

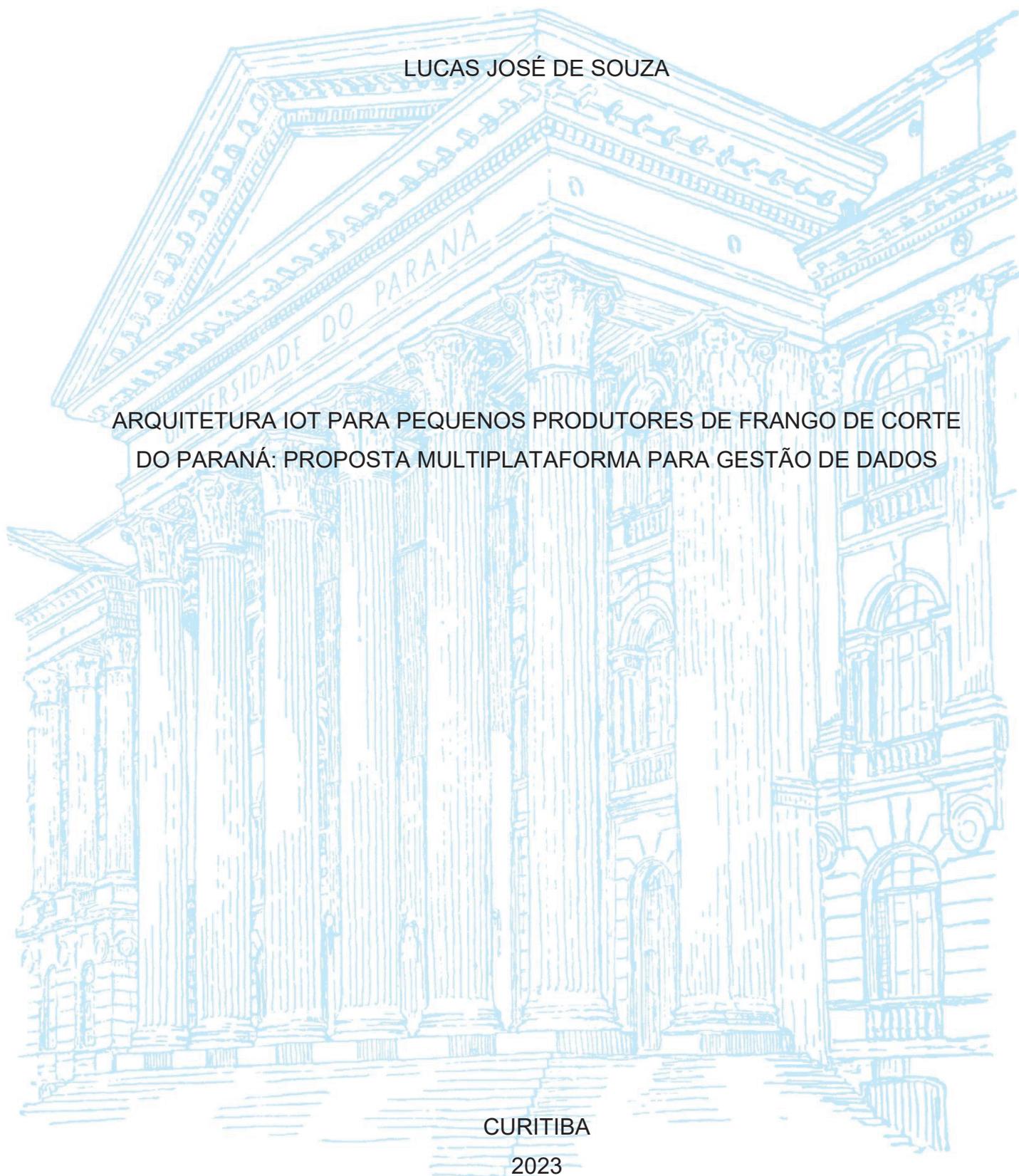
UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS JOSÉ DE SOUZA

ARQUITETURA IOT PARA PEQUENOS PRODUTORES DE FRANGO DE CORTE
DO PARANÁ: PROPOSTA MULTIPLATAFORMA PARA GESTÃO DE DADOS

CURITIBA

2023



LUCAS JOSÉ DE SOUZA

ARQUITETURA IOT PARA PEQUENOS PRODUTORES DE FRANGO DE CORTE
DO PARANÁ: PROPOSTA MULTIPLATAFORMA PARA GESTÃO DE DADOS

Tese apresentada ao Programa de Pós-Graduação em Gestão da Informação, Setor de Ciências Sociais Aplicadas, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Gestão da Informação.

Orientador: Prof. Dr. Egon Walter Wildauer
Co-orientador: Prof. Dr. André Bellin Mariano

CURITIBA

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIAS SOCIAIS APLICADAS

Souza, Lucas José de
Arquitetura IoT para pequenos produtores de frango de corte do Paraná : proposta multiplataforma para gestão de dados / Lucas José de Souza. – Curitiba, 2023.
1 recurso on-line : PDF.

Tese (Doutorado) – Universidade Federal do Paraná, Setor de Ciências Sociais Aplicadas, Programa de Pós-Graduação em Gestão da Informação.

Orientador: Prof. Dr. Egon Walter Wildauer.
Coorientador: Prof. Dr. André Bellin Mariano.

1. Internet das coisas. 2. Avicultura. 3. Frango de corte. 4. Open source. I. Wildauer, Egon Walter. II. Mariano, André Bellin. III. Universidade Federal do Paraná. Programa de Pós-Graduação em Gestão da Informação. IV. Título.

Bibliotecária: Maria Lidiane Herculano Graciosa CRB-9/2008



MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIÊNCIAS SOCIAIS E APLICADAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO GESTÃO DA
INFORMAÇÃO - 40001016058P1

