI TL072 começa a saturar e não captar as oscilações. Dessa forma, o sinal resultante do ruído pode variar de 0V a 3,4V. A tensão momentânea resultante V_r pode ser obtida pela Equação 3.1, em que V_{cc} é a tensão de alimentação (5V), V_s a amplitude gerada pelo microfone em volts e R4, R5, R3, R6 são as resistências do circuito desenvolvido.

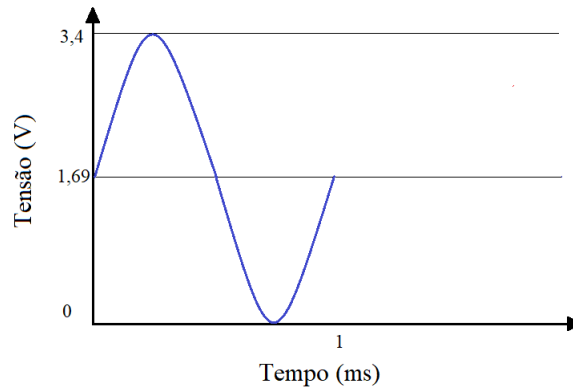


Figura 12 – Representação do sinal de ruído de máxima amplitude

Fonte: Autor

$$V_r = \frac{R6}{R5 + R6} * VCC \pm \left(1 + \frac{R7}{R4}\right) * V_s \quad (3.1)$$

Substituindo os valores na Equação 3.1 e considerando o ganho do amplificador $\left(1 + \frac{R7}{R4}\right)$ igual a 238, obtém-se a Equação 3.2.

$$V_r = 1,69 \pm 238 * V_s \quad (3.2)$$

Com o sinal tratado, para a conversão do ruído obtido em pressão sonora (dB), foi implementado uma função na IDE do Arduino. Nela é calculada a média da tensão de pico a pico (Vpp) obtida em leituras de tensão em uma frequência de amostragem alta. Seu pseudo-código está representado no Algoritmo 1.

O Arduino teoricamente consegue fazer até 10000 leituras por segundo utilizando sua função para valores analógicos, mas perto dos limites começa a apresentar erros ou até retornar valores repetidos do registrador. Buscando aproveitar o máximo de seu poder, foi possível determinar sem comprometer a integridade dos valores, um período de amostragem de $200\mu s$, que equivale a 5000 leituras por segundo.

A variável `N_AMOSTRAS_LEITURA` representa o número de amostras de leituras do ADC para considerar uma oscilação. No código final foi definido `N_AMOSTRAS_LEITURA = 10`. Para melhor compreensão, se for multiplicado 10 amostras de leitura (j) por $200\mu s$ de período entre cada uma, obtém-se o período de uma amostra de onda tratada (i) equivale a 2ms. Isso significa que em um ruído de frequência igual a 1000Hz, seriam observados duas oscilações em cada amostra de onda tratada (i).

Algorithm 1 Função de calculo da intensidade sonora

```

1: função CALCULA(dB)
2:    $noise\_avg \leftarrow 0$  ▷ Variável para acumular média
3:   para (Cada amostra  $i < N\_AMOSTRAS\_ONDAS$ ) faça
4:      $noise\_top \leftarrow 0$  ▷ Zera variável de crista do sinal
5:      $noise\_bot \leftarrow 0$  ▷ Zera variável de vale do sinal
6:     para (Cada amostra  $j < N\_AMOSTRAS\_LEITURA$ ) faça
7:        $time\_reg \leftarrow microssegundos()$  ▷ Registra o tempo em microssegundos
8:        $noise\_read \leftarrow ADC$  ▷ Variável noise_read recebe leitura do sensor
9:       se  $noise\_read > noise\_top$  então ▷ Compara com valor máximo (crista)
10:         $noise\_top \leftarrow noise\_read$  ▷ Atualiza valor da crista
11:       fim se
12:       se  $noise\_read < noise\_bot$  então ▷ Compara com valor mínimo (vale)
13:         $noise\_bot \leftarrow noise\_read$  ▷ Atualiza valor do vale
14:       fim se
15:       enquanto Falso faça
16:         Passaram  $200\mu s$  após  $time\_reg$ ?
17:       fim enquanto
18:     fim para
19:      $noise\_avg \leftarrow noise\_avg + (noise\_top - noise\_bot)$  ▷ Acumula variável
20:   fim para
21:   Divide  $noise\_avg$  pelo número de amostras  $N\_AMOSTRAS\_ONDAS$ 
22:   Converte  $noise\_avg$  em tensão média
23:   Aplica Equação encontrada para obter dB
24:   devolve dB
25: fim função

```

Observe que o V_{pp} da amostra i é acumulada em cada iteração. No final é feito uma média dessa amplitude. Foi considerado no código $N_AMOSTRAS_ONDAS = 350$. De certa forma, a intensidade do ruído medida estará relacionado ao volume da integral dos ruídos captados em um período de tempo de aproximadamente 700ms.

A calibração do sensor foi feita tomando-se como referência um decibelímetro Instrutherm DEC-460 e utilizando a ferramenta linha de tendência logarítmica do *software Microsoft Excel*. O sensor foi posicionado ao lado do decibelímetro, ambos foram orientados para uma fonte de ruído branco de 1kHz. Foram realizadas amostras com diferentes intensidades sonoras. O resultado da calibração pode ser observado por meio da Figura 13.

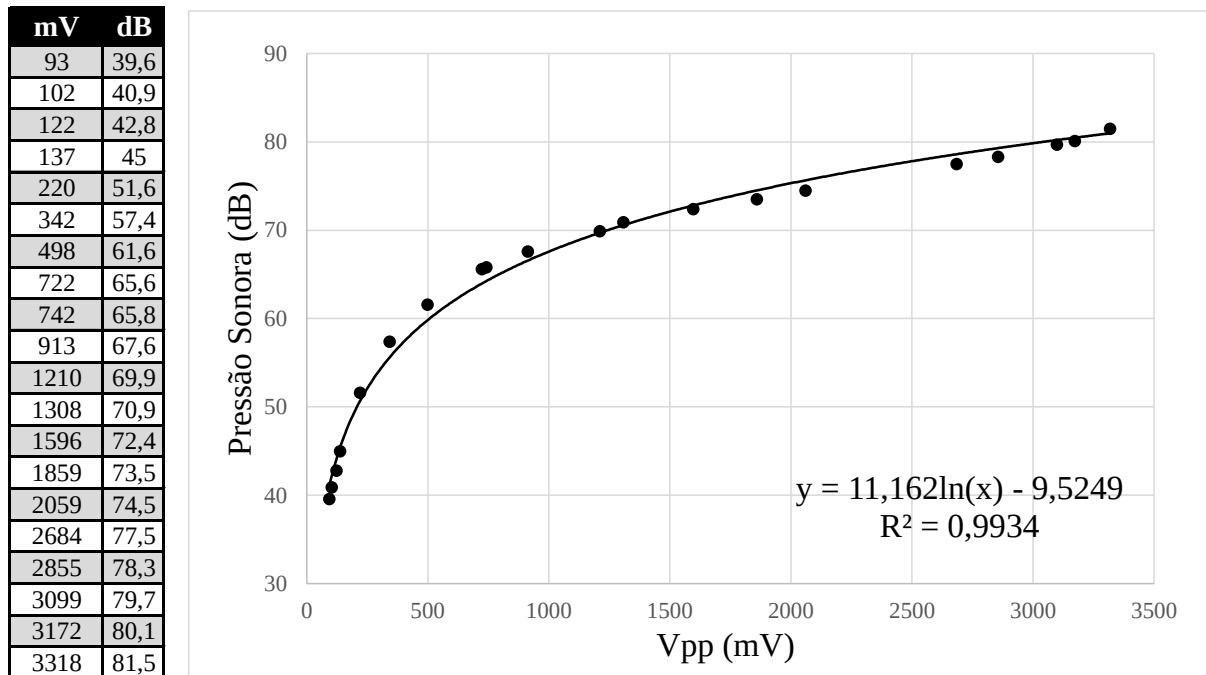


Figura 13 – Dados e curva de calibração do sensor de intensidade sonora

Fonte: Autor

Foram feitas três observações para cada intensidade de ruído, levando em conta vinte intensidades diferentes. Os valores médios foram registrados na tabela da Figura 13 e então gerado o gráfico, em que: Vpp(mV) é o valor de pico a pico médio em milivolts, obtido pela função descrita no Algoritmo 1, notoriamente sem a aplicação da equação de calibração. Pressão Sonora (dB) é o valor retornado pelo decibelímetro em decibéis.

3.3.2 Iluminância - LDR

Lux é a unidade de intensidade de iluminação ou iluminância, que corresponde à incidência perpendicular de um lúmen por metro quadrado. O fluxo luminoso é compreendido como a radiação emitida por uma fonte de luz visível, ou seja, em comprimentos de onda entre 380 a 780 nm (LUZ, 2010). O LDR (*Light Dependent Resistor*) (Figura 14) é um componente que apresenta resistência variável de acordo com a intensidade de luz incidida sobre ele. Os fótons da luz que incidem sobre o semicondutor liberam elétrons para a banda condutora que irão melhorar a sua condutividade, diminuindo a resistência.



Figura 14 – Sensor LDR

Fonte: Autor

A utilização do LDR como sensor de iluminância é feita através da medida de sua resistência, pois ela é única para cada valor de iluminância recebida em uma determinada faixa de comprimentos de onda (TECHNOLOGIES, 2008). Como não é possível medir a resistência diretamente pelas portas I/O (entrada e saída) do Arduino, o LDR é utilizado como uma das resistências de um divisor de tensão. Na Figura 15 ilustra esse circuito, em que o valor da resistência do LDR pode ser obtido pela Equação 3.3.

$$R_{LDR} = R1 \times \frac{V_a}{V_{cc} - V_a} \quad (3.3)$$

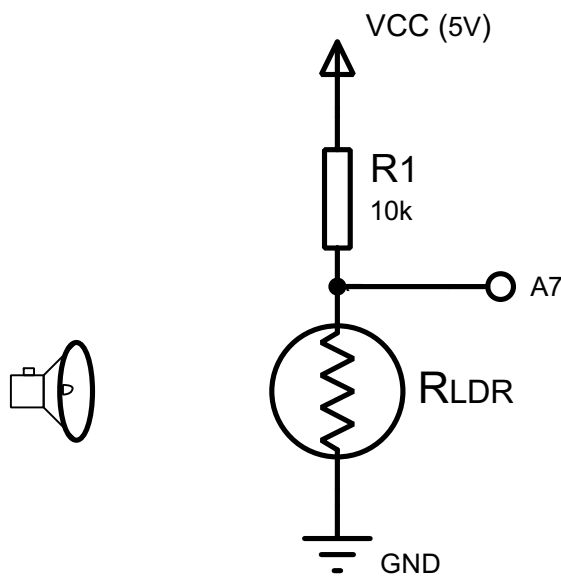


Figura 15 – Circuito do LDR

Fonte: Autor

Em que: V_{cc} é a tensão de alimentação do circuito e a referência do conversor analógico/digital (ADC), que corresponde a 5 Volts. V_a é tensão obtida com a leitura da porta analógica A7 do Arduino e $R1$ uma resistência de $10k\Omega$. Obtido o valor de R_{LDR} , já é possível determinar a iluminância captada através de uma equação de calibração, que deve ter comportamento exponencial segundo a Figura 16 presente na folha de dados do sensor.

A calibração foi feita em laboratório, com uma luz de temperatura semelhante à luz do biotério. Um luxímetro foi posto ao lado do LDR com o mesmo ângulo sobre a fonte de luz. Então foram comparados os valores obtidos de lux do luxímetro com a resistência do LDR calculada no Arduino. Foram considerados 14 diferentes intensidade de luz para a calibração. Na Figura 17 observa-se os dados obtidos e curva de calibração gerada pela regressão exponencial realizada no *software Microsoft Excel*.

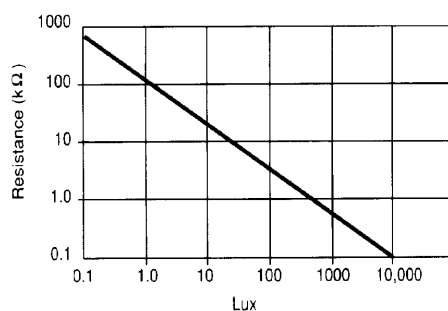


Figura 16 – Curva de comportamento do LDR

Fonte: (TECHNOLOGIES, 2008)

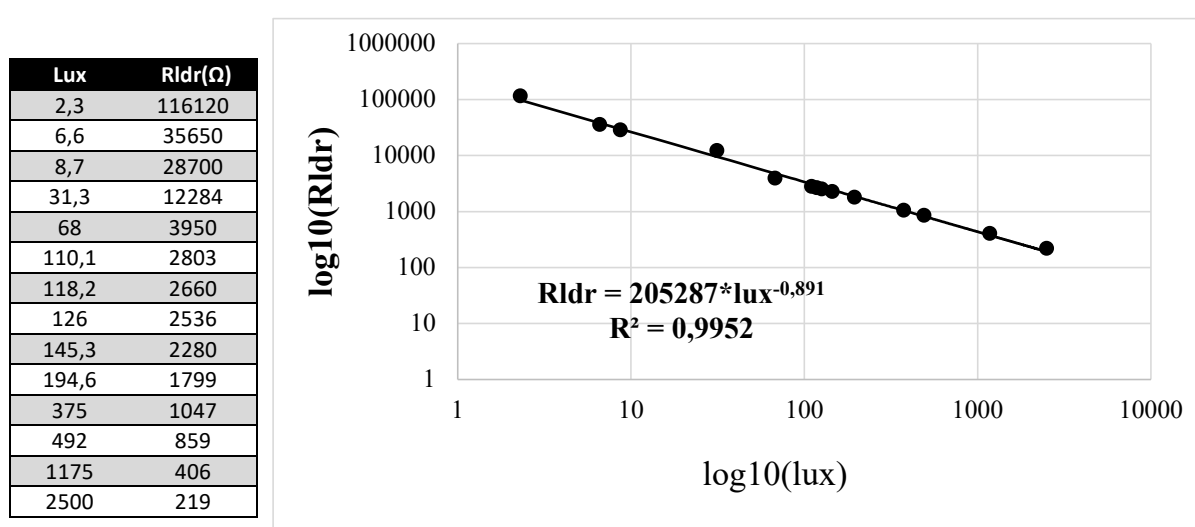


Figura 17 – Dados e curva de calibração do LDR

Fonte: Autor

Manipulando a Equação 3.4 obtida da curva de calibração, tem-se a Equação 3.5, sendo essa última possível determinar a iluminância do biotério em lux a partir da resistência do LDR - R_{LDR} calculada. Todos os cálculos são feitos diretamente no programa do Arduino, onde a entrada é a tensão lida pela porta analógica A7 e a saída é a iluminância em lux calculada como apresentado.

$$R_{ldr} = 205287 \times lux^{-0,891} \quad (3.4)$$

$$Lux = \left(\frac{R_{ldr}}{205287} \right)^{-1,12233} \quad (3.5)$$

3.3.3 Temperatura e umidade - DHT11

O DHT11 (*Digital Humidity and Temperature Sensor*) é um sensor digital de temperatura e umidade relativa do ar. Sua faixa de medição abrange 0 a 50 graus Celsius (°C) para a temperatura e 20% a 90% para a umidade relativa do ar (UR%), com precisão de $\pm 2,0^{\circ}\text{C}$ e $\pm 5,0\%$, respectivamente. Quando o sensor recebe um sinal de requisição do microcontrolador, ele transmite de volta seus dados de leitura em forma de pulsos. Esse procedimento pode levar até 270ms e, ainda, o sensor necessita de um intervalo mínimo de 2 segundos para aferir uma nova leitura. Seus dados técnicos foram consultados da folha de dados disponibilizada pelo fabricante na internet (TECHNOLOGIES, 2012).

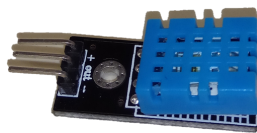


Figura 18 – Sensor de temperatura e umidade relativa DHT11

Fonte: Autor

No projeto, foi utilizado o módulo do DHT11 (Figura 18) que, além do sensor (parte azul), é equipado de um circuito de filtro e um conector padrão de três pinos que facilita a sua utilização. O pino "+" do sensor é conectado a alimentação de 5V, o pino "-" ao GND e "out" em algum pino digital do Arduino. Não foi necessário efetuar a calibração do sensor, pois o mesmo já foi testado pelo fabricante, que garante seu funcionamento e precisão em sua folha de dados. É possível encontrar sua biblioteca de programação no gerenciador de bibliotecas da IDE do Arduino. As variáveis de umidade e temperatura foram definidos e obtidos sua leitura utilizando as seguintes funções da bibliotecas:

- `#include <DHT.h> //biblioteca do sensor DHT11`
- `#define pin_dht A4 //define pino utilizado no Arduino`
- `DHT dht(pin_dht, DHT11); //cria um objeto DHT11 da biblioteca`
- `float humidity = dht.readHumidity(); //retorna umidade`
- `float temperature = dht.readTemperature(); //retorna temperatura`

3.3.4 Índice de amônia - MQ135

A sensor MQ135 (Figura 19) apresenta resistência interna variável de acordo com a concentração de amônia no ar e outros gases. Além da amônia, o MQ135 também detecta gases que fazem parte de reações, catalisadores ou produtos finais como o CO, CO₂, NH₄, álcool e fumaça. Algoritmos podem ser usados para otimizar a sensibilidade do sensor à concentração desses gases no ambiente (GHOSH et al., 2012).

A partir da curva exponencial de resposta do sensor, presente em sua folha de dados, é possível fazer uma interpolação de pontos para se obter a equação equivalente. Com essa técnica, Jesus (2017) determina que a relação resistência variável do sensor e a concentração de amônia no ar é estabelecida por meio do modelo da Equação 3.6.

$$ppm(NH_3) = 111,2 * \left(\frac{R_s}{R_0}\right)^{-2,801} \quad (3.6)$$

Onde ppm(NH₃) é a concentração de amônia no ar em partícula por milhão, R_s é a resistência variável calculada e R₀ é uma constante do sensor. R₀ corresponde ao valor de resistência R_s em condição de 100ppm de amônia no ar. Porém, não foi possível encontrar o valor de R₀ para a recalibração do sensor, devido a falta de um sensor comercial de referência. Então foi utilizado o mesmo valor empregado por Jesus (2017), como na Equação 3.7.

$$ppm(NH_3) = 111,2 * \left(\frac{R_s}{34503}\right)^{-2,801} \quad (3.7)$$

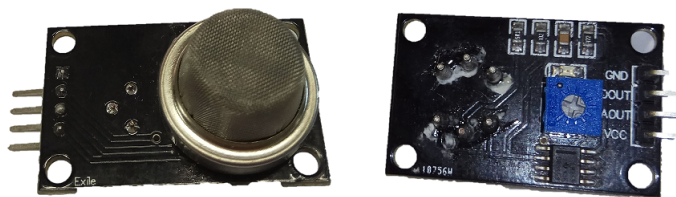


Figura 19 – Sensor de amônia MQ135

Fonte: Autor

R_s compõem um circuito divisor de tensão da saída do sensor. Sendo que R_l é fixa, em série com a saída e R_s é a resistência próxima a alimentação, calculada pela equação 3.8.

$$R_s = R_l \times \frac{V_{cc} - V_s}{V_s} \quad (3.8)$$

Em que: R_l equivale a $1k\Omega$, $V_{cc} = 5V$ e V_s é a tensão de saída lida pela porta analógica do Arduino. Como explicado no método de calibração do LDR, todo cálculo é feito pelo Arduino.

Note que o sensor possui quatro pinos, mas são usados apenas três (V_{cc} , GND, Sinal analógico). O quarto pino faz parte do módulo em especial, que acompanha de um amplificador operacional comparador. Esse circuito apenas compara o valor analógico de saída do sensor com a saída ajustável do potenciômetro para gerar um sinal digital, como um alarme por exemplo. O pino digital não foi utilizado, pois todo tratamento é feito no *software*.

3.4 Módulo rádio - NRF24L01

O NRF24L01 da *Nordic Semiconductor* é o chip principal do módulo de comunicação sem fio. É um comunicador sem fio que utiliza frequências de 2,4GHz e opera a 3,3V nominais. Utiliza protocolo de comunicação SPI com qualquer placa microcontrolada. Um único módulo pode ser programado para transmitir ou receber dados de um segundo módulo. No código do microcontrolador, é possível atribuir ao NRF24L01 a função de escrita ou leitura de dados a qualquer momento.

Inicialmente foi utilizado a versão do módulo NRF24L01 simples, com antena embutida. Porém, durante os testes não demonstrou um alcance satisfatório. Por esse motivo, uma segunda versão foi adquirida e então instalada ao projeto, o módulo NRF24L01+ com amplificador e antena externa (Figura 20).



Figura 20 – Módulo NRF24L01+ com amplificador e antena externa

Fonte: Autor

O NRF24L01 também possui o conceito de plataforma aberta, GitHub se encontram várias bibliotecas para o módulo transmissor sem fio. GitHub é uma plataforma de hospedagem de código-fonte com controle de versão. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos *Open Source* de qualquer lugar do mundo. A biblioteca nRF24/RF24 (TMRH20, 2017) para Arduino disponível no repositório do GitHub foi utilizada no projeto. Para melhor entendimento de seu funcionamento, abaixo está um exemplo de programação para envio e recebimento de dados e os respectivos comentários:

- `#include <SPI.h>`
- `#include "nRF24L01.h"` //bibliotecas para o módulo NRF24L01.
- `#include "RF24.h"`
- `RF24 radio(7,8);` //declara objeto da biblioteca utilizando os pinos 7(CSN) e 8(CE) da comunicação SPI do Arduino.
- `const uint64_t pipe = 0xF1F2F3F4F5LL;` //endereço de comunicação.
- `radio.openWritingPipe(pipe);` //abre porta de escrita.
- `radio.write(&data, sizeof(MyData));` //envia estrutura "data"do tipo Mydata.
- `radio.openReadingPipe(1,pipe);` //abre porta de leitura.
- `radio.startListening();` //começa a ouvir.
- `if (radio.available()) {radio.read(&data, sizeof(MyData));}` //verifica se algo chegou e então lê os dados e armazena em "data".

3.5 Esquemático do circuito

Os circuitos foram elaborados no *software* Proteus 8.6. Ele permite além de fazer o desenho esquemático do circuito de forma padronizada, simulá-lo e posteriormente confeccionar a layout da placa de circuito impresso. Também possui o recurso de visualização em 3D da placa, auxiliando o usuário na visualização das dimensões de seu projeto. Esses recursos estão conectados entre si, de forma que, a confecção da layout da placa é assistida pelo esquemático elaborado.

Na Figura 21 tem-se o esquemático da versão final do módulo transmissor. O esquemático do módulo receptor está na Figura 22. Não foi necessário a simulação, pois como é um projeto de continuação, os materiais já estavam disponíveis e testados. Por se tratar de comunicação sem fio, uma simulação não seria eficaz ao representar a realidade.

Ao observar o projeto nas Figuras 21 e 22 com atenção, parece incondizente ter usado reguladores de tensão LM7805 para o Arduino, pois o mesmo já possui embarcado em sua placa, não é mesmo? Pois bem, isso foi necessário, pois na primeira versão do projeto, que foi utilizado o regulador interno do Arduino, percebeu-se que ele não retornava uma tensão constante e estável, o que comprometia as leituras dos sensores. Isso ocorreu devido a alta demanda de corrente do sensor MQ-135, que possui em sistema de aquecimento para funcionar. Além da instabilidade, esse problema gerava aquecimento indevido ao regulador interno.

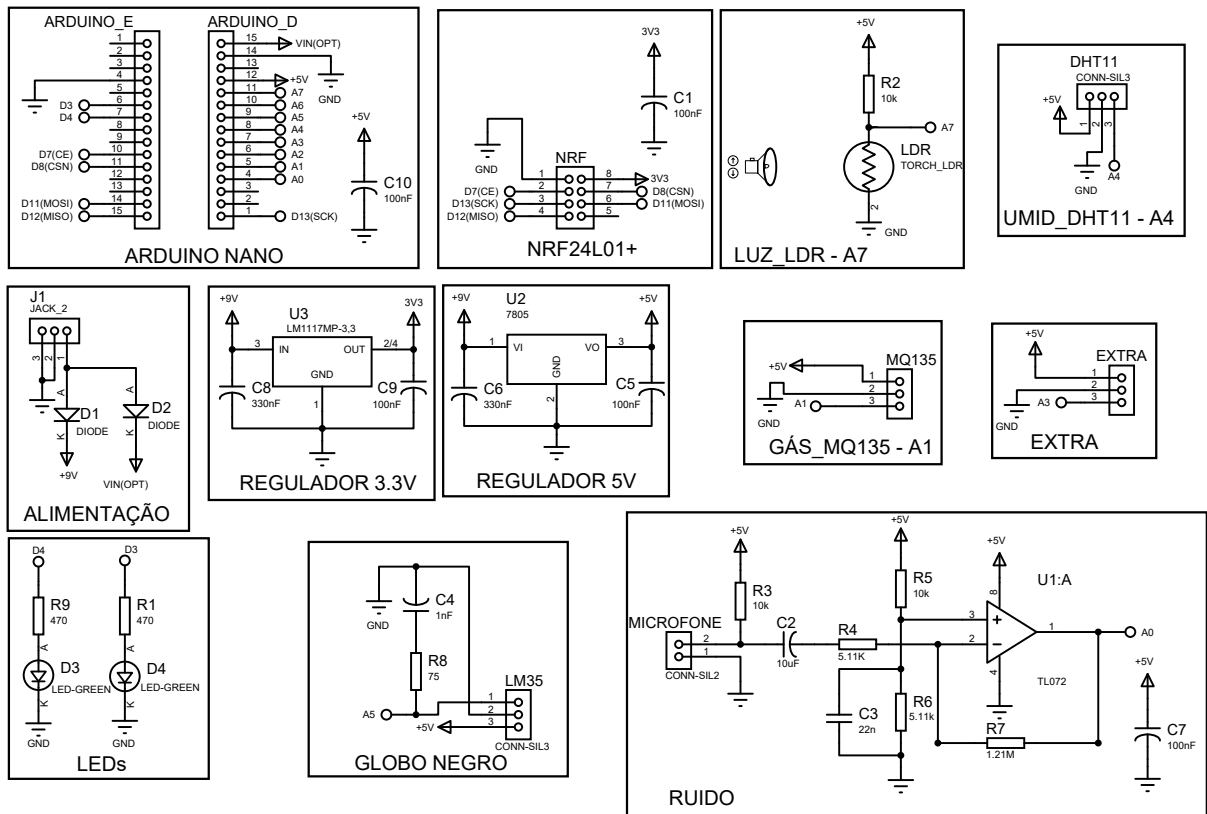


Figura 21 – Esquema eletrônico do módulo emissor

Fonte: Autor

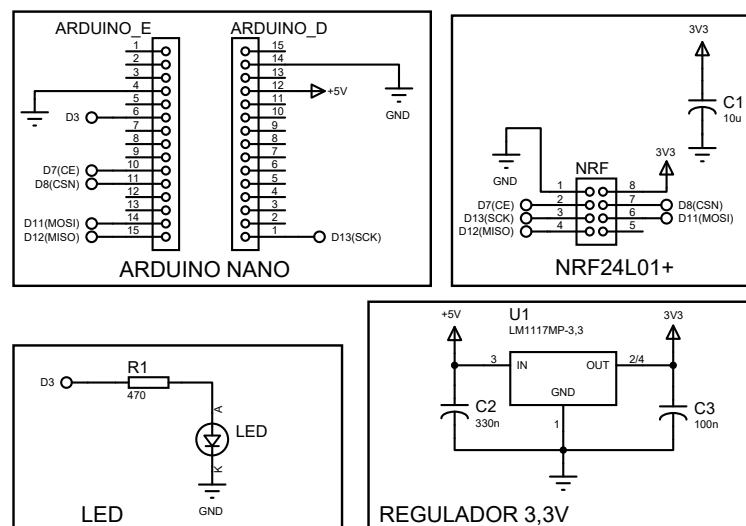


Figura 22 – Esquema eletrônico do módulo receptor

Fonte: Autor

Sobre a saída 3,3V do Arduino, ela é apenas auxiliar do chip FTDI de comunicação e não entrega corrente suficiente para o NRF24L01+, sendo necessário então um segundo

regulador, o LM1117MP-3,3V.

O módulo emissor é alimentado com uma fonte de 9V, seus reguladores garantem 5V ao Arduino e sensores e 3,3V ao módulo NRF24L01+. O módulo receptor é alimentado via USB. Como não há muitos gastos de energia para receber dados, o regulador de tensão LM1117MP-3,3V consegue potência suficiente para manter o módulo rádio funcionando bem.

Todas as conexões estão representadas nas Figuras 21 e 22 no padrão de conexão do Arduino. As portas que começam com D são apenas digitais e o número representa a posição da porta. As que começam com A são as que estão conectadas ao conversor analógico/digital, podendo fazer leituras analógicas. Mas no caso do DHT11, a porta A4 em que está conectado funciona normalmente como uma entrada digital.

Foram utilizados conectores de espaçamento de 0,1 polegadas (2,5mm) para representar as conexões. Isso para padronizar a conexão dos sensores, pois eles têm arranjos diferentes de alimentação, terra e sinal. Isso foi possível devido a proposta do layout da placa, que seria de cabear o sensor até a conexão na PCI, podendo alterar a posição final do conector.

Os LEDs auxiliam a visualização do funcionamento dos módulos. O arduino e alguns sensores já possuem LEDs, mas como o projeto seria encapsulado, foi necessário a utilização de LEDs externos.