ATA Nº152023

**ATA DE SESSÃO PÚBLICA DE DEFESA DE DOUTORADO PARA A OBTENÇÃO DO
GRAU DE DOUTOR EM GESTÃO DA INFORMAÇÃO**

No dia vinte e oito de abril de dois mil e vinte e três às 14:00 horas, na sala Videoconferência PPGCONT, UFPR - Setor de Ciências Sociais Aplicadas, 1º Andar Jardim Botânico, foram instaladas as atividades pertinentes ao rito de defesa de tese do doutorando **LUCAS JOSÉ DE SOUZA**, intitulada: **ARQUITETURA IOT PARA PEQUENOS PRODUTORES DE FRANGO DE CORTE DO PARANÁ: PROPOSTA MULTIPLATAFORMA PARA GESTÃO DE DADOS**, sob orientação do Prof. Dr. EGON WALTER WILDAUER. A Banca Examinadora, designada pelo Colegiado do Programa de Pós-Graduação GESTÃO DA INFORMAÇÃO da Universidade Federal do Paraná, foi constituída pelos seguintes Membros: EGON WALTER WILDAUER (UNIVERSIDADE FEDERAL DO PARANÁ), JOSE MARCELO ALMEIDA PRADO CESTARI (UNIVERSIDADE FEDERAL DO PARANÁ), MAURICIO GUY DE ANDRADE (UNIVERSIDADE FEDERAL DO PARANÁ), RAFAEL RODRIGUES DA FRANCA (UNIVERSIDADE DE BRASÍLIA), JOSÉ SIMÃO DE PAULA PINTO (UNIVERSIDADE FEDERAL DO PARANÁ). A presidência iniciou os ritos definidos pelo Colegiado do Programa e, após exarados os pareceres dos membros do comitê examinador e da respectiva contra argumentação, ocorreu a leitura do parecer final da banca examinadora, que decidiu pela APROVAÇÃO. Este resultado deverá ser homologado pelo Colegiado do programa, mediante o atendimento de todas as indicações e correções solicitadas pela banca dentro dos prazos regimentais definidos pelo programa. A outorga de título de doutor está condicionada ao atendimento de todos os requisitos e prazos determinados no regimento do Programa de Pós-Graduação. Nada mais havendo a tratar a presidência deu por encerrada a sessão, da qual eu, EGON WALTER WILDAUER, lavrei a presente ata, que vai assinada por mim e pelos demais membros da Comissão Examinadora.

CURITIBA, 28 de Abril de 2023.

Assinatura Eletrônica
02/05/2023 17:18:35.0
EGON WALTER WILDAUER
Presidente da Banca Examinadora

Assinatura Eletrônica
03/05/2023 14:19:23.0
JOSE MARCELO ALMEIDA PRADO CESTARI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
02/05/2023 16:42:43.0
MAURICIO GUY DE ANDRADE
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
02/05/2023 21:49:42.0
RAFAEL RODRIGUES DA FRANCA
Avaliador Externo (UNIVERSIDADE DE BRASÍLIA)

Assinatura Eletrônica
02/05/2023 18:35:35.0
JOSÉ SIMÃO DE PAULA PINTO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Avenida Prefeito Lothário Meissner, 632 - HORÁRIO DE ATENDIMENTO AO PÚBLICO: 8h30 às 13h e das 14h às 16h30. - CURITIBA - Paraná - Brasil
CEP 80210-170 - Tel: (41) 3360-4191 - E-mail: ppggi@ufpr.br

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.
Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 280742
Para autenticar este documento/assinatura, acesse <https://www.prppg.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp>
e insira o código 280742



MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIÊNCIAS SOCIAIS E APLICADAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO GESTÃO DA
INFORMAÇÃO - 40001016058P1

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação GESTÃO DA INFORMAÇÃO da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **LUCAS JOSÉ DE SOUZA** intitulada: **ARQUITETURA IOT PARA PEQUENOS PRODUTORES DE FRANGO DE CORTE DO PARANÁ: PROPOSTA MULTIPLATAFORMA PARA GESTÃO DE DADOS**, sob orientação do Prof. Dr. EGON WALTER WILDAUER, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 28 de Abril de 2023.

Assinatura Eletrônica
02/05/2023 17:18:35.0
EGON WALTER WILDAUER
Presidente da Banca Examinadora

Assinatura Eletrônica
03/05/2023 14:19:23.0
JOSE MARCELO ALMEIDA PRADO CESTARI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
02/05/2023 16:42:43.0
MAURICIO GUY DE ANDRADE
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
02/05/2023 21:49:42.0
RAFAEL RODRIGUES DA FRANCA
Avaliador Externo (UNIVERSIDADE DE BRASÍLIA)

Assinatura Eletrônica
02/05/2023 18:35:35.0
JOSÉ SIMÃO DE PAULA PINTO
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Avenida Prefeito Lothário Meissner, 632 - HORÁRIO DE ATENDIMENTO AO PÚBLICO: 8h30 às 13h e das 14h às 16h30. - CURITIBA - Paraná - Brasil
CEP 80210-170 - Tel: (41) 3360-4191 - E-mail: ppggi@ufpr.br

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.
Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 280742
Para autenticar este documento/assinatura, acesse <https://www.prppg.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp>
e insira o código 280742

RESUMO

A conexão de dados entre máquinas e pessoas é um dos pilares da Indústria 4.0 e da geração da Internet das Coisas (IoT- Internet of Things), favorecendo a produtividade na saúde, manufatura, logística e cidades inteligentes. No caso do agronegócio brasileiro, as projeções indicam uma menor difusão dessas tecnologias, porque ainda existe a dificuldade na adesão massiva de produtores rurais a novas tecnologias, principalmente no setor primário. Além da dependência de tecnologias estrangeiras; estas questões acabam gerando um cenário de seletividade social e tecnológica no qual por vezes, apenas grandes produtores rurais aderem às tecnologias da indústria 4.0. Outro aspecto encontrado na literatura é a ausência de identificação dos usuários com aplicativos para IoT existentes, ou seja, os usuários do contexto rural encontram dificuldade na utilização destas interfaces gráficas e acabam por abandoná-las. Neste contexto, esta tese tem por objetivo propor uma arquitetura informacional baseada em Internet das Coisas que seja open source, baixo custo e multiplataforma para pequenos produtores do setor de avicultura de corte. Para atingir os objetivos propostos foi realizado o mapeamento de necessidades, prioridades e lacunas tecnológicas com os atores da cadeia produtiva. A partir dessas informações o sistema foi modelado e construído contemplando os requisitos da camada física com o registrador de dados e a partes lógicas do back-end e front-end. A proposta também foi avaliada na perspectiva da experiência do usuário e validada estatisticamente com relação a seu impacto na melhora real do desempenho financeiro dos avicultores. Os resultados mostraram que houve um maior faturamento do produtor no lote com a arquitetura implementada comparada aos lotes anteriores à pesquisa. Outros resultados indicaram que os dados coletados pela arquitetura IoT da pesquisa foram estatisticamente significativas quando comparados aos dados de um sistema comercial implementado em um produtor de frango de médio porte. Foi possível concluir que pequenos produtores que fizerem uso desta ferramenta poderão tomar melhores decisões, identificar um baixo desempenho e agir de forma rápida, de baixo custo, simples e ergonômica para atender às exigências de competitividade internacional, quando em comparação com outros produtores da mesma região.