3.6 Layout da placa

O layout final foi projetado pensando nas caixas de acoplamento de projetos PATOLA PB107 para o transmissor e PATOLA PB-075 para o receptor. A Figura 23 representa o layout da placa de circuito impresso do módulo transmissor, com 90mm de comprimento e 64mm de largura.

A Figura 24 mostra o layout da placa de circuito impresso do módulo receptor, com 60mm de comprimento e 54mm de largura.

Ambos projetos são para placas de cobre de face simples e serem confeccionadas em uma máquina CNC (Comandos Numéricos Computadorizado). O percurso da CNC foi criado no ArtCAM 2008, um *software* que permite a vetorização de imagens, como os das layouts. Feito a vetorização, o percurso de fresar a placa virgem de cobre para formar as trilhas e ilhas e o percurso de furação das placas é gerado. Então é só posicionar a placa e colocar a CNC para executar os comandos automaticamente.

Apenas as partes em preto nas Figuras 23 e 24 são onde a CNC executa a fresa ou furação. As demais áreas são preservadas, formando o circuito. Alguns pinos não utilizados do Arduino foram retirados, liberando uma área maior de aterramento.

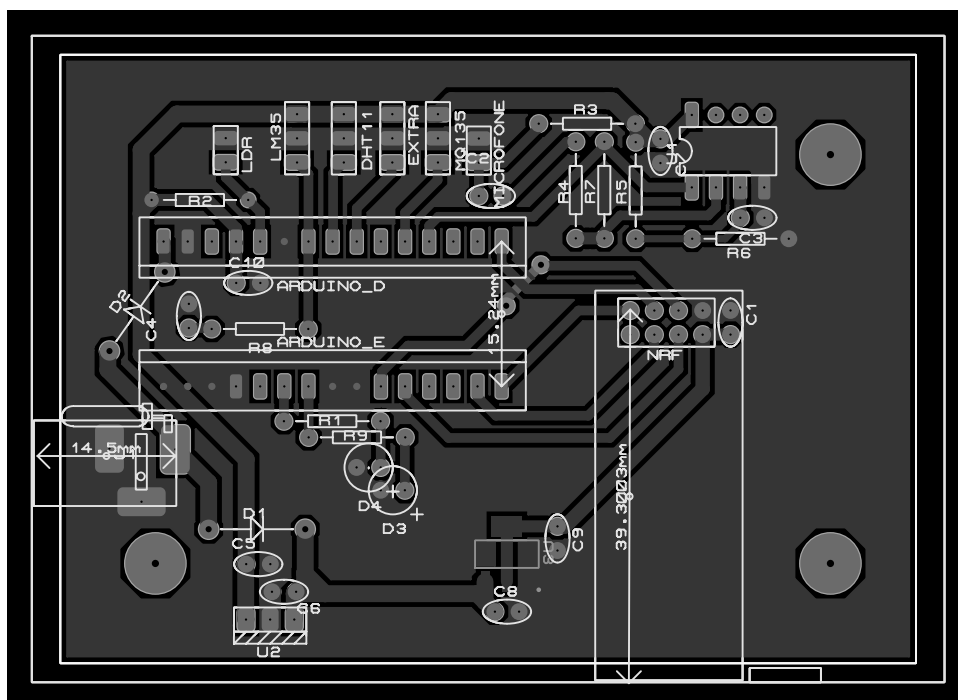


Figura 23 – Layout da placa de transmissão

Fonte: Autor

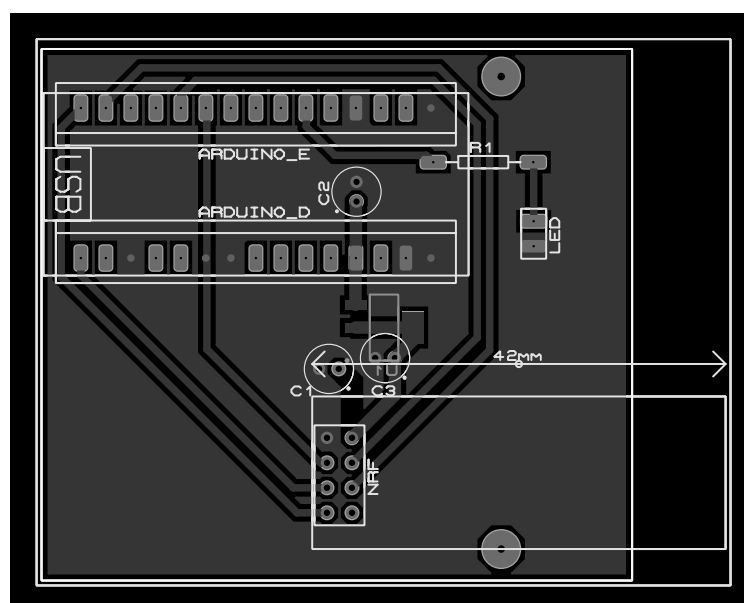


Figura 24 – Layout da placa de recepção

Fonte: Autor

Observe a padronização de conexão dos sensores. Desta forma, fica mais fácil a elaboração do layout da placa, evitando *jumpers*, que são pequenas ligações por fio sobre a placa.

3.7 Prototipagem dos módulos

Os módulos transmissor e receptor foram desenvolvidos da forma bastante compacta. Como demonstrado na Seção 3.6, os layouts das placas foram trabalhados para comportar nas caixas de projeto selecionadas.

No módulo dos sensores, a placa de circuito impresso com todos os componentes e conectores é parafusada na tampa da caixa Patola PB-107. Os sensores ficam acoplados no fundo da caixa, em seus respectivos orifícios fresados na CNC. Lembrando que os sensores são cabeados até a placa, então a caixa é fechada e parafusada (Figura 25). Foi levado em consideração o tamanho de cada componente, dissipação de calor, possíveis interferências, posição ideal para cada sensor e espaço disponível.

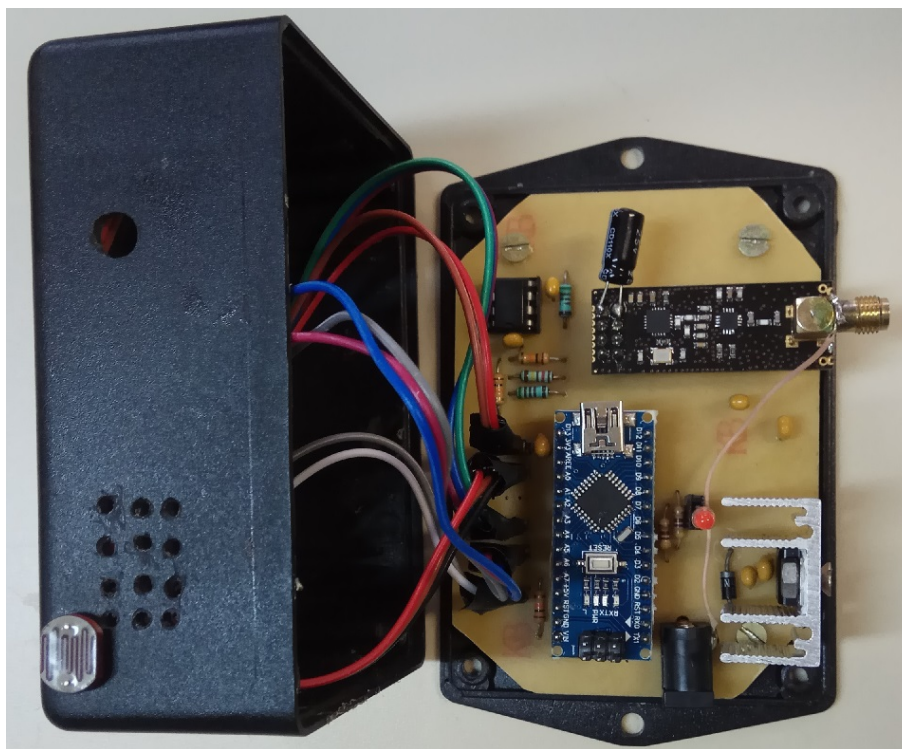


Figura 25 – Módulo emissor aberto

Fonte: Autor

Na Figura 26 se encontra o módulo emissor finalizado. Os sensores de amônia, ruído, temperatura, umidade relativa do ar e temperatura de globo negro ficam a frente do módulo. O LDR capta a iluminação de cima. Na parte de cima também se encontra o conector de antena do NRF24L01+ e furos para dissipação de calor. Na parte lateral está o conector do tipo jack para a fonte de alimentação. O globo negro é acoplado ao módulo através de um conector P10 fêmea adaptado para ele.

Para o módulo receptor foi preciso fazer fresas na caixa para a porta USB, saída da antena do NRF24L01+ e um pequeno orifício para o LED externo. Na Figura 27 se



Figura 26 – Módulo emissor

Fonte: Autor

encontra o módulo receptor em funcionamento.



Figura 27 – Módulo receptor

Fonte: Autor

3.8 Software

O *software* do presente projeto é composto pelas programações realizadas na IDE Arduino para os módulos transmissor e receptor, bem como da programação do sistema

supervisório feita no Visual Studio Community 2017 com linguagem C# para Windows 7 ou superior.

Cada módulo transmissor faz a leitura dos sensores em aproximadamente 2 segundos e entra em estado de envio de dados sem fio. O pacote de dados a ser enviado é uma estrutura de 10 variáveis do tipo inteiras de 16 bits. Quando o pacote de dados é enviado com sucesso, o LED indicador executa um sinal longo. Caso não consiga enviar o pacote, o LED indica com três sinais curtos e o Arduino aguarda 200ms para tentar novamente, caso isso se repita até um intervalo máximo de aproximadamente 15 segundos, o Arduino aguarda mais um pouco e entra em um novo ciclo. Por programação é garantido que cada ciclo de leitura de dados, transmissão e espera dure 20 segundos.

No Algoritmo 2 está representado o protocolo de envio de dados do módulo emissor.

Algorithm 2 Protocolo de envio de dados sem fio

```

1: enquanto Arduino ligado faça
2:    $tempo\_inicio \leftarrow$  tempo atual em milissegundos           ▷ Grava tempo de inicio
3:   Struct  $dados \leftarrow$  leituras dos sensores                 ▷ Até 2 segundos
4:   boolean  $enviado \leftarrow false$                              ▷ Variável que monitora envio
5:   enquanto ( $enviado = false$ ) faça
6:      $enviado \leftarrow função\_envia()$                          ▷ Retorna verdadeiro caso sucesso
7:     se enviado então
8:       piscada longa do LED
9:       aguarda 20 segundos após  $tempo\_inicio$ 
10:      sai do enquanto                                          ▷ dados enviados e retorna o ciclo
11:     senão
12:       pisca o LED rapidamente 3 vezes
13:       se (passaram 20 segundos após  $tempo\_inicio$ ?) então
14:         Sai do enquanto
15:       senão
16:         Aguarda 200ms
17:       fim se
18:     fim se
19:   fim enquanto
20: fim enquanto

```

O módulo receptor fica conectado ao computador em que está instalado o sistema supervisório desenvolvido. Ele está apto a receber os dados dos módulos transmissores a todo momento. Quando um pacote de dados é recebido, ele é tratado em uma string de transmissão para o supervisório. A conexão Arduino-computador é realizada através do chip FTDI embarcado no Arduino. O FTDI simula uma porta serial ao computador através da porta USB, seu drive está incluso no IDE do Arduino.

3.8.1 Supervisor

O sistema supervisor desenvolvido na plataforma Visual Studio Community 2017 foi testado e modificado diversas vezes a fim de otimizar as suas funções e atender as necessidade do usuário. O programa foi reestruturado em camadas de projetos e implementado o banco de dados local SQL Server. Foi adicionado também a geração de tabelas e gráficos com filtros pré-selecionados.

A programação em camadas consiste em criar projetos com classes e bibliotecas separadamente, mas que comunicam entre si. Essa técnica facilita a organização da programação, deixando o código mais estruturado. Isso permite que o código seja mais facilmente compreendido por múltiplos programadores. Além de tudo, facilita a implementação de novas funções ao projeto. Na Figura 28 mostra a janela de gerenciamento da solução, onde estão as quatro camadas criadas para o projeto.

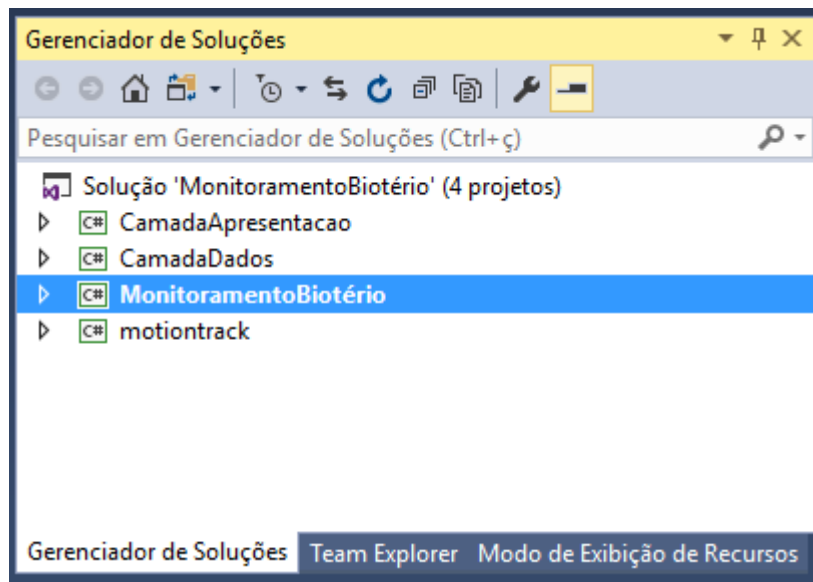


Figura 28 – Tela de programação do projeto

Fonte: Autor

A camada em destaque "MonitoramentoBiotério", inicializa o *software* e faz a chamada da camada de apresentação.

A camada de apresentação compõe as duas janelas do programa de monitoramento, janela principal de visualização em tempo real dos dados e a janela de consulta do histórico de dados.

A camada de dados armazena os dados coletados pelo sistema no banco de dados. Apenas tem acesso a conexão com o banco de dados, evitando conflitos.

A última camada "motiontrack", corresponde ao sistema de rastreamento com visão computacional. Foi implementada após a conclusão do sistema de monitoramento e seu

desenvolvimento será abordado posteriormente no Capítulo 6.

Na janela principal do programa desenvolvido (Figura 29) é feito o monitoramento em tempo real dos parâmetros. No seu canto superior direito encontra-se a caixa de configuração, onde é feita a conexão serial/USB com o Arduino receptor. Para efetua-la, basta selecionar a porta serial emulada que corresponde ao Arduino e então clicar em "Start". Todos os tratamentos de possíveis erros são feitos, se tudo ocorrer bem, a mediada que o supervisor recebe dados, o gráfico e os gauges são atualizados.

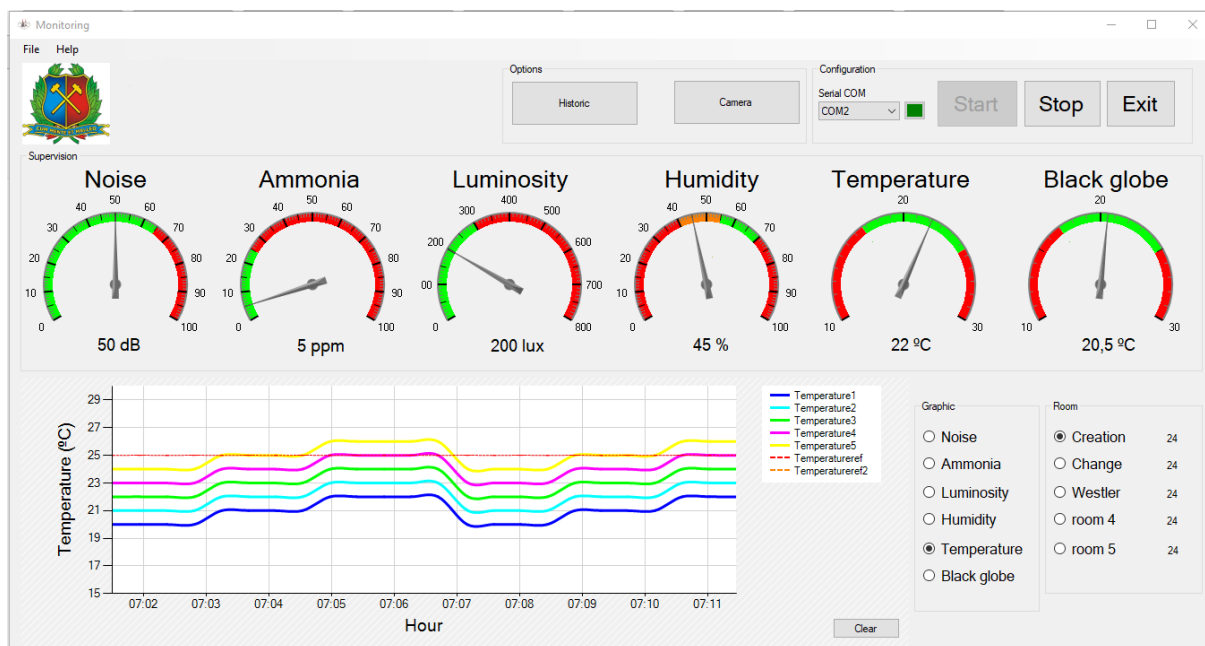


Figura 29 – Tela principal do programa

Fonte: Autor

A Figura 29 foi obtida com dados de uma simulação de valores crescentes e decrescentes de temperatura, apenas para demonstrar seu funcionamento.

O monitoramento em tempo real é composto por seis indicadores de parâmetros do tipo gauges, um gráfico dinâmico e filtros. As gauges monitoram todos os parâmetros de uma sala por vez. No filtro "Room", é selecionado a sala em que as gauges tomarão como referência. Já o gráfico mostra os valores mais recentes de um único parâmetros em todas as salas ao mesmo tempo. No filtro "Graphic" é possível alterar o parâmetro a ser monitorado no gráfico. O gráfico é limpo a cada dia, ou ainda pode ser limpo manualmente pelo botão "Clear".

Os valores de leitura são recebidos via serial e são salvos no banco de dados através da camada de dados. Na caixa "Room", ao lado de cada nome da sala se encontra um número que indica o recebimento de pacotes de dados. Espera-se que a cada 20 segundos seja atualizado, caso contrário, indica que o sistema não está recebendo dados de tal sala.

Sobre a conexão com o banco de dados, basta que o computador tenha instalado o pacote do banco de dados local SQL Server, que foi incluso no pacote de instalação do programa. Ao iniciar o programa, é chamada uma função que testa a conexão com o banco de dados. Caso dê certo, a mensagem da Figura 30(a) será exibida. Caso não consiga conectar por falta de algum arquivo ou outro problema, a mensagem da Figura 30(b) será exibida.

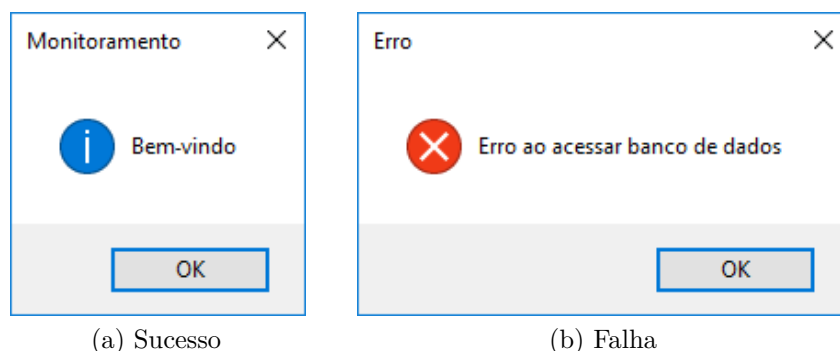


Figura 30 – Mensagens de conexão com o banco de dados

Fonte: Autor

Na parte superior da tela principal encontra-se o botão "Historic", que abre a janela de histórico (Figura 31) para consulta do banco de dados. Na tabela são exibidos: identificação do dado, sala de onde veio, data, hora, temperatura, umidade, iluminância, ruído, nível de amônia, temperatura de globo negro, tempo de transmissão de dados sem fio em milissegundos e o número de erros de transmissão acumulados caso tenham ocorrido. A imagem foi obtida com dados de um teste realizado no biotério.

Para realizar uma consulta, primeiramente o usuário deve selecionar a sala em "Select room", podendo escolher a opção "All", que não filtra por sala. Em seguida, selecionar o período de visualização em "Select period", e o intervalo de tempo em dias. O filtro por valores é opcional. O botão "Update" faz a busca no banco de dados. São gerados uma tabela de dados e um gráfico que podem ser visualizados alternando entre as abas Table e Graphic. O botão "Excel" na aba de exportação salva a tabela gerada no formato xls. No canto direito, no filtro "View", é possível selecionar os parâmetros desejados para visualização.

O período de detalhamento "Detail" ou detalhado mostra todos os dados obtido. Foi utilizado no exemplo da Figura 31. é muito útil para buscar alguma anormalidade utilizando junto ao filtro de valores.

O período de detalhamento "Hour" ou por hora, calcula a média dos dados pré-selecionados de cada hora. Na Figura 32 tem-se um exemplo do gráfico gerado dessa seleção dos dados do dia 26/07/2017 ao dia 28/07/2017 da sala 1 para umidade. A opção de período "Day" tem a mesma lógica da anterior, porém com intervalo de dias e não horas.

Historic

Select room: ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ All

Select period: ☒ Detailed ☐ Hour ☐ Day

Filter by Date: From: 26/07/2017 To: 01/08/2017

Filter by Value: Filter: Value: From: To: Update

Export: Excel

Id	Room	Date	Time	Lux	Temperature1	Humidity	Noise	Ammonia	Temperature2	Readtime	Erros
9747	1	26/07/2017	06:53:18	4	17	63	64	4	15,1	87	0
9748	1	26/07/2017	06:53:38	4	17	63	64	4	15,1	87	0
9750	1	26/07/2017	06:53:58	5	17	63	64	4	15,1	87	0
9751	1	26/07/2017	06:54:18	5	17	63	63	4	15,1	87	0
9752	1	26/07/2017	06:54:38	61	17	63	63	4	15,1	87	0
9753	1	26/07/2017	06:54:58	65	17	63	64	4	15,1	87	0
9755	1	26/07/2017	06:55:18	71	17	63	63	4	15,1	87	0
9757	1	26/07/2017	06:55:38	78	17	63	64	4	15,1	87	0
9758	1	26/07/2017	06:55:57	82	17	63	63	4	15,1	87	0
9759	1	26/07/2017	06:56:17	87	17	63	64	4	15,1	87	0
9761	1	26/07/2017	06:56:37	88	17	64	64	4	15,1	87	0
9762	1	26/07/2017	06:56:57	91	17	63	64	4	15,1	87	0
9766	1	26/07/2017	06:57:17	95	17	64	64	4	15,1	87	0
9768	1	26/07/2017	06:57:37	97	17	63	63	4	15,1	87	0
9770	1	26/07/2017	06:57:57	98	17	64	63	4	15,1	87	0
9772	1	26/07/2017	06:58:17	99	17	64	63	4	15,1	87	0
9773	1	26/07/2017	06:58:37	101	17	64	63	4	15,1	87	0
9776	1	26/07/2017	06:59:57	101	18	64	63	4	15,1	87	0
9778	1	26/07/2017	06:59:17	101	17	64	63	4	15,1	87	0
9779	1	26/07/2017	06:59:37	103	17	64	63	4	15,1	87	0
9780	1	26/07/2017	06:59:57	103	17	64	63	4	15,1	87	0
9782	1	26/07/2017	07:00:17	103	18	64	64	4	15,1	87	0
9783	1	26/07/2017	07:00:37	106	17	64	63	4	15,1	87	0

View: ☒ Noise ☒ Ammonia ☒ Luminosity ☒ Humidity ☒ Temperature1 ☒ Temperature2 ☒ Ammonia2 ☒ Errors ☒ Select all

Figura 31 – Janela de histórico

Fonte: Autor

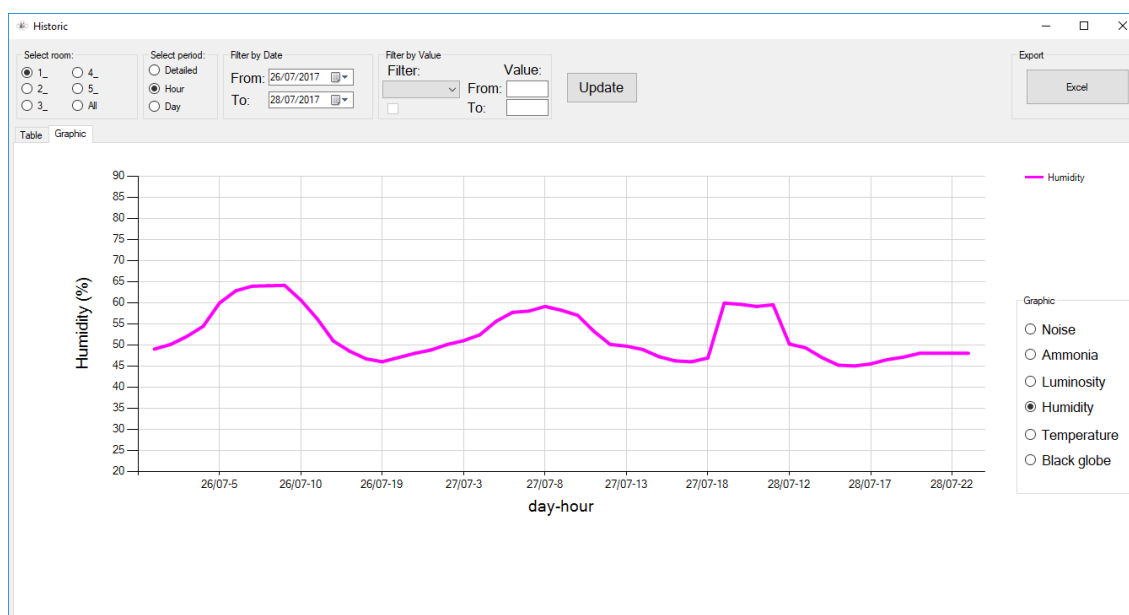


Figura 32 – Exemplo de busca com média horária

Fonte: Autor

Os eixos dos gráficos são dinâmicos. A medida que os dados de visualização são alterados, a escala é ajustada. Note que na Figura 32 por exemplo, o eixo X é representado por dia-hora, pois não faz sentido representar minutos e segundos em uma média horária.

4 Resultados da Primeira Etapa

Este trabalho contribuiu bastante para o aperfeiçoamento do sistema de monitoramento de biotérios. Com os módulos prototipados em caixas especiais para projetos, já é possível utilizá-los por um grande período de tempo nos biotérios, o que não era possível antes devido às suas normas de operação. A versão final obtida dos módulos ficaram de tamanho satisfatório, não comprometendo o espaço do biotério nem do escritório. Na Figura 33 estão ilustrados dois módulos transmissores em funcionamento.



(a) Módulo 1



(b) Módulo 2

Figura 33 – Módulos transmissores em funcionamento

Fonte: Autor

A aplicação do regulador de tensão LM7805 ao projeto corrigiu a instabilidade de tensão presente nas versões anteriores. Além disso, por poder ser posicionado na parte mais externa da placa e acoplado a um dissipador, otimizou-se a dissipação de calor geral do módulo.

Dependendo da posição da janela na sala monitorada, os valores de iluminância variam. Além do que pode ocorrer incidência solar próxima ao sensor LDR, mesmo que não haja sobre o rack de gaiolas. Isso representa mal a medição de iluminação no ambiente, sendo necessário analisar a iluminação de cada sala durante o período de um dia inteiro para escolher a melhor posição do sensor. Considerando também que o ideal para um biotério seria a iluminação artificial e controlada, fornecendo um período de 12h de luz por dia. A altura de um metro do chão foi determinada pela consulta na literatura, onde

essa posição é tomada como referência para mensurar a iluminância do local (LUZ, 2010).

Observe que o sensor da Figura 33(a) está sem a antena. Isso porque houve problemas com ela durante os testes. Foi observado que com a antena, a maioria dos pacotes não eram enviados com sucesso. Sem a antena o sistema perdeu um pouco de distância útil, mas pelo menos não ocorriam tantas falhas de transmissão de dados. Foram realizadas diversas tentativas de correção, como aterramento e reprogramação, mas o problema não foi resolvido completamente. Este problema pode ter ocorrido devido a defeito físico com o módulo NRF24L01+ ou apenas com sua antena. Optou-se então em utilizar o módulo sem conectar a antena para as três salas mais próxima ao escritório. De qualquer forma, mesmo sem usar a antena, o módulo NRF24L01+ obteve melhor alcance em relação à versão do módulo rádio utilizado anteriormente.

O módulo receptor ficou bem compacto e pôde ser facilmente instalado no computador em que será feito o monitoramento. Na Figura 34 está representado o seu funcionamento.



Figura 34 – Módulo receptor em funcionamento

Fonte: Autor

No *software* foi melhorado a aquisição de dados pela implementação do banco de dados e funções de consulta. A representação por tabelas e gráficos também foi aperfeiçoada. Sua interface foi demonstrada na seção 3.8.1.

Os sensores demonstraram funcionar corretamente. Mas o parâmetros de amônia ainda não está adequado, pois não foi possível averiguar a exatidão de seu valor estável, devido a falta de um sensor comercial de referência. Ou seja, não pode-se afirmar os níveis de ppm medidos pelo sistema ainda.

Os resultados obtidos durante os testes realizados entre os dias 24/07/2017 e

07/08/2017 no biotério definido como sala 1 no supervísório estão dispostos nas Figuras 35, 36, 37, 38 e 39. Todos os gráficos foram gerados pelo supervísório, sem a necessidade de um *software* externo. Foi utilizado a seleção por média horária para os parâmetros, com exceção do ruído que foi detalhado.