Palavras-chave: Aplicação IoT. Avicultura de corte. Internet das Coisas. Open Source

ABSTRACT

The data connection between machines and people is one of the pillars of Industry 4.0 and the generation of the Internet of Things (IoT), favoring productivity in healthcare, manufacturing, logistics, and smart cities. In the case of agribusiness 4.0, projections indicate a lower diffusion of these technologies because there is still difficulty in the mass adoption of new technologies by rural producers, especially in the primary sector, and a dependence on foreign technologies. These issues generate a scenario of social and technological selectivity in which only large rural producers sometimes adhere to Industry 4.0 technologies. Another aspect found in the literature is the lack of identification of users with existing IoT applications, that is, users in rural contexts find it difficult to use these graphical interfaces and end up abandoning them. In this context, this thesis aims to propose an information architecture based on the Internet of Things that is open source, low-cost, and multi-platform for small producers in the broiler poultry sector. To achieve the proposed objectives, mapping of needs and priorities was carried out with actors in the production chain, including technological gaps. With this information, the system was modeled and built, including the physical layer with the data logger and the logical part of the back-end and front-end. The proposal was also evaluated from the perspective of user experience and statistically validated with regard to its impact on the real improvement of the financial performance of poultry farmers. The results showed that there was higher revenue for the producer in the lot with the implemented architecture compared to the lots prior to the research. Other results indicated that the data collected by the research IoT architecture was statistically significant when compared to the data from a commercial system implemented in a medium-sized chicken producer. It was possible to conclude that small producers who make use of this tool will be able to make better decisions, identify low performance, and act quickly, inexpensively, simply, and ergonomically to meet the demands of international competitiveness when compared to other producers in the same region.

Keywords: Poultry Farming. Internet of Things. Open Source

LISTA DE FIGURAS

Figura 1 - Cadeia Produtiva de Frango de Corte.....	31
Figura 2 - Arquitetura IoT em quatro camadas.....	40
Figura 3 - Arquitetura do protocolo HTTP	42
Figura 4 - Arquitetura MQTT	43
Figura 5 - Modelo de captura de dados de um sensor.....	46
Figura 6 - Estrutura de arquitetura SOAP	53
Figura 7 - Estrutura da arquitetura REST	54
Figura 8 - Elementos chave e valor JSON	57
Figura 9 - Elementos de UX	64
Figura 10 - Fluxo metodológico	66
Figura 11 - Fotos do galpão da pesquisa	69
Figura 12 - Matriz de <i>Needs</i>	71
Figura 13 - Preenchimento da matriz de <i>needs</i>	72
Figura 14 - NodeMCU placa Wemos Lolin	73
Figura 15 - Sensor BME280.....	75
Figura 16 - Sensor DHT11	76
Figura 17 - Sensor MQ-2.....	76
Figura 18 - MQ-135.....	77
Figura 19 - Sensor NH3	77
Figura 20 - Sensor BH1750.....	78
Figura 21 - Sensor ST100	78
Figura 22 - Sensor NTC10K.....	79
Figura 23 - Croqui da granja com o registrador de dados	82
Figura 24 – Comparação entre os bancos de dados NOSQL	84
Figura 25 – Exemplo de conexão dos módulos relés com microcontrolador	87
Figura 26 - Diagrama de caso de uso do sistema embarcado	87
Figura 27 - Diagrama de caso de uso das aplicações web e mobile.....	89
Figura 28 - Fonte dos dados da validação	93
Figura 29 – Fluxo dos dados para a validação.....	94
Figura 30 - Sequência do protocolo de análise	97
Figura 31 - Cruzamento de dados somente do PRODUTOR 2.....	98
Figura 32 - Cruzamento de dados do PRODUTOR 2 x pequenos produtores	99

Figura 33 - Posicionamento dos elementos na matriz de needs	103
Figura 34 – Posicionamento dos elementos de demanda tecnológica.....	106
Figura 35 - Fluxo de comunicação da arquitetura IoT	107
Figura 36 – Esquema elétrico do atuador - modulo rele.....	108
Figura 37 - Integração do Data Logger com o atuador.....	109
Figura 38 – Registro das montagem do Data Logger.....	110
Figura 39 – Fluxograma 1 de extração dos dados	111
Figura 40 - Fluxograma 2 de extração de dados.....	112
Figura 41 - Exemplo de documento JSON armazenado no MongoDB	114
Figura 42 - Fluxo de comunicação dos atuadores.....	116
Figura 43 - Estrutura do servidor.....	117
Figura 44 - Paleta de cores front-end.....	118
Figura 45 - Tela de Cadastro web	119
Figura 46 - Tela de <i>Login web</i>	119
Figura 47 - Tela de inicial da aplicação web	120
Figura 48 - Tela de acompanhamento de dados sensorizados web	121
Figura 49 - Tela de manipulação dos atuadores web.....	121
Figura 50 - Tela de Geolocalização Web	122
Figura 51 - Exemplo da página Guias para o Agronegócio 4.0	123
Figura 52 - Tela de cadastro <i>mobile</i>	124
Figura 53 - Tela de login <i>mobile</i>	124
Figura 54 - Tela <i>Home mobile</i>	125
Figura 55 - Tela de indicadores <i>mobile</i>	126
Figura 56 - Tela de Sensores e Atuadores.....	126
Figura 57 - Geolocalização <i>mobile</i>	127
Figura 58 - Guias do Agro 4.0 <i>mobile</i>	127
Figura 59 - Boxplots da comparação entre sistemas	140
Figura 60 – Gráfico em linha da comparação entre os sistemas.....	140
Figura 61 – Modelo conceitual da Arquitetura IoT	144

LISTA DE GRÁFICOS

Gráfico 1 - Nota geral das avaliações da plataforma por cargo	129
Gráfico 2 - Boxplot das avaliações por cargo	130
Gráfico 3 - Fatores da ACM.....	131
Gráfico 4 - Distribuição gráfica das categorias	132
Gráfico 5 - Distribuição gráfica das categorias com menos fatores.....	133
Gráfico 6 - Boxplot de remuneração por ave.....	135
Gráfico 7 - Boxplot do IEP por lotes	136
Gráfico 8 - Boxplot da conversão alimentar por lotes.....	137
Gráfico 9 - Gráfico Q-Q remuneração por ave PRODUTOR 2.....	137
Gráfico 10 - Gráfico Q-Q remuneração por ave pequenos produtores.....	138