Primeiro tem-se o gráfico da amônia (Figura 35). Os distúrbios observados na Figura 35 ocorreram por dois fatores. Primeiro que o sensor de amônia ao ligar, apresenta sinal de saída muito elevado, sendo necessário aguardar até que estabilize. Logo, alguns picos foram observados quando houve queda de energia. Um segundo fator para as instabilidades foi que durante o período de leitura, foram utilizados produtos de limpeza sobre o aparelho, caracterizando uma interferência.

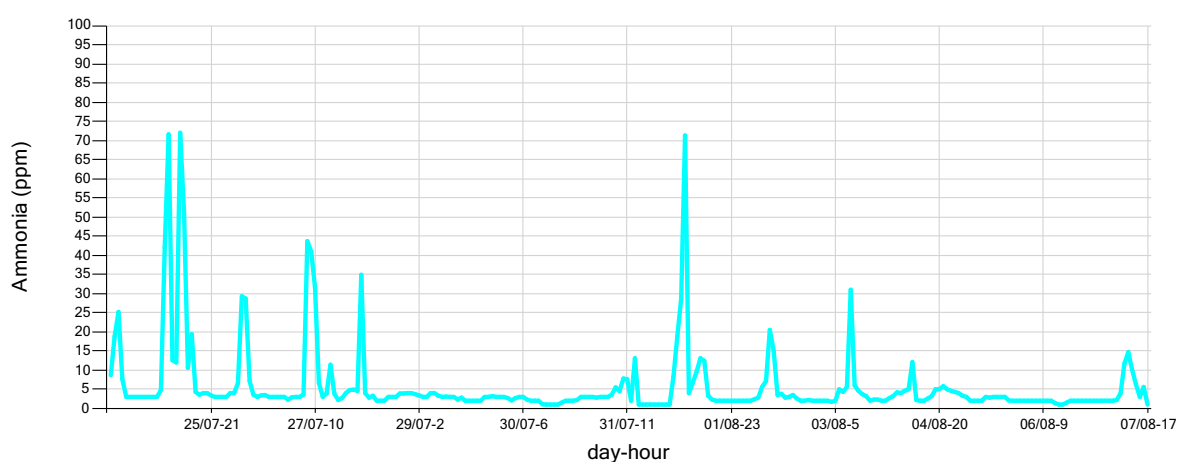


Figura 35 – Gráfico de teor de amônia(ppm) x tempo

Fonte: Autor

Além da calibração necessária para o sensor, foi observado a necessidade de correção dessas instabilidades. Para o primeiro caso sugere-se aguardar um período maior após a inicialização do Arduino, para então começar a fazer as leituras. As interferências podem ser corrigidas via firmware, bastando que os picos no sinal de saída sejam desconsiderados. Outra alternativa seria a utilização de uma bateria de backup, que atue durante a falta de energia.

Os valores de iluminação em lux obtidos foram satisfatórios. Porém, no gráfico de iluminação (Figura 36), percebe-se picos acima de 250lux que correspondem a um período do dia em que ocorre a incidência do sol bem próximo ao sensor. Isso indica que a iluminação do biotério não está adequada, pois nas normas de bioterismo é sugerido uma iluminação artificial e constante de 12 por dia. Futuramente pretende-se controlar a iluminação do biotério, que será integralmente feita por lâmpadas e com isolamento da luz solar.

Os níveis de temperatura e umidade não devem variar tanto, pois os biotérios são

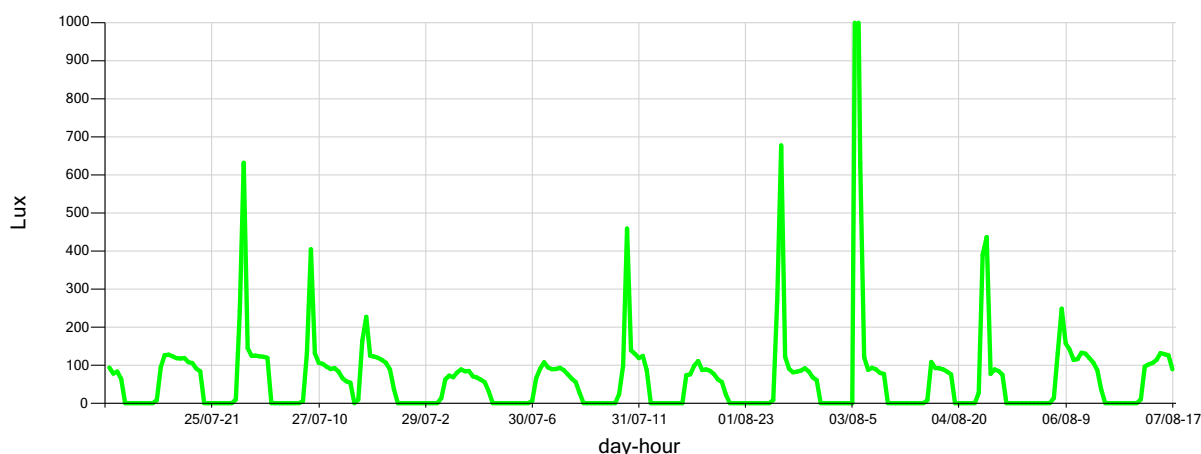


Figura 36 – Gráfico de Iluminância(lux) x tempo

Fonte: Autor

equipados de ar condicionado. Valores de temperatura mais altos são atingidos nas horas mais quentes do dia, isso demonstra um bom funcionamento do sensor. As medidas são importantes para serem registradas caso o sistema de ventilação falhe, por exemplo. Os dados foram coletados em período de inverno, então foram registrados temperaturas abaixo da zona de conforto. Na Figura 37 está representado o resultado da temperatura.

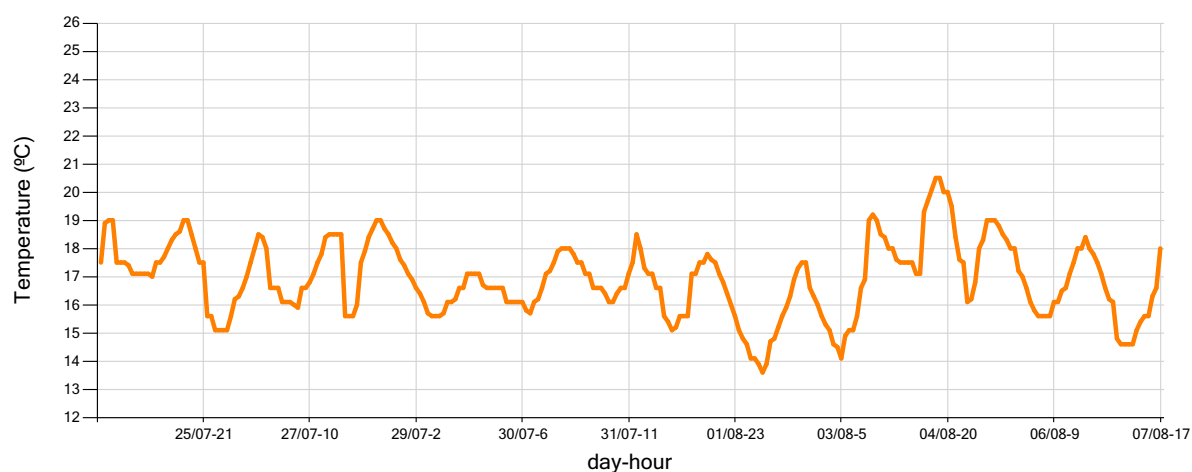


Figura 37 – Gráfico de temperatura(°C) x tempo

Fonte: Autor

Ouro Preto possui um clima de montanha, onde há bastante oscilação da umidade e temperatura. Na Figura 38 estão ilustrados os resultados de umidade relativa do ar, onde pode-se observar essas oscilações.

Foi feito uma consulta com pessoas da área do bioterismo. Foi relatado que o sistema de ventilação não atente totalmente aos padrões de qualidade, principalmente no

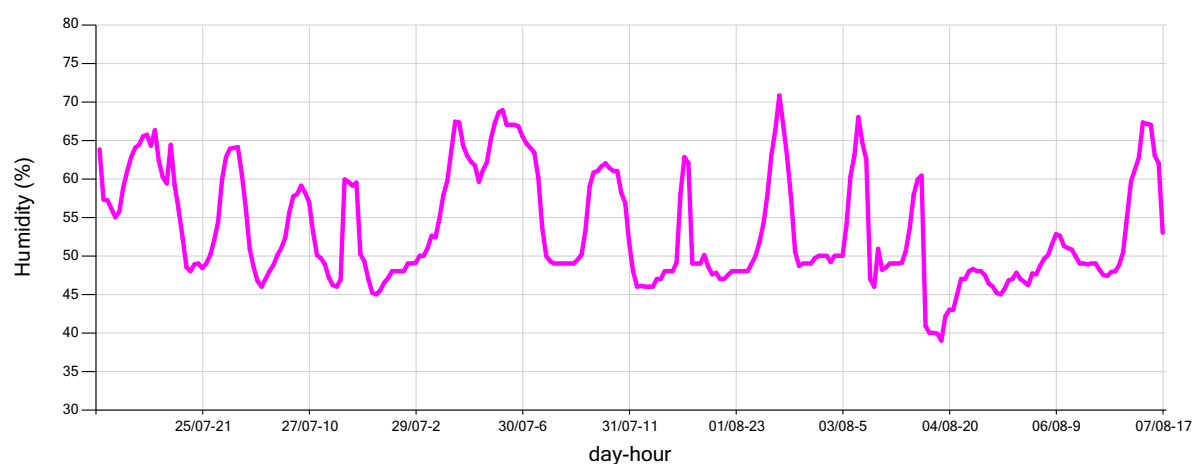


Figura 38 – Gráfico de umidade relativa do ar(%) x tempo

Fonte: Autor

verão onde as temperaturas são maiores.

A respeito do ruído, ele foi mensurado corretamente e averiguado sua intensidade durante testes. Porém, sua representatividade precisa ser melhor elaborada, uma vez que a média de ruídos em um período de tempo em horas será quase que constante durante o dia. Outro fato é que a posição do sensor em relação ao sistema de ventilação influencia muito na intensidade do ruído.

Por esse motivo foi empregado a seleção de dados detalhados, dessa forma os picos de ruído são registrados. O que determina ruído mínimo é o sistema de ventilação, que tem intensidade considerada média, comum para tal sistema. Na Figura 39 está representado seu resultado.

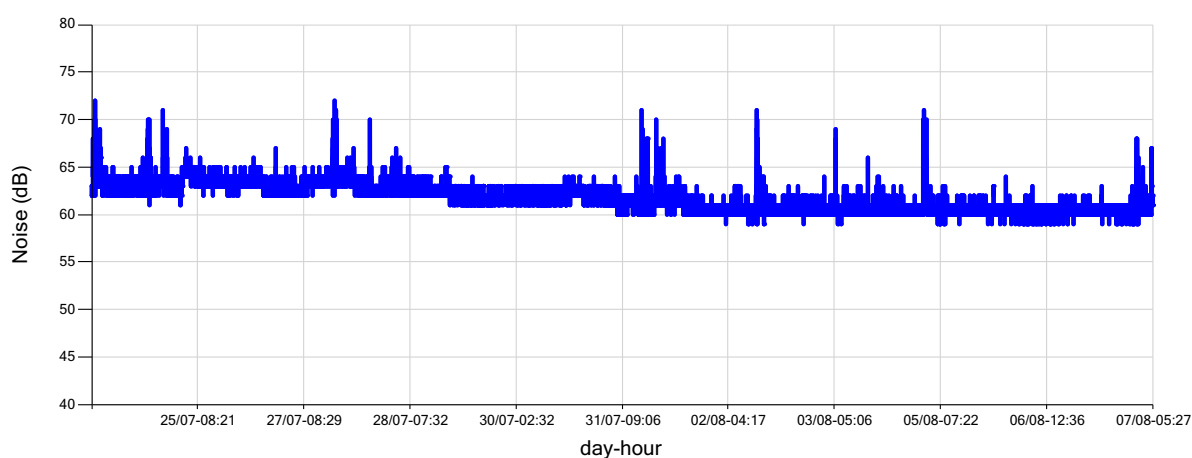


Figura 39 – Gráfico de ruído(dB) x tempo

Fonte: Autor

5 Índice de Temperatura de Globo Negro e Umidade

A partir do aperfeiçoamento do sistema de monitoramento de biotérios que foi apresentado, já é viável a aplicação do mesmo para técnicas automatizadas de análise mais complexas e de controle. A título de exemplo, com os parâmetros medidos pelo sistema é possível criar métodos de controle de iluminação artificial para que os animais recebam luz adequada. A radiação solar incidente no biotério compromete também o ambiente térmico, devendo ser evitada. Bem como o controle eficiente de temperatura e umidade entregues pelo sistema de ventilação traria grandes melhorias no conforto térmico dos animais estudados.

A aplicação de métodos de controle não foi abordada no projeto, visto que achou-se conveniente primeiro um estudo aprofundado em técnicas de análise do ambiente e do animal. O conforto térmico é tratado no presente trabalho como uma dessas possíveis análises, em que determinar o ambiente térmico provê a melhor compreensão das variáveis de temperatura e umidade do ar em função do conforto do animal.

Neste capítulo é apresentado a implementação do globo negro ao módulo de sensores do sistema de monitoramento. Em seguida são calculados índices de temperatura de globo e umidade - ITGU. A finalidade do ITGU é agrupar e padronizar em único valor as condições de umidade do ar, temperatura, ventilação e radiação, que são fatores que comprometem o conforto térmico.

A determinação do conforto térmico em zonas de conforto, frio e calor com respectivas intensidades se faz com várias amostras que envolvem o uso com animais. Por esse motivo, no presente trabalho é calculado o ITGU, mas não será determinado o conforto térmico para os animais do biotério.

5.1 Implementação do globo negro

O globo negro empregado no projeto foi desenvolvido pelo professor coorientador em (NICACIO,). Ele é formado por um sensor analógico de temperatura LM35DZ encoberto por um globo negro oco e possui um conector de áudio padrão do tipo P10 stereo para facilitar a utilização. Na Figura 40 está representado o globo negro descrito.

O globo negro tem essa coloração preta e fosca para maximizar a absorção de radiação solar. Ele indica, através de sua temperatura interna mediada, os efeitos da energia

radiante, temperatura ambiente e velocidade do ar. Esses fatores afetam diretamente o conforto térmico.



Figura 40 – Globo negro

Fonte: Autor

O sensor LM35DZ produz tensão linearmente proporcional ao valor da temperatura em graus centígrados. Não requer calibração, já que em sua folha de dados garante uma variação em sua saída de $10\text{mV}/^{\circ}\text{C}$. Sua faixa de operação está entre 0°C e 100°C e tem precisão de $0,5^{\circ}\text{C}$. É um sensor de baixa corrente que não produz aquecimento considerável, podendo ser empregado para a construção do globo negro.

Na última versão do módulo emissor do sistema de monitoramento foi adicionado uma entrada para receber o globo negro. Um conector P10 fêmea foi disposto na parte frontal do módulo, de forma que permite plugar o globo negro. A Figura 41 ilustra essa ligação.

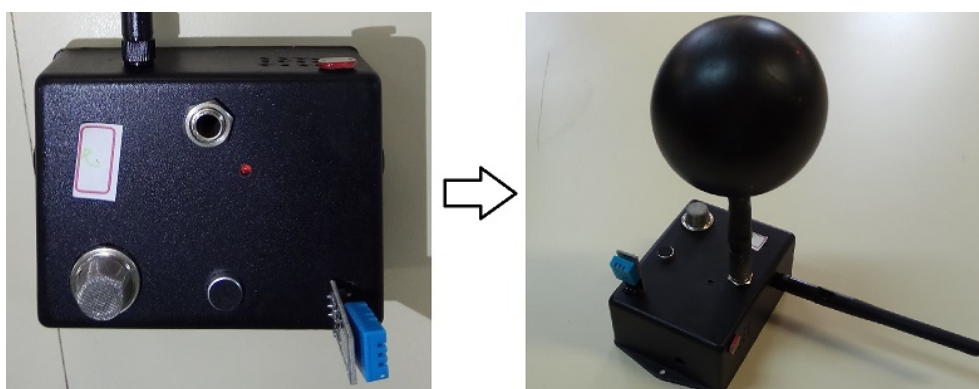


Figura 41 – Conexão do globo negro

Fonte: Autor

5.2 Cálculo do ITGU

Segundo NICACIO, pode-se calcular o índice de temperatura de globo negro e umidade com o valor de temperatura de globo negro e umidade relativa do ar, como na Equação 5.1. Onde T_g é a temperatura de globo negro e UR a umidade relativa do ar em decimal.

$$ITGU = 0,81T_g + UR(0,99T_g - 14,3) + 46,3 \quad (5.1)$$

Esta é uma das equação tradicionais para se obter o ITGU. Na literatura também há definições de equações que relacionam a temperatura de globo negro com a temperatura de bulbo úmido ou com temperatura de ponto de orvalho. Contudo, os resultados são os mesmos para as diferentes equações. O que determinou o uso da Equação 5.1 foram os sensores disponíveis e a referência citada, que faz o seu uso para cálculo de ITGU.

O globo negro já havia sido embutido ao projeto antes dos testes realizados entre os dias 24 de julho e 08 de agosto de 2017 apresentados no Capítulo 4. Logo, com os dados de umidade relativa do ar proveniente do DHT11 e de temperatura de globo negro, foi calculado o ITGU para cada medição. Para isso foi implementado a variável ITGU ao banco de dados, que atualiza conforme a Equação 5.1 para todos os valores de temperatura e umidade registrados. Os valores de ITGU calculados podem ser visualizados no histórico de dados em forma de tabela ou gráfico. Nas Figuras 42 e 43 estão ilustrados as médias horária e diária dos índices calculados pelo *software*.

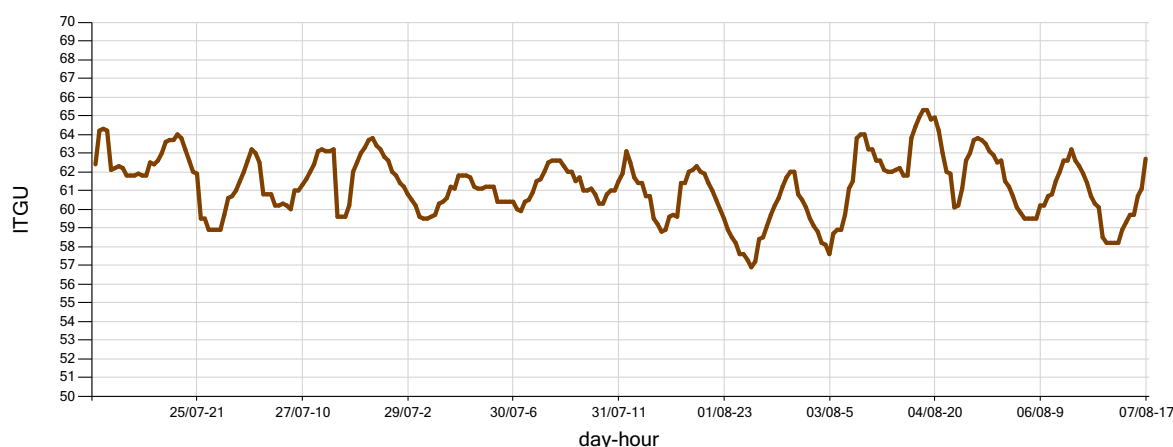


Figura 42 – Gráfico de média horária do ITGU calculado

Fonte: Autor

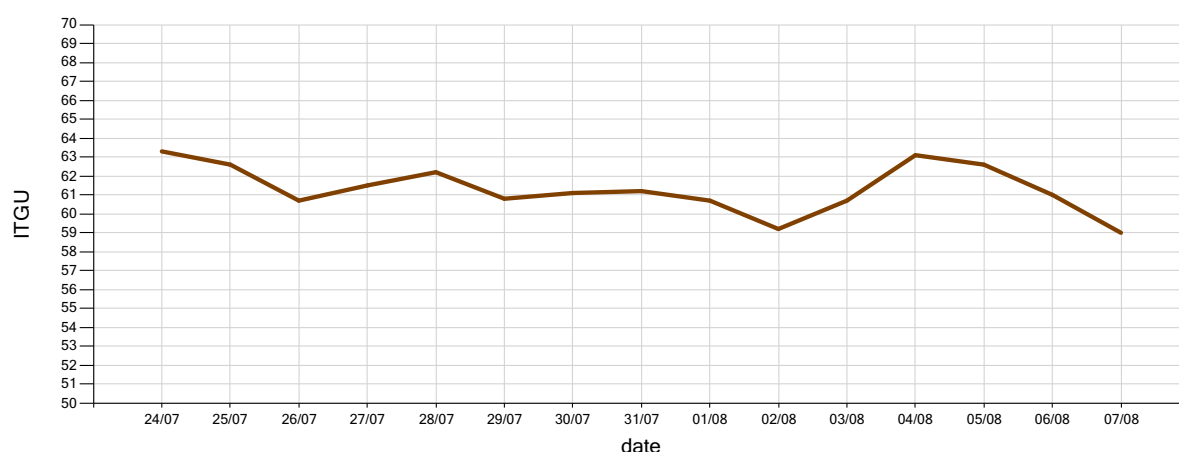


Figura 43 – Gráfico de média diária do ITGU calculado

Fonte: Autor

Durante os testes, o ITGU variou de 56,8 a 65,4. Pela Figura 43 percebe-se que a média está em torno de 61.

Com propósito de comparação, AMARAL et al. demonstra que em codornas, o aumento ITGU provoca redução no consumo de ração, observado pela menor frequência ao comedouro e pelo peso da ração consumida. O autor determina que a faixa de ITGU em que há maior consumo de ração está entre 74,6 e 80,4, atribuindo tal zona como a de conforto. Por outro lado, segundo NICACIO, o conforto térmico ambiental para humanos é dividido em faixas, como na Tabela 1.

Tabela 1 – Determinação do conforto térmico para índices de temperatura de globo negro e umidade

ITGU	55 - 59	61 - 62	63 - 74	75 - 79	80 - 85	86 - 98
Conforto térmico	Frio intenso	Frio	Conforto	Calor	Calor intenso	Calor insuportável

Se for aplicado os valores de conforto determinado para codornas (74,6 a 80,4) na Tabela 1, observa-se que se encaixaria mais zona de calor, aproximando a zona de calor intenso. Isso prova que não se pode determinar a faixa de conforto térmico para espécies ou grupo de espécies diferentes.

Acredita-se que a faixa de conforto térmico dos ratos e camundongos está próxima ao que foi estipulado na Tabela 1, até porque são animais mamíferos como nós. Mas, não foi possível averiguar essa dedução ainda.

6 Visão Computacional para Estudo Comportamental Animal

Neste capítulo é apresentado uma técnica de visão computacional para rastreamento de ratos e camundongos. A aplicação dessa técnica no projeto se faz no estudo comportamental do animal em relação aos parâmetros medidos pelo sistema de monitoramento, inclusive o ITGU que foi implementado. O diferencial do trabalho presente é essa comparação em uma única plataforma.

A proposta foi considerada relevante, devido ao estudo apresentado de que os níveis de temperatura, umidade, iluminância, ruído e amônia no ar interferem no comportamento do animal. Além disso, com o avanço do projeto, como vem sendo aperfeiçoado, será possível usar o sistema integrado para determinar o conforto térmico dos animais.

A partir das comparações dos trabalhos citados no capítulo de revisão bibliográfica, foi determinado que não serão necessários, a primeiro momento, sensores para rastrear o animal, para isso apenas o processamento de imagens da câmera serão considerados. Essa escolha se deve aos bons resultados obtidos do trabalho (CHAROENPONG et al., 2012), em que é empregado essa técnica.

6.1 Materiais e montagem do experimento

Para o teste do sistema foi utilizado um animal da espécie *Cricetulus Griseus* (hamster chinês), um roedor com média de 7,5cm a 9cm de comprimento. É um hamster doméstico já acostumado em viver isolado em gaiola acolchoada de serragem, com água e alimentação a vontade. Possui coloração escura, saudável e de bom temperamento. Foi optado o uso do hamster por estar acessível e representar similaridade com ratos e camundongos. Considere também que a proposta inicial é apenas testar o sistema de rastreamento.

Não foi executado nenhum procedimento ou intervenção física que comprometa a saúde do animal. Além disso, o teste foi realizado por um curto período de tempo e após isso o animal foi devolvido a sua gaiola.

Além do sistema de monitoramento desenvolvido, foram utilizados: uma webcam de resolução SD(640 x 480 pixel); uma caixa plástica com 570mm de comprimento, 410mm de largura e 350mm de altura; itens de consumo para hamster doméstico, como papel de forro branco, serragem, ração e água em recipientes adequados.

O tipo de experimento escolhido foi o de arena de teste em campo aberto, com adaptação para o tipo de animal utilizado. A caixa foi forrada com papel branco neutro e serragem especial para roedores. A webcam foi adaptada a um suporte de forma que capture imagens com vista de cima da área de circulação do hamster. O intuito dessa posição é representar a movimentação do animal em duas dimensões apenas, uma vez que não há obstáculos que ele possa subir ou descer. O módulo emissor com os sensores do sistema de monitoramento foi posicionado no interior da caixa, em uma altura próxima do limite alcançável do animal, buscando representar mais fielmente as condições ambientais sofridas por ele. Na Figura 44 observa-se a configuração dos testes executados.



Figura 44 – Configuração dos Testes em Campo Aberto

Observe que foi alcançado uma iluminação incandescente fraca e indireta dentro da área de testes, o que é ideal para os roedores. O experimento foi realizado sem controle de temperatura e umidade, mas o clima estava também favorável para o hamster. Os ruídos do ambiente não foram controlados também, ocorrendo alguns picos periodicamente, o que simula a manutenção de biotérios, por exemplo. A sala é arejada, evitando o acúmulo de partículas de amônia no ar.

6.2 Software

Todo o processamento de imagens é feito graças a biblioteca de código aberto OpenCV. Como o projeto foi desenvolvido em plataforma .NET, especificamente em linguagem de programação C#, foi utilizado a extensão da biblioteca OpenCV própria para esse tipo de aplicação, a EmguCV.

A seleção dos *softwares* de desenvolvimento e da biblioteca OpenCV foi devido a sua facilidade de programação e por serem gratuitas. Assim como o Arduino, que segue a iniciativa de código aberto. As plataformas *Open Source* representam um compartilhamento eficaz de informações pela internet, onde é possível encontrar bibliotecas como a OpenCV gratuitamente e compartilhamento de códigos e projetos desenvolvidos.

O *software* foi desenvolvido na mesma solução de projetos do Visual Studio 2017 do programa de monitoramento. A tela de análise comportamental é acessada a partir da tela principal do programa. A Figura 45 ilustra a janela de análise comportamental desenvolvida.

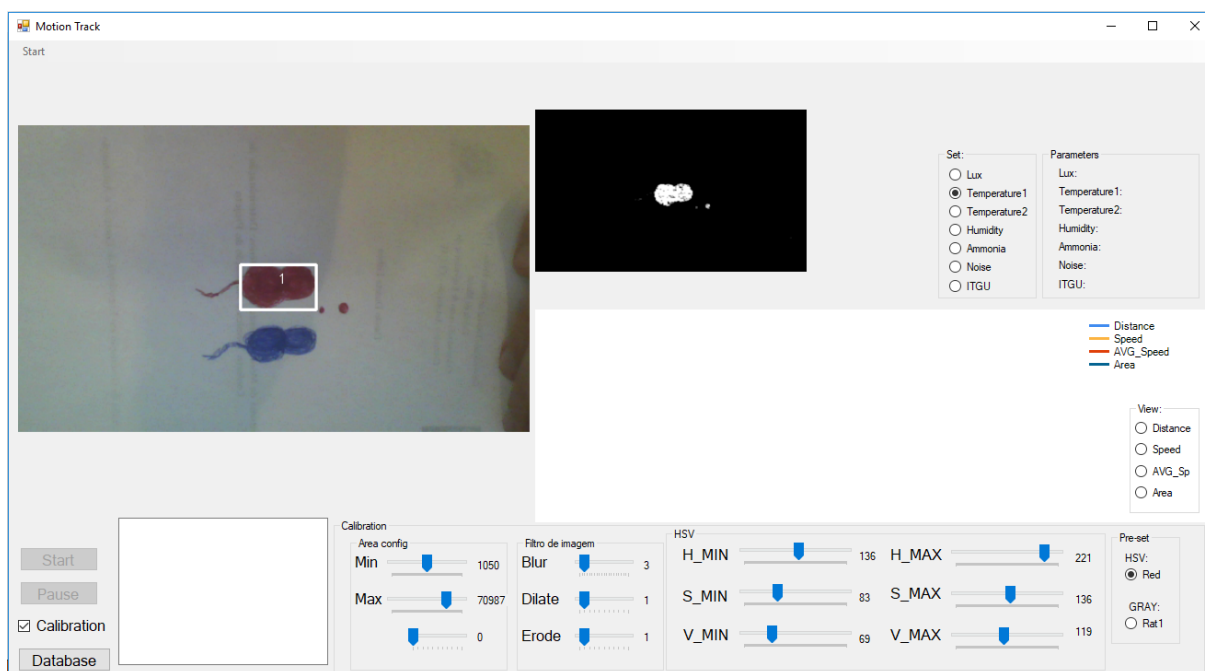


Figura 45 – Janela de análise comportamental

Fonte: Autor

Na aba superior esquerda é selecionado a fonte de imagens, que pode ser de um arquivo de vídeo, de uma câmera conectada ao computador ou uma câmera IP. Ao conectar com a fonte de imagem irá aparecer dois quadros. O quadro da esquerda para visualização real do vídeo e o da direita que mostra a imagem tratada pelo *software*. Há também filtros para calibração do tratamento de imagem e um gráfico de que monitora distância,

velocidade instantânea, velocidade média ou área do objeto detectado com indicativos de alteração do parâmetro ambiental selecionado.