LISTA DE QUADROS

Quadro 1 – Síntese de propostas de Arquiteturas IoT	25
Quadro 2 - Arquiteturas IoT para o Agro 4.0	26
Quadro 3 - Valores ideias de ambiência para frangos de corte.....	35
Quadro 4 - Descrição de estudos sobre TI e GI no agronegócio	36
Quadro 5 - Classificação de Tecnologias para IoT.....	38
Quadro 6 - Características dos sensores	47
Quadro 7 - Categoria de tipos de sensores.....	48
Quadro 8 - Métodos HTTP	55
Quadro 9 - Atividades do ciclo de desenvolvimento Web/Mobile	59
Quadro 10 - Roteiro de entrevistas	70
Quadro 11 - Descrição das grandezas e unidades de medida dos sensores	79
Quadro 12 – Conjunto de materiais utilizados para montar o Data Logger	81
Quadro 13 - Tecnologias utilizadas no desenvolvimento	85
Quadro 14 - Etapas de coleta dos dados	89
Quadro 15 – Dimensões de avaliação UX da arquitetura IoT	90
Quadro 16 – Protocolo de análise da avaliação	91
Quadro 17 - Descrição da base de dados.....	94
Quadro 18 - Protocolo de análise da validação.....	96
Quadro 19 - Cruzamento de dados	97
Quadro 20 - Matriz de <i>Needs</i> para fase de alojamento.....	102
Quadro 21 - Matriz de needs para fase de crescimento e pré-abate	102
Quadro 22 - Matriz de needs para as demandas de tecnologia IoT.....	105
Quadro 23 - Rotas da API do tipo POST.....	113
Quadro 24 - Estatísticas descritivas avaliação da proposta	129

LISTA DE TABELAS

Tabela 1 - Número de aviários de corte por região	67
Tabela 2 - Estatísticas descritivas por granja	134
Tabela 3 – Estatísticas descritivas da mensuração.....	139

LISTA DE ABREVIATURAS OU SIGLAS

ABPA	- Associação Brasileira de Proteína Animal
AMQP	- <i>Advanced Message Queuing Protocol</i>
API	- <i>Application Programming Interface</i>
CNA	- Confederação Nacional do Agronegócio
CNAE	- Cadastro Nacional de Atividade Econômica
CoAP	- <i>Constrained Application Protocol</i>
CONAB	- Companhia Nacional de Abastecimento
CRUD	- <i>Create, Read, Update, Delete</i>
CSS	- <i>Cascading Style Sheets</i>
DDS	- <i>Data Distribution Service</i>
EMBRAPA	- Empresa Brasileira de Pesquisa Agropecuária
ERP	- <i>Enterprise Resource Planning</i>
FAO	- <i>Food and Agriculture Organization</i>
GPIO	- <i>General Purpose Input/Output</i>
GPS	- <i>Global Positioning System</i>
HTML	- <i>HyperText Markup Language</i>
HTTP	- <i>HyperText Transfer Protocol</i>
I2C	- <i>Inter-Integrated Circuit</i>
IBGE	- <i>Instituto Brasileiro de Geografia e Estatística</i>
IDE	- <i>Integrated Development Environment</i>
ITGU	- Índice de Temperatura do Globo negro e Umidade
IoT	- <i>Internet of Things</i>
IP	- <i>Internet Protocol</i>
JSON	- <i>JavaScript Object Notation</i>
JWT	- <i>JSON web token</i>
MAPA	- Ministério da Agricultura, Pecuária e Abastecimento
MQTT	- <i>Message Queue Telemetry Transport</i>
MVC	- <i>Model - View – Controller</i>
NoSQL	- <i>Not Only Structured Query Language</i>
ONU	- Organização das Nações Unidas
PDF	- Portable Document Format
PLC	- Programador lógico controlável

REST	- <i>Representational State Transfer</i>
RFID	- <i>Radio Frequency IDentification</i>
SAM	- <i>Self Assessment Manikin</i>
SINDIAVIPAR	- <i>Sindicato das Indústrias de Produtos Avícolas do Estado do Paraná</i>
SO	- <i>Sistema Operacional</i>
SOAP	- <i>Simple Object Access Protocol</i>
SQL	- <i>Structured Query Language</i>
TCP	- <i>Transmission Control Protocol</i>
UDP	- <i>User Datagram Protocol</i>
UEQ	- <i>User Experience Questionnaire</i>
UPnP	- <i>Universal Plug-and-Play</i>
URL	- <i>Uniform Resource Locator</i>
USB	- <i>Universal Serial Bus</i>
USDA	- <i>United States Department of Agriculture</i>
UX	- <i>User eXperience</i>
WiFi	- <i>Wireless Fidelity</i>
WWW	- <i>World Wide Web</i>
XLS	- <i>Extensible Stylesheet Language</i>
XML	- <i>Extensible Markup Language</i>
XMPP	- <i>Extensible Messaging and Presence Protocol</i>

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVOS	21
1.2 JUSTIFICATIVAS	22
1.3 ESTUDOS CORRELATOS	23
2 REFERENCIAL TEÓRICO	29
2.1 AGRONEGÓCIO 4.0	29
2.1.1 Avicultura de corte.....	30
2.1.2 Ambiência animal na avicultura de corte	34
2.2 GESTÃO DE DADOS NO AGRONEGÓCIO	36
2.3 INTERNET DAS COISAS.....	37
2.3.1 Arquiteturas IoT	39
2.3.2 Protocolos de comunicação IoT	41
2.3.3 Microcontroladores, Sensores e Atuadores.....	44
2.4 TECNOLOGIAS PARA DESENVOLVIMENTO	49
2.4.1 Banco de dados	49
2.4.2 Linguagem de programação para web, mobile e embarcados.....	50
2.5 CICLO DE DESENVOLVIMENTO DE UMA APLICAÇÃO	52
2.5.1 Desenvolvimento back-end	52
2.5.2 JavaScript Object Notation (JSON)	56
2.5.3 Desenvolvimento front-end Web e Mobile.....	57
2.5.4 Desenvolvimento de Embarcados	60
2.5.4.1 Ambientes de desenvolvimento para sistemas embarcados	61
2.5.5 Segurança de dados em aplicações Web, Mobile e Embarcados.....	62
2.6 EXPERIÊNCIA DO USUÁRIO (UX)	63
3 METODOLOGIA	66
3.1 DELIMITAÇÃO DA PESQUISA E SELEÇÃO DOS ATORES	67
3.2 PROCEDIMENTOS DE COLETA E ANÁLISE DE DADOS	69
3.3 MATERIAIS	73
3.3.1 Microcontroladores	73
3.3.2 Sensores	74
3.3.3 Demais componentes eletrônicos	80
3.3.4 Registrador de Dados (<i>Data Logger</i>)	80
3.4 TECNOLOGIAS E FERRAMENTAS	82

3.5 PROCEDIMENTOS DE DESENVOLVIMENTO	86
3.5.1 Procedimentos de desenvolvimento do sistema embarcado.....	86
3.5.2 Desenvolvimento do back-end	88
3.5.3 Procedimentos do desenvolvimento front-end Web e Mobile.....	88
3.6 PROCEDIMENTOS DE AVALIAÇÃO	89
3.7 PROCEDIMENTOS DE VALIDAÇÃO	92
4 MAPEAMENTO DAS NECESSIDADES E PRIORIDADES.....	100
5 DESENVOLVIMENTO DA ARQUITETURA IOT	107
5.1 MODELAGEM DA CAMADA FÍSICA	107
5.2 MODELAGEM DO BACK-END	110
5.2.1 Módulo de coleta, armazenamento e tratamento dos dados.....	111
5.2.2 Camada de serviços para os atuadores	115
5.2.3 Configurações do servidor.....	116
5.3 MODELAGEM DO FRONT-END.....	117
5.3.1 Front-End Web	118
5.3.2 Front-End Mobile.....	123
6 AVALIAÇÃO DA PROPOSTA.....	128
7 VALIDAÇÃO DA PROPOSTA.....	134
8 VISÃO CONCEITUAL DA ARQUITETURA IOT.....	142
9 CONSIDERAÇÕES FINAIS	147
9.1 LIMITAÇÕES.....	148
9.2 TRABALHOS FUTUROS	149
REFERÊNCIAS.....	150
APÊNDICE 1 – GUIA DE ENTREVISTA:	166
APÊNDICE 2 – CUSTO DOS COMPONENTES UTILIZADOS	167
APÊNDICE 3 – FOTOS DOS DATA LOGGER	168
APÊNDICE 4 – FORMULÁRIO DE AVALIAÇÃO DA ARQUITETURA IOT	169
APÊNDICE 5 – SCRIPT ANÁLISES AVALIAÇÃO	171
APÊNDICE 6 – DADOS TABULADOS DA AVALIAÇÃO	173
APÊNDICE 7 – BASES DE DADOS DA VALIDAÇÃO	174
APÊNDICE 8 – SCRIPT VALIDAÇÃO	176
APÊNDICE 9 – DOCUMENTAÇÃO API.....	180
APÊNDICE 10 – DIAGRAMA DE SEQUÊNCIA CADASTRO E LOGIN	196

1 INTRODUÇÃO

Com o advento da transformação digital, a forma de comunicação e interação entre pessoas e coisas estão sendo rearranjadas, essas novas configurações se aplicam também nos negócios; esse processo de digitalização é fruto de uma nova geração tecnológica nomeada de Geração 4.0.

De modo a acelerar a transformação digital e se adequar às novas tendências nos negócios, o Governo Federal do Brasil, no ano de 2019, instituiu o Plano Nacional da *Internet* das Coisas (IoT – do inglês “*Internet of Things*”) com a finalidade de acelerar a indústria 4.0 no país. Esse plano visa fomentar o desenvolvimento de soluções de base tecnológica utilizando a IoT.

No agronegócio, as projeções de inserção das tecnologias 4.0 em cenário nacional são crescentes e estão alcançando patamares até maiores do que outras categorias de indústrias como as da construção civil, logística, têxtil, papel e celulose, entre outras (OLIVEIRA et al, 2022). Embora essas projeções sejam ligeiramente maiores do que em outros setores, os impactos sociais, econômicos e ambientais são considerados relevantes, como no exemplo da agricultura com o uso mais sustentável de fertilizantes, pesticidas, economia de água, energia e demais insumos para as lavouras (LANGBECKER; BEURON, 2020).

Em outros setores, como avicultura, suinocultura, bovinocultura e etc; as melhorias se dão principalmente no bem-estar animal e no controle sanitário; na pecuária podem ser listados os controles de produção de leite, controle de infecções, entre outros. Em outras palavras, independente do setor agro, a modernização e expansão dessas tecnologias só trazem benefícios para o setor. Não somente na perspectiva produtiva, mas também na qualidade de vida do produtor, inclusive com a possibilidade de redução de carga de trabalho (ZANIN, 2022; MENDES, GERÓLAMO; ZUIN, 2020).

Mesmo com um cenário de crescimento, o agronegócio brasileiro tem dois desafios principais para os próximos anos, o primeiro é a inclusão em massa de tecnologias 4.0 nas pequenas propriedades rurais, que até o ano de 2021, somaram aproximadamente 4 milhões de propriedades (BRASIL, 2019; CONTE; ARAUJO, 2020; BRASIL, 2021). Segundo Bolfe et. al. (2020), o segundo desafio é a maturidade no domínio e uso de novas tecnologias no campo e a dependência de tecnologia estrangeira.

Zhai et. al. (2020) apontam que as implementações de sistemas da geração 4.0 estão acontecendo, na maioria das vezes, em grandes propriedades rurais, que possuem maiores condições de realizar investimentos, têm acesso a melhores condições de financiamentos, e possuem uma maior capacidade de adaptação. Enquanto médios e pequenos produtores, por vezes, dependem de *cooperativas* para terem acesso a sistemas de gestão ou demais soluções tecnológicas.

Se por um lado, essa transformação digital é importante para o desempenho geral da cadeia produtiva do agronegócio como um todo, por outro, a mesma, vem conduzindo a um processo de seletividade social (MAISTRO; MONTEBELLO; SANTOS, 2019); a crescente complexidade da gestão rural, atrelada à dificuldade que a maioria dos pequenos produtores têm de se apropriar de conhecimento tecnológico adequado, e a condição dos mesmos de implementar uma arquitetura digital em suas propriedades está fazendo com que parte dos pequenos produtores rurais desistam de permanecer em suas atividades tradicionais de produção e migrem para a produção de commodities voltadas ao mercado externo; ou até mesmo, em casos extremos, deixem suas atividades, gerando conseqüentemente um enfraquecimento nas ofertas por alimentos do mercado nacional (ZOCCA et al; 2019; HARBELI; OLIVEIRA; YANAZE, 2019).

Outros dois tópicos relevantes na questão da modernização das pequenas propriedades rurais e que segundo Ramos (2005), Mishra e El-osta (2008), Burton e Fischer (2015), Suess-Reyes e Fuetsch (2016), Rapela (2019) entre outros são as questões hereditárias e de sucessão familiar. A questão de herança é citada por autores como Ramos (2005), Hansen, Jensen e Skovsgaard (2015), Huning e Wahl (2021) é que uma vez que os produtores ao ficarem incapacitados de cuidar das fazendas deixam a mesma sob o controle de um ou nenhum filho levando a venda da propriedade rural, ou seja, há uma descontinuação do trabalho. Inclusive isso tem agravado o problema agrário de concentração de terras na mão de poucos fazendeiros (HUNING; WAHL ,2021).

Especificamente sobre o tema da sucessão familiar, os estudos de Mishra e El-osta (2008) citam o viés econômico, no qual ressaltam a importância do financiamento adequado para que os produtores tenham fôlego para manterem seus negócios e investirem em inovações. Além de que permite que os agricultores que entram no ramo superem mais facilmente os endividamentos. Os autores ainda ressaltam que se o proprietário desejar continuar a operação da fazenda após sua

aposentadoria, um planejamento deve ser realizado para mitigar riscos de insucesso de uma operação sob nova administração por parte do herdeiro.

Um outro viés, nessa perspectiva de sucessão familiar, aparece nos estudos de Burton e Fischer (2015) os quais levantam os fatores de taxa de fracassos sendo relevantes a falta de conhecimento, interesse e envolvimento suficiente dos sucessores.

Nas pesquisas de Ramos (2005), Burton e Fischer (2015), Suess-Reyes e Fuetsch (2016) são citados os incentivos, políticas voltadas para a preservação das atividades rurais na Europa principalmente os pequenos produtores, nos referidos estudos existe um viés tecnológico e inovador, pois a agricultura familiar está exposta a severas mudanças econômicas, políticas, sociais e ecológicas. Para possibilitar a sucessão e salvaguardar a sobrevivência da fazenda no longo prazo, as famílias agricultoras são cada vez mais forçadas a buscar estratégias inovadoras, sustentáveis e voltadas para o mercado (SUESS-REYES; FUETSCH, 2016).

A criação e desenvolvimento das políticas agrícolas comuns na União Europeia visam fomentar e dinamizar o setor agrícola europeu, tendo assumido como principais pressupostos o fomento e apoio aos agricultores em promover uma melhor qualidade da produção, aumento de produtividade no setor. Além disso, essas políticas visam garantir a estabilidade dos preços e segurança de estoque de alimentos. Consequentemente dignificar a qualidade de vida dos agricultores europeus e promover a economia rural do emprego na agricultura, nas indústrias agroalimentares e nos setores conexos (CARVALHO, 2016; COMISSÃO EUROPEIA, 2020).

Enquanto que nos países desenvolvidos ocorrem pactos, e ações derivadas de políticas públicas direcionadas ao setor, em países em desenvolvimento, especificamente no Brasil, ainda há a dependência de tecnologias estrangeiras seja no hardware ou software, a migração para a produção somente de commodities voltadas para a exportação e os problemas de sucessão familiar, refletem até mesmo sobre a avicultura de corte.

Segundo a Associação Brasileira de Proteína Animal (ABPA) é terceiro maior produtor mundial (aproximadamente 10% da produção mundial em 2021), ficando atrás somente da China e Estados Unidos que lideram a produção (VANNIER ET AL; 2022).

Estima-se que menos de 1% dos 180 mil produtores de frango de corte brasileiros possuam algum tipo de automação e controle em tempo real em seus aviários (IBGE, 2020). O Paraná, maior produtor de frango do Brasil (mais de um terço de toda a produção nacional), possui cerca de 921 aviários (6% do total de aviários de corte) com algum tipo de automação (SINDIAVIPAR; 2021).

Dando indícios que mesmo com os avanços tecnológicos o setor primário brasileiro ainda está atrasado na inserção de tecnologia nas atividades corriqueiras. Contudo, com o aperfeiçoamento das redes de *internet*, em especial com a chegada do 5G, que visa atender a uma grande quantidade de dispositivos conectados com um maior alcance e velocidade no fluxo de dados, atrelada ao plano de desenvolvimento nacional da IoT, há uma tendência da inclusão gradual de pequenos produtores rurais na geração 4.0.

O conceito que interliga agronegócio ao paradigma IoT pode ser chamado de Agricultura Inteligente (*Smart Farming*) ou Agricultura Digital; e permite o monitoramento e controle de processos da cadeia produtiva do agronegócio em tempo real utilizando a *internet*, tornando o controle da produção cada vez mais mensurável, uma vez que é possível utilizar dados que antes não eram possíveis de se coletar e quantificar (GRIMBLATT, 2020; KUMARI; KUMAR; SINGHAL, 2020; ROCHA; OLIVEIRA; TALAMINI, 2021).

As tecnologias utilizadas na agricultura inteligente incluem dispositivos de *hardwares* de baixo custo, sensores e atuadores que quando instalados em locais específicos monitoram e atuam em conformidade com os objetivos definidos pela regra do negócio, apoiando suas tomadas de decisão e dando maior eficiência na gestão (OLIVEIRA; MOREIRA; SILVA, 2021).

No entanto, ainda há barreiras na implementação dos sistemas IoT voltados para as atividades rurais, seja pelos custos de implementação e manutenção, seja pela falta de infraestrutura local como, por exemplo, o acesso à internet, resistência por parte dos produtores e até dificuldades na obtenção de linhas de crédito destinadas para esses projetos.

Outro obstáculo, que tangencia a adesão de soluções tecnológicas, é a ausência de identificação do usuário com os aplicativos existentes, ou seja, há um paradoxo que “soluções não solucionam” efetivamente um determinado problema, ou atendem a uma necessidade (HUSSAIN ET AL., 2018; DHARMAYANTI; BACHTIAR; WIBAWA, 2018; ALFARIDZI; YULIANTI, 2020). O que acontece é uma grande

quantidade de aplicações com recursos não úteis ou não intuitivos, que levam ao desestímulo (ALFARIDZI; YULIANTI, 2020). Nesse contexto, elementos de experiência do usuário tornaram-se relevantes para o desenvolvimento de aplicações, entendendo seu público e suas demandas, não somente atendendo às regras do negócio (JOHNSON, 2020; RICHARDSON; CAMPBELL-YEO; SMIT, 2021).

Por conseguinte, concatenando as problemáticas de democratização tecnológica no campo, barreiras de infraestrutura, experiência do usuário e iniciativas para fortalecimento do desenvolvimento de tecnologias nacionais voltadas a indústria 4.0 surge a questão norteadora da tese: **“Como propor uma arquitetura IoT integrada e multiplataforma de gestão de dados para pequenos produtores da cadeia produtiva do frango de corte, considerando tecnologias que utilizem softwares e hardwares livres, baixo custo e atendendo a requisitos de usabilidade e experiência de usuário?”**.

1.1 OBJETIVOS

Esta tese tem por objetivo geral propor uma arquitetura de gestão de dados IoT *open source*, de baixo custo, multiplataforma para pequenos produtores da cadeia produtiva de frango de corte.

Para o atingimento do objetivo geral, foram definidos cinco objetivos específicos:

- a) Mapear as necessidades, prioridades e lacunas tecnológicas para o contexto da cadeia produtiva de pequenos produtores frango de corte do estado do Paraná;
- b) Estruturar os módulos de captura, armazenamento e integração de dados IoT (camada física);
- c) Desenvolver o *back-end* e *front-end* contemplando a versão web e mobile (Android e iOS);
- d) Avaliar a proposta com pequenos produtores e outros atores da cadeia produtiva de frango de corte;
- e) Validar estatisticamente a eficiência da arquitetura comparada a um sistema comercial.

1.2 JUSTIFICATIVAS

A tese almeja contribuir com uma alternativa viável sob o ponto de vista socioeconômico e tecnológico que proporcione a inclusão e democratização das tecnologias IoT para os produtores de frango de corte de menor renda.

No tocante à perspectiva socioeconômica, a Companhia Nacional de Abastecimento (CONAB), em 2021, mapeou que os pequenos agricultores correspondem a cerca de 80% da ocupação no setor rural, e são responsáveis por aproximadamente 45% da produção agropecuária. Evidenciando a massiva presença dos pequenos produtores rurais no país, a justificativa se molda no sentido que muitos dos pequenos produtores possuem baixa escolaridade e um menor contato com tecnologias, por isso devem ser incluídos nas políticas de desenvolvimento tecnológico rural de modo a aprimorar cada vez mais seus processos produtivos (CRUZ, 2021).

A ênfase tecnológica da justificativa se alinha com as políticas públicas voltadas à transformação digital no Brasil, especialmente com o decreto nº 9.854 de 2019 que instituiu o Plano Nacional de *Internet* das Coisas, o qual, tem por objetivo melhorar a qualidade de vida das pessoas e promover ganhos de eficiência nos processos produtivos e serviços por meio da implementação de soluções de IoT.

Além dele, em 2022, foi criado também o Plano Nacional de Agricultura e Pecuária de Precisão que visa incentivar a adoção de ferramentas tecnológicas, satélite, internet e dados em tempo real para melhor aproveitamento de solo e dos rebanhos. A lei apoia a ampliação da eficiência na aplicação de recursos e insumos de produção, de forma a diminuir o desperdício, reduzir os custos de produção, aumentar a produtividade e a lucratividade, bem como garantir a sustentabilidade ambiental, social e econômica.

Sendo assim, a proposta encontra seus alicerces nessas políticas públicas, uma vez que promove o desenvolvimento de uma nova arquitetura de comunicação entre o *software* e os dispositivos IoT (sensores e atuadores), que utilizando *softwares* livres e *hardwares* de baixo custo pode oferecer um sistema robusto de gestão de dados com acompanhamento em tempo real de sua produção, além de consultar dados externos de órgãos governamentais ligados ao agronegócio como Embrapa, Conab e IBGE.

A tese contribui com a concepção de um mundo baseado essencialmente em informação, que requer novos modelos de serviços e interfaces que garantam a conectividade, colaboração, e a solução dos mais diversos problemas decorrentes das questões inerentes à IoT aplicadas ao agronegócio, principalmente aos pequenos produtores rurais. Os impactos dessa pesquisa também incluem o aumento da produtividade e a redução do uso de recursos como água e energia, bem como a qualidade de vida do produtor.

Segundo Yadav et al. (2020) a mão-de-obra escassa e baixa tecnologia nas produções dificultam o controle da ambiência animal, principalmente nas questões meteorológicas locais, uma vez que as estações ficam espalhadas longe uma das outras, e não conseguem aferir precisamente em regiões específicas, principalmente longe dos maiores centros urbanos. Ademais, podem haver mudanças repentinas durante a semana ou mesmo durante um dia, assim sendo a arquitetura IoT dessa pesquisa também contribui com a expansão das áreas sensoriadas com dados de temperatura e pressão as quais podem ser compartilhadas com produtores vizinhos, criando uma rede de dados para uso coletivo (YADAV ET AL. ,2020).

Também se evidenciou nas pesquisas de Phasinam et al (2022) e Rehman et al. (2022) que a automação e digitalização no campo contribuem com a eliminação de atividades redundantes e desgastantes no âmbito físico e psíquico, além de gerar uma demanda maior por mão de obra especializada, valorizando o trabalho humano e aumentando a rentabilidade do negócio.

Devido às mudanças do comportamento humano frente às Tecnologias de Informação, observa-se que a utilização de diferentes dispositivos para acesso à *Internet* pode apresentar desvios de padrão de acordo com a formação cultural, intelectual, situação econômica e idade dos indivíduos, influenciando, assim, a forma de utilização dos ambientes digitais, nesse sentido a tese também incide na atenção do ciclo da informação no contexto da experiência do usuário, para que o mesmo consiga acessar, recuperar, analisar e tomar decisões a partir de um sistema com interface amigável, intuitiva e robusta (ULFADA; NURFIANA; HANDAYANI; 2022). Com isso, a tese mostra a integração das qualidades objetivas e subjetivas da Experiência do Usuário em ambientes informacionais (aplicações *web* e *mobile*).

1.3 ESTUDOS CORRELATOS

Para o desenvolvimento desta subseção foi realizada uma revisão exploratória da literatura de modo a encontrar lacunas teóricas e práticas em que a tese possa contribuir.

Foi adotado o método proposto por Toschi, Campos e Cugnasca (2017), que consiste em quatro etapas: (i) Definição da questão de pesquisa, (ii) Definição dos critérios de inclusão e exclusão, (iii) Busca dos trabalhos e (iv) Seleção de trabalhos correlatos. A temática pesquisada englobou a definição da questão de pesquisa e os critérios para inclusão na revisão exploratória eram trabalhos com propostas de arquiteturas IoT em diversos contextos (arquitetura e construção civil, saúde, área militar, cidades inteligentes, logística) e em pesquisas que propõem arquiteturas com aplicação específica no agronegócio, seja para pequenos, médios e grandes produtores rurais.

Para encontrar os trabalhos correlatos foram utilizadas cinco bases de dados: *Google Scholar*, *Scopus*, *Web of Science*, *Science Direct* e *Spell* com vistas a publicações indexadas com os termos derivados do título da tese: “Arquitetura IoT”, “Sistema multiplataforma para gestão de dados IoT” e “Sistemas IoT” e seus termos correspondentes em inglês.

Na literatura existem diversas propostas de arquiteturas IoT, sejam focadas na camada de serviços, apenas na coleta e armazenamento dos dados, outras, mais completas, com o desenvolvimento da camada de interação com o usuário (*front-end*).

No tocante a trabalhos somente com a camada de serviços está a proposta de Mehmood; Culmone; Mostarda (2017) o qual apresenta uma arquitetura IoT utilizando bancos de dados não relacional - *NoSQL*. Os autores propuseram um esquema de captura e armazenamento de dados sensorizados de temperatura e frequência cardíaca, *timestamp* e id de identificação do paciente; inicialmente criaram um servidor local com um banco de dados *MongoDB* responsável por armazenar os dados advindos dos sensores e, em um segundo momento, escreveram consultas ao banco retornando os valores coletados persistidos utilizando uma ferramenta de modelagem de dados para o *mongoDB*. Outra pesquisa nessa linha foi publicada por TUN, Z. et al. (2020), que propôs um esquema de banco de dados *MongoDB* para armazenar dados de temperatura e umidade utilizando um serviço em nuvem (*MongoDB cloud*) do próprio desenvolvedor; além do banco de dados estar conectado a *web* ao invés de um servidor local, é possível a consulta dos dados por meio de um gráfico oferecido dentro da plataforma. Ambos trabalhos propuseram o

armazenamento e consulta de dados advindos de sensores IoT sem utilizar uma camada de serviços, ou seja, todas as operações eram realizadas diretamente ao banco de dados.

Com propostas mais robustas, incluindo protocolos de comunicação via *internet* para IoT (*HTTP/MQTT/CoAP*), com uma camada de serviços para interação com banco de dados, e viabilizando escalabilidade e interoperabilidade as pesquisas de Cruz et al. (2019), Garg e Dave (2019) têm em comum o uso de bancos de dados não relacionais *NoSQL* e o desenvolvimento de uma API REST que utiliza protocolo de comunicação *HTTP* para manipulação de dados através de arquivos *JSON*. Além disso, na pesquisa de Cruz et al. (2019) os autores criam um *framework* de conversão de protocolo *MQTT* para *HTTP* facilitando a comunicação entre sistemas *web*.

Nas propostas incluindo a camada de interação com o usuário (*front-end*), Amali, Guru e Sridevi (2020) apresentam uma solução IoT para lidar com dados de informações de tráfego veicular e quase acidentes, como a velocidade, posição, direção e destino de veículos. Nesse trabalho também há o uso de um modelo híbrido onde informações sociais são armazenadas utilizando o modelo relacional e as informações referentes ao tráfego utilizam o modelo não relacional. Nesse modelo, os autores propõem uso de uma estrutura em nuvem contendo o banco de dados não relacional, uma camada de serviço com requisições *HTTP* e uma aplicação *web* para a gestão e acompanhamento em tempo real do posicionamento veicular.

O Quadro 1 sintetiza, em ordem cronológica, pesquisas que apresentam arquiteturas IoT aplicadas a diversos contextos de negócios apresentando ou não uma camada de interação com o usuário.

Quadro 1 – Síntese de propostas de Arquiteturas IoT

Área de aplicação	Objetivo da pesquisa	Arquitetura proposta	Autor(es)
Cidades inteligentes	Apresentar um protótipo de aplicativo <i>mobile</i> que monitoraria a disponibilidade de vagas de estacionamento e automatiza o sistema de bilhetagem eletrônica.	Usando <i>leds</i> para a identificação visual das vagas os motoristas estacionam o carro em uma vaga e por meio de um aplicativo <i>android</i> podem fazer o check-in usando o GPS e ao sair da vaga sensores fazem automaticamente o <i>check-out</i> .	TJB; SUBRAMANI; SOLANKI (2017)
Segurança Militar em cidades inteligentes	Explorar o potencial e capacidades de comunicação de sensores IoT em cidades inteligentes para operações militares	Usando o protocolo de comunicação MQTT e HTTP sensores são alocados para monitorar o fluxo de pessoas, captura de ruídos e sensores de gases integrados a um serviço de geolocalização em tempo real	JOHNSEN et al. (2018)

Logística	Propor uma arquitetura IoT para a cadeia de suprimentos na logística que consiga compartilhar informações de um único fornecedor para vários clientes monitorando cada etapa das entregas.	Utilizando RFID e GPS para monitorar e acompanhar as etapas dos processos logísticos. Por meio de um serviço <i>web</i> os dados eram coletados e enviados de forma automática para o e-mail dos clientes.	TU; LIM; YANG (2018)
Construção Civil	Propor um sistema autônomo que monitora e alerta trabalhadores da construção civil em zonas de riscos.	Com dispositivos utilizando frequência rádio os dados são integrados a um servidor <i>back-end</i> que monitora em tempo real as condições do local e dispara alertas sonoros e visuais em condições de risco	KANAN; ELHASSAN; BENSALAM (2018)
Construção Civil	A pesquisa constrói um sistema de conforto térmico que por meio de indicadores de dados do ambiente processados no MATLAB são acionados atuadores para regular temperatura, umidade e luminosidade do ambiente.	Utilizando um microcontrolador Arduino para a coleta dos dados e um módulo ESP-01 para envio via para um banco de dados onde MATLAB consultava e realizava as análises com os parâmetros definidos nos indicadores e retornava os comandos aos atuadores	SUNG; HSIAO; SHIH (2019)
Saúde	Revisão da literatura sobre o Sars-cov 19 (Covid-19) e técnicas de monitoramento para sugestão de uma arquitetura baseada em IoT, que pode ser usada para minimizar a disseminação do vírus.	Propõe uma arquitetura IoT utilizando microcontrolador NodeMCU e sensor de temperatura corporal para mapear os locais de maior risco de transmissão do vírus. Usando o protocolo HTTP e um aplicativo <i>mobile</i> desenvolvido com <i>App-Inventor</i> é possível consultar geograficamente as zonas críticas.	KUMAR; KUMAR; SHAH (2020)
Saúde	Propor uma arquitetura em quatro camadas baseadas em <i>blockchain</i> para o gerenciamento seguro de dados originados de dispositivos médicos IoT.	A arquitetura proposta tem