Na Figura 45 tem-se um exemplo de calibração que foi salva para objetos de cor vermelha e com tratamento de ruídos. Note que na imagem tratada, apenas o objeto em vermelho aparece e o azul é apagado. Dois pontos em vermelho foram desenhados para representar objetos indesejados ou ruídos da câmera. O *software* consegue determinar o tamanho mínimo e máximo a ser considerado. Neste caso, o ponto vermelho menor foi filtrado da imagem e o ponto maior aparece, mas não é considerado como objeto de rastreamento. O quadrado em branco com a numeração é gerado em cima do objeto detectado. Isso foi implementado em uma função para poder visualizar a execução do *software*.

No botão "Database" da Figura 45, o usuário tem acesso aos dados gravados para uma análise posterior. É aberto uma nova janela para consulta desses dados, como na parte de monitoramento. Pode ser visualizado dois eixos verticais simultaneamente, um de parâmetro ambiental e outro da variável comportamental.

6.3 Funcionamento e implementações

Entre as variáveis de entrada para o processamento de imagem estão: tamanho do objeto, cor, formato, contraste de cores e variações entre quadros de imagem em vídeo. As entradas são trabalhadas no *software* desenvolvido para gerar as variáveis de trabalho, ou seja, que são utilizadas como parâmetros de comportamento do roedor. Posição, distância percorrida, velocidade, período de captura de imagens e área detectada compõem dessas variáveis de trabalho.

Algumas variações de posição do animal e de sua área também podem ser analisadas. Como por exemplo, se ele está esticado na horizontal ou em pé, que pode representar comportamentos e ações a serem investigados como correr, comer, ficar de pé para investigar ou cheirar, entre outros.

As variáveis ambientais são adquiridas e atualizadas automaticamente do programa principal, bastando estar conectado ao módulo receptor que capta os dados dos sensores via comunicação *wireless*. A variável ambiental a ser comparada com a comportamental pode ser selecionada no programa.

A captura de um vídeo depende das limitações da câmera. Em geral uma câmera comum processa em até 30 fotos por segundo, que equivale a um período de 33ms entre cada quadro. A luminosidade do local determina a qualidade do vídeo, ou seja, quanto mais luz, mais rápida é feita a captura de uma imagem pela câmera. Considerando também que a velocidade máxima do hamster será inferior a 1 m/s dentro da caixa, que equivalem a 10cm percorridos em 100ms, esse foi o período de atualização e processamento de

imagem da câmera. Observe que o período de 100ms utilizado está bem acima do limite da câmera de 33ms. Isso foi adotado para garantir o processamento de todos os dados e da imagem com pouca iluminação.

Para arquivos de vídeo, é levado em consideração o período de atualização dos quadros do vídeo. Em um vídeo com 24 quadros por segundo por exemplo, a base de tempo entre os quadros para se calcular velocidade e aceleração, deve ser de 41,67ms.

Para efetuar a calibração, inicialmente deve-se ajustar os níveis de H(matriz), S(saturação) e V(valor) para as imagens coloridas que serão convertidas em HSV. No caso que foi estudado, a imagem tinha poucas cores e contraste, sendo mais eficiente utilizar o tratamento de uma imagem em escala de cinza, realçando os contornos. Neste caso, apenas o valor de H é ajustado. Destacado o objeto, os demais filtros devem ser ajustados.

A base de tempo é controlada pelo recurso *timer* do C#, que garante a execução periódica da função de processamento de imagem. Essa função é descrita no Algoritmo 3. Quando em execução, o *software* faz um tratamento a cada 10 amostras de posição para calcular deslocamento, distância total, velocidade, velocidade média, área e período de captura no intervalo determinado. Exibe as informações tratadas em tempo real e as salva em banco de dados junto aos parâmetros ambientais atualizados.

Os filtros utilizados na função descrita no Algoritmo 3 são aplicados de funções do OpenCV, que permitem ser chamadas através do Emgu CV em linguagem C#. É considerado dez amostras de imagem para interpretar o deslocamento do animal, dessa forma ruídos são amenizados e ainda economiza na armazenagem de dados.

Algorithm 3 Função de rastreo do animal

```

função RASTREIA OBJETO
  imgOriginal  $\leftarrow$  novo quadro de imagem
  se imgOriginal = vazio então                                 $\triangleright$  Caso não capture imagem, retorna
    return
  fim se
  tempo_antigo  $\leftarrow$  tempo_novo
  tempo_novo  $\leftarrow$  lê tempo atual
  x_antigo  $\leftarrow$  x_novo
  y_antigo  $\leftarrow$  y_novo
  Corta imgOriginal                                            $\triangleright$  Tira bordas da imagem
  imgCinza  $\leftarrow$  conversão de imgOriginal em escala de cinza
  imgCinza  $\leftarrow$  limites de matriz(H)
                                                                 $\triangleright$  Operações para filtrar imagem:
  imgCinza  $\leftarrow$  operação Gaussiana para homogeneizar imagem
  imgCinza  $\leftarrow$  operação de contrair região detectada
  imgCinza  $\leftarrow$  operação de dilatar região detectada
  Delimita área mínima e máxima para detecção de objeto
  Executa função de detectar objeto
  Desenha retângulo sobre objeto
  area  $\leftarrow$  área do retângulo
  x_atual  $\leftarrow$  x retornado da função
  y_atual  $\leftarrow$  y retornado da função
  se Modo de rastreo então
    dt  $\leftarrow$  tempo_atual – tempo_antigo
    dx  $\leftarrow$  dx + x_atual – x_antigo
    dy  $\leftarrow$  dy + y_atual – y_antigo
    area_media  $\leftarrow$  area_media + area
    amostras  $\leftarrow$  amostras + 1
    se Acumulou 10 amostras então
      dxy  $\leftarrow$  cálculo da hipotenusa de dx e dy
      distancia  $\leftarrow$  distancia + dxy
      velocidade  $\leftarrow$  dxy/(dt * 1000)
      tempo_total  $\leftarrow$  tempo_atual – tempo_inicio
      velocidade_media  $\leftarrow$  distancia/tempo_total
      area_media  $\leftarrow$  area_media/10
      Salva variáveis no banco de dados
      Atualiza gráfico
      Zera variáveis acumuladas
    fim se
  fim se
  Mostra imagens
fim função

```

6.4 Realização do experimento

Para a realização do experimento, foi posto em prática a configuração da Figura 44 com todas as observações feitas e o hamster no local. Na Figura 46 observa-se o programa em execução. O rastreo é iniciado e então a calibração de detecção do animal é realizada.

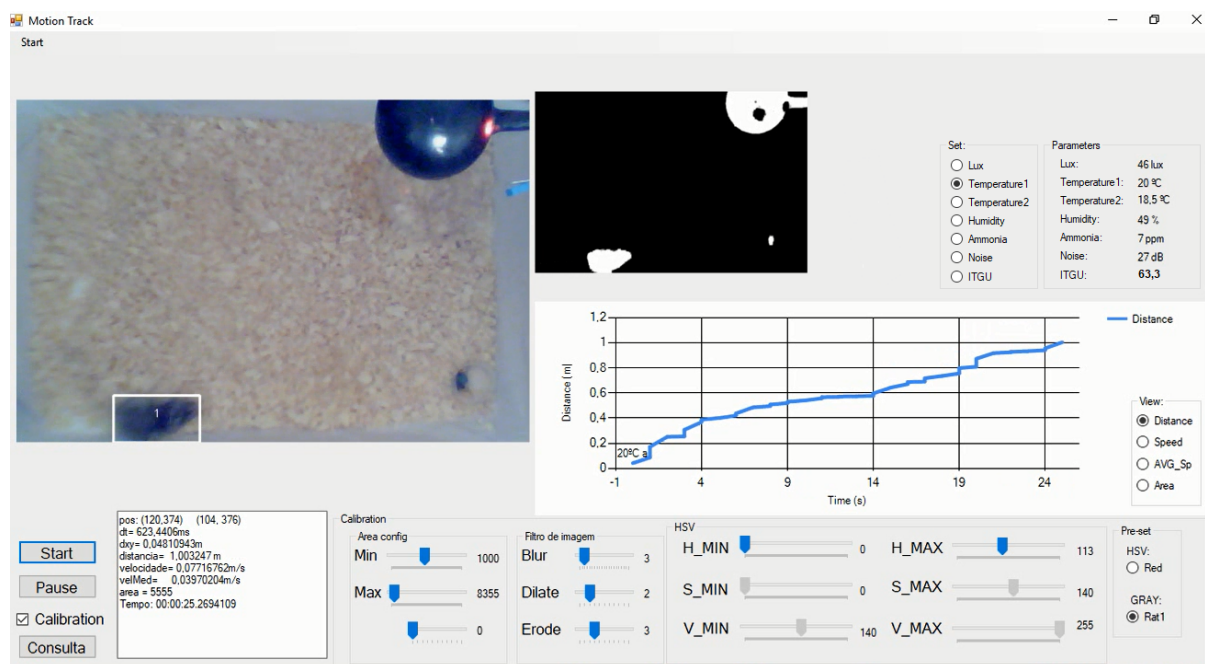


Figura 46 – Software em execução

Fonte: Autor

No quadro direito da Figura 46 está representado o tratamento de imagem feito pelo *software*, no qual ele "enxerga" apenas o hamster, a comida e o sensor que está acima. A água está logo abaixo do sensor em um recipiente transparente.

Na calibração, foi necessário usar o filtro de tamanho de objeto, podendo determinar uma faixa de área compreendida como o objeto de rastreo (hamster). Foi ajustado, então, uma área mínima um pouco maior que a da comida no canto direito da imagem de tratamento e ligeiramente menor que a área do sensor. Dessa forma não houve interferência de outros objeto na detecção do animal.

Feito a calibração, o botão "Start" é acionado e os dados começam a ser processados e salvos em banco de dados. O gráfico de distância percorrida e temperatura em função do tempo é observado da Figura 46. Cada alteração no parâmetro ambiental é registrado sobre o gráfico, com o intuito de assimilar tal mudança com o comportamento.

6.5 Resultados preliminares

Um teste foi executado no intervalo de cinquenta minutos e então registrado todos os dados. Na Figura 47 estão alguns dados representados através da janela de consulta.

Id	Date	Time	Period	Dx	Dy	Distance	Speed	DistanceSUM	SpeedSUM	Area	Lux	Temperature1	Temperature2	Humidity	Ammonia	Noise	ITGU
3630	31/08/2017	01:15:35	608.45739...	-0.0269606...	-0.00086...	0.026974...	0.04433...	1.9271219968...	0.02481675...	5824.79...	46	20	18.5	49	7	27	63.3
3631	31/08/2017	01:15:35	630.44274...	-0.0086969...	0.020872...	0.022612...	0.03586...	1.9497342109...	0.02490574...	6886.79...	46	20	18.5	49	7	27	63.3
3632	31/08/2017	01:15:36	601.42938...	-0.0008696...	-0.00086...	0.001229...	0.00204...	1.9509640932...	0.02473145...	6951.39...	46	20	18.5	49	7	27	63.3
3633	31/08/2017	01:15:37	623.43884...	0.0017393...	0.002609...	0.003135...	0.00502...	1.9540998935...	0.02457697...	6614.60...	46	20	18.5	49	7	27	63.3
3634	31/08/2017	01:15:37	608.43420...	-0.0008696...	0.032178...	0.032190...	0.05290...	1.9862905740...	0.02479211...	6492.39...	46	20	18.5	49	7	27	63.3
3635	31/08/2017	01:15:38	623.44097...	-0.0069575...	0.042615...	0.043179...	0.06926...	2.0294702053...	0.02513547...	6395.39...	46	20	18.5	49	7	27	63.3
3636	31/08/2017	01:15:38	624.44421...	-0.0200030...	0.020872...	0.028910...	0.04629...	2.0583803653...	0.02529788...	6292.60...	46	20	18.5	49	7	27	63.3
3637	31/08/2017	01:15:39	615.43707...	-0.0373971...	0.001739...	0.037437...	0.06083...	2.0958178043...	0.02556463...	6109.60...	46	20	18.5	49	7	27	63.3
3638	31/08/2017	01:15:40	616.45733...	-0.0600093...	-0.04435...	0.074622...	0.12104...	2.1704399585...	0.02627727...	8013.79...	46	20	18.5	49	7	27	63.3
3639	31/08/2017	01:15:40	609.43292...	-0.0278303...	-0.02348...	0.036413...	0.05974...	2.2068533897...	0.02652243...	8358.40...	46	20	18.5	49	7	27	63.3
3640	31/08/2017	01:15:41	623.44140...	0.0017393...	-0.01913...	0.019212...	0.03081...	2.2260656356...	0.02655437...	8162.20...	46	20	18.5	49	7	27	63.3
3641	31/08/2017	01:15:41	607.45288...	-0.0095667...	-0.02609...	0.027789...	0.04574...	2.2538552284...	0.02659245...	8078.39...	46	20	18.5	49	7	27	63.3
3642	31/08/2017	01:15:42	624.44378...	0.0078273...	0	0.007827...	0.01253...	2.2616825103...	0.02658852...	7495.20...	46	20	18.5	49	7	27	63.3
3643	31/08/2017	01:15:43	608.43103...	0.0191334...	-0.01478...	0.024180...	0.03974...	2.2858626842...	0.02668193...	6878.39...	46	20	18.5	49	7	27	63.3
3644	31/08/2017	01:15:43	622.44140...	-0.0026090...	0.000869...	0.002750...	0.00441...	2.2886120425...	0.02652134...	6125.79...	46	20	18.5	49	7	27	63.3
3645	31/08/2017	01:15:44	609.43383...	0	0.001739...	0.001739...	0.00285...	2.2903523445...	0.02635537...	6218	46	20	18.5	49	7	27	63.3
3646	31/08/2017	01:15:45	629.44799...	0	0.000869...	0.000869...	0.00138...	2.2912220954...	0.02617578...	6261.79...	46	20	18.5	49	7	27	63.3
3647	31/08/2017	01:15:45	602.47009...	-0.0008696...	0.001739...	0.001944...	0.00322...	2.2931668758...	0.02601891...	6326	46	20	18.5	49	7	27	63.3
3648	31/08/2017	01:15:46	624.44421...	-0.0034787...	-0.00347...	0.004919...	0.00787...	2.2980866432...	0.02589129...	6435.60...	46	20	18.5	49	7	27	63.3
3649	31/08/2017	01:15:46	606.43139...	-0.0017393...	0.002609...	0.003135...	0.00517...	2.3012223243...	0.02575068...	6509.39...	46	20	18.5	49	7	27	63.3
3650	31/08/2017	01:15:47	623.46148...	-0.0069575...	0.005218...	0.008697...	0.01394...	2.3099193572...	0.02566892...	6044.20...	46	20	18.5	49	7	27	63.3
3651	31/08/2017	01:15:48	615.41491...	-0.0026090...	0.005218...	0.005834...	0.00947...	2.3157534599...	0.02555896...	5504.20...	46	20	18.5	49	7	27	63.3
3652	31/08/2017	01:15:48	617.43847...	0.0008696...	-0.00434...	0.004434...	0.00718...	2.3201880455...	0.02543458...	5575.39...	46	20	18.5	49	7	27	63.3

Figura 47 – Parte da tabela gerada do teste realizado

Fonte: Autor

Além do período observado na Figura 47 ser muito curto, o teste foi executado pela madrugada e não apresentou variações de temperatura e umidade relativa do ar. Foi registrado um ITGU de 63,3. Uma proposta de utilização da plataforma seria realizar um experimento em que possa variar a temperatura e umidade relativa no microambiente. Então associar o resultado do processamento das imagens com essa variação para determinar o conforto térmico do animal.

No teste, apenas houve variação de ruído. Na Figura 48 está representado com o mesmo filtro de tempo os gráficos de velocidade momentânea em laranja e de ruído em azul. As variáveis de visualização podem ser alteradas pelo caixas ao lado direito.

O resultado foi promissor, pois pareceu a primeiro momento haver alguma relação da velocidade do hamster com o ruído do ambiente. Então, foi averiguado os dados no Excel com a função *CORREL()* em busca de alguma correlação. Porém, o índice de correlação encontrado foi de 0.108, o que caracteriza falta de relação entre os dados. Uma segunda tentativa foi realizada, porém excluindo os dados obtidos a partir de 01:43h, que notoriamente pelo gráfico percebe-se menos relação. O índice de correlação obtido foi de 0.203.

Possivelmente o tempo observado foi curto para se calcular o índice de correlação.

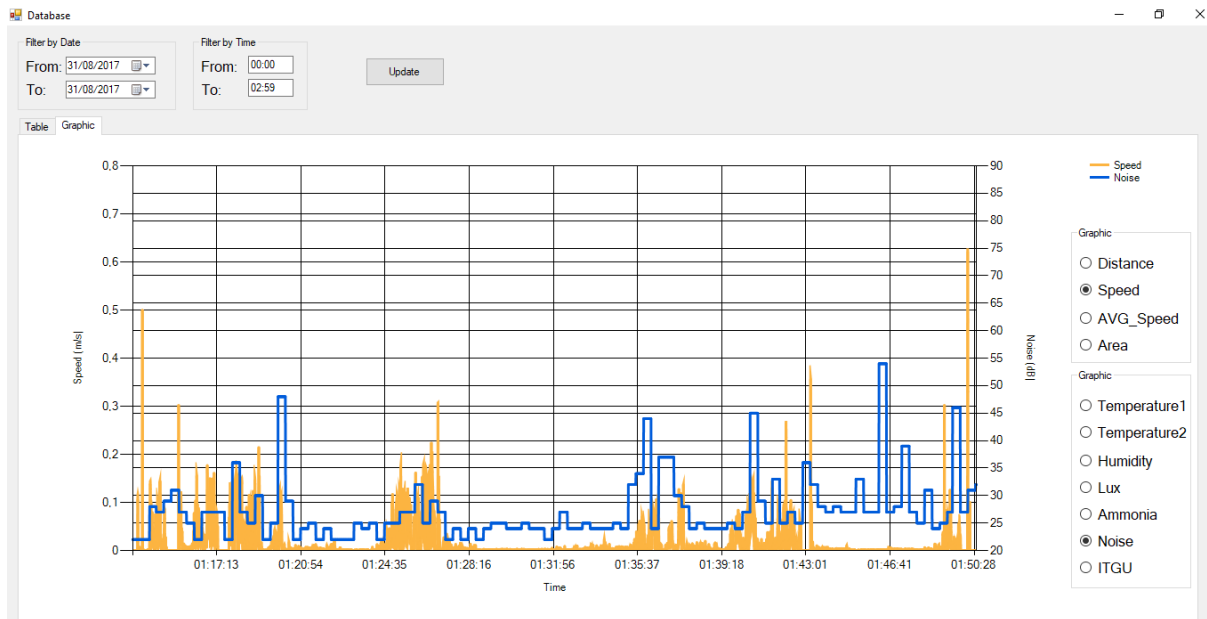


Figura 48 – Gráficos de ruído e velocidade do hamster em função do tempo

Fonte: Autor

Além disso, a representatividade do ruído em relação à velocidade foi diferente, uma vez que a taxa de atualização da velocidade foi muito maior que a do ruído, devido às limitações do sistema. Sendo assim, espera-se que a comparação realizada com variáveis de alteração lenta, como temperatura e distancia deslocada irá resultar em melhores resultados. Mas para isso seria necessário um longo período de coleta de dados e possibilidade alterar o ambiente.

Se neste último experimento fossem obtidos volume e variação suficientes dos dados de ITGU, como os resultados da primeira etapa do projeto, talvez seria possível alcançar bons resultados de correlação. Sendo assim, poderia ser estimado uma zona de conforto térmico relacionado ao registro comportamental do animal. Mas como mencionado, o teste teve limitações, não podendo ser executando por um longo período de tempo.

O programa demonstrou calcular bem os parâmetros de deslocamento, velocidade e área detectada. Porém notou-se, a necessidade de implementar novas abordagens ao sistema criado. Funções que registram o número vezes e tempo que o animal foi ao comedouro e bebedouro seria a primeira recomendação para o aperfeiçoamento do *software*.

7 Conclusão

De acordo com os resultados apresentados neste trabalho, pode-se concluir que o aperfeiçoamento do sistema de monitoramento de biotérios teve resultados satisfatórios. Como principal melhoria, o *hardware* finalmente foi encapsulado, podendo ser utilizado nos biotérios, sem comprometer as normas de operação. Os módulos encapsulados puderam ser instalados facilmente no biotério, sem nenhuma intervenção na estrutura do mesmo. Os sensores apresentaram bom funcionamento, com exceção do MQ-135, já que não foi possível conferir sua exatidão.

A implementação do banco de dados com consulta de gráficos e tabelas facilitou muito no acompanhamento e visualização dos valores. Agora não é preciso mais copiar arquivo com dados para visualizar em outro *software*. O gráfico em tempo real de um parâmetro em múltiplas salas auxiliou na comparação entre elas.

A melhoria na transmissão de dados permitiu monitorar até três salas ao mesmo tempo. O objetivo deste trabalho seria de cinco salas, mas o problema com a antena no módulo NRF24L01 comprometeu o resultado. Espera-se que isso seja resolvido em breve.

A implementação do globo negro ao projeto permitiu um monitoramento mais detalhado das condições térmicas sofridas pelos animais e pessoas no biotério. Com o cálculo do ITGU já é possível realizar comparações com valores que são considerados de conforto térmico para humanos na literatura. Determinar o conforto térmico para os animais em estudo requer mais tempo de pesquisa.

A movimentação do roedor foi captada adequadamente na maior parte do tempo. As variáveis foram calculadas em uma base de tempo fixa, mantendo uma confiança nos cálculos. O objetivo em criar uma plataforma integrada para análise comportamental automática com comparação de parâmetros ambientais foi atingido.

Para trabalhos futuros, o autor recomenda a implementação de alarmes para os níveis de parâmetros ambientais medidos e para falhas do sistema. Os alarmes poderiam ser enviados via e-mail para os supervisores do biotério. Para a aplicação de visão computacional, recomenda-se a implementação de um algoritmo que estime a posição do roedor no caso em que a detecção do mesmo é perdida. O registro de tempo em que o animal encontra-se comendo, ou se hidratando e o reconhecimento de padrões de imagem que determine diversos tipos de comportamento, como levantar, roer, cheirar, entre outros. Para isso seria necessário uma câmera de melhor qualidade e mecanismos de processamento de imagens mais avançado, mas que já existem na biblioteca OpenCV.

Com a plataforma mais eficaz e atualizada, seria então possível a implementação de aprendizado de máquina, em que o *software* será capaz de determinar um comportamento, sensação ou até mesmo diagnosticar alguma doença do roedor através do seu rastreo por imagem com as equivalências de parâmetros ambientais. Esse é um trabalho que requer um longo estudo na área e uma base de dados muito grande.

Referências

- AMARAL, A. G. d. et al. Processamento de imagens digitais para avaliação do comportamento e determinação do conforto térmico de codornas de corte. 2012. Universidade Federal de Viçosa, 2012. Citado 2 vezes nas páginas 17 e 59.
- ANGNES, D. Introdução ao microcontrolador basic step. **Santa Cruz do Sul/RS. Disponível em:** <http://www.radiopoint.com.br/>, acesso dia, 2003. v. 4, n. 04, p. 2011, 2003. Citado na página 18.
- ARDUINO. **Arduino Nano Description**. 2017. Disponível em: <<https://store.arduino.cc/usa/arduino-nano>>. Citado na página 19.
- CARDOSO, T. A. de O. Considerações sobre a biossegurança em arquitetura de biotérios. **Boletim Central Panamaense Fiebre Aftosa**, 2001. v. 64, n. 67, p. 3–17, 2001. Citado 2 vezes nas páginas 10 e 15.
- CHAROENPONG, T.; PROMWORN, Y.; THANGWIWATCHINDA, P.; SENAVONGSE, W.; THONGSAARD, W. An experimental setup for measuring distance and duration of rat behavior. In: IEEE. **Biomedical Engineering International Conference (BMEi-CON)**, 2012. [S.l.], 2012. p. 1–5. Citado 3 vezes nas páginas 23, 24 e 60.
- CLOUGH, G. Environmental effects on animals used in biomedical research. **Biological Reviews**, 1982. Wiley Online Library, v. 57, n. 3, p. 487–523, 1982. Citado na página 17.
- CRUZ, I. T. F.; BARCELOS, M. L. de; FILHO, P. C. G. de B.; OGAWA, W. N. Estresse sonoro e suas implicações no eixo hipotálamo-hipófise-adrenal de ratos. **Revista Cereus**, 2010. v. 1, n. 1, 2010. Citado na página 17.
- ELECTRONICS, N. Y. **CZN-15E**. 2017. Disponível em: <<https://www.sunrom.com/get-/585900>>. Citado na página 31.
- EMGUCV, T. **Main Page EmguCV**. 2017. Disponível em: <http://www.emgu.com/wiki/index.php/Main_Page>. Citado na página 23.
- ESMAY, M. L. et al. Principles of animal environment. **Principles of animal environment.**, 1969. Westport, Conn.: Avi Publishing Co., Inc., 1969. Citado na página 17.
- GHOSH, S.; ROY, A.; SINGH, S.; SAHA, H.; OJHA, V. K.; DUTTA, P. Sensor array for manhole gas analysis. In: IEEE. **Physics and Technology of Sensors (ISPTS), 2012 1st International Symposium on**. [S.l.], 2012. p. 9–12. Citado na página 38.
- HUNZIKER, M. H. L.; MORENO, R. Análise da noção de variabilidade comportamental. **Psicologia: Teoria e Pesquisa**, 2000. SciELO Brasil, v. 16, n. 2, p. 135–143, 2000. Citado 2 vezes nas páginas 11 e 23.

ISHII, H.; OGURA, M.; KURISU, S.; KOMURA, A.; TAKANISHI, A.; IIDA, N.; KIMURA, H. Development of autonomous experimental setup for behavior analysis of rats. In: IEEE. **Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on**. [S.l.], 2007. p. 4152–4157. Citado 2 vezes nas páginas 23 e 24.

JESUS, G. C. **Desenvolvimento de uma plataforma de monitoramento do ambiente via wireless para o biotério da UFOP**. 51 p. Monografia — Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2017. Citado 4 vezes nas páginas 10, 26, 27 e 38.

LUZ, J. Luminotécnica. **Obtido de <http://www.iar.unicamp.br/lab/luz/ld/Livros>**, 2010. 2010. Citado 2 vezes nas páginas 34 e 52.

MAJEROWICZ, J. et al. **Procedimentos de biossegurança para as novas instalações do laboratório de experimentação animal (Laean) de Bio-Manguinhos**. Tese (Doutorado) — Instituto de Tecnologia em Imunobiológicos, 2005. Citado 3 vezes nas páginas 11, 15 e 16.

MCROBERTS, M. Arduino básico. **São Paulo: Novatec**, 2011. 2011. Citado 2 vezes nas páginas 18 e 19.

MICROSOFT. **Microsoft Visual Studio IDE**. 2017. Disponível em: <<https://www.visualstudio.com/pt-br/vs/>>. Citado na página 21.

NICACIO, J. V. Medidor do Índice termico ambiental. Citado 5 vezes nas páginas 11, 17, 56, 58 e 59.

OPENCV, T. **About OpenCV**. 2017. Disponível em: <<https://opencv.org/about.html>>. Citado na página 22.

PELLIZZONI, L. Desenvolvimento de arquitetura de software reutilizável para implementação de aplicações em. net. 2017. 2017. Citado na página 22.

PEREIRA, J. H. V. **SISTEMA DE MONITORAMENTO PARA BIOTÉRIOS**. Monografia (Graduação em Engenharia de Controle e Automação) — Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2014. Citado na página 10.

POLITI, F.; PIETRO, R.; SALGADO, H. R. N.; MAJEROWICZ, J.; CARDOSO, T. d. O. Caracterização de biotérios, legislação e padrões de biossegurança. **Revista de Ciências Farmacêuticas Básica e Aplicada**, 2008. p. 17–28, 2008. Citado 2 vezes nas páginas 15 e 16.

SANCHEZ, M. S. **MONITORAMENTO DE TEMPERATURA, UMIDADE, LUMINOSIDADE, NÍVEL DE RUÍDO E TEOR DE AMÔNIA EM BIOTÉRIOS UTILIZANDO ARDUINO**. 103 p. Monografia (Graduação em Engenharia de Controle e Automação) — Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2014. Citado 3 vezes nas páginas 10, 25 e 26.

SHI, Q.; MIYAGISHIMA, S.; FUMINO, S.; KONNO, S.; ISHII, H.; TAKANISHI, A. Development of a cognition system for analyzing rat's behaviors. In: IEEE. **Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on**. [S.l.], 2010. p. 1399–1404. Citado na página 23.

TECHNOLOGIES, S. **Light Dependent Resistor - LDR**. 2008. <http://kennarar.vma.is/thor/v2011/vgr402/ldr.pdf>. Citado 2 vezes nas páginas 35 e 36.

TECHNOLOGIES, S. **DHT11 - Humidity and Temperature Sensor**. 2012. <http://www.sunrom.com/get/328100>. Citado na página 37.

TMRH20. **TMRh20 nRF24/RF24**. 2017. Disponível em: <<https://github.com/nRF24/RF24>>. Citado na página 39.

WENDLING, M. Sensores. **Universidade Estadual Paulista. São Paulo**, 2010. 2010. Citado 2 vezes nas páginas 20 e 21.

ZHANG, Z.; GUAN, J.; CHANG, W.; WANG, W.; SUN, M.; ODAGAKI, M.; LIU, T.; DUAN, F. Automatic motion tracking and data analysis system for a rat. In: **IEEE. Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on**. [S.l.], 2016. p. 1691–1696. Citado 2 vezes nas páginas 23 e 24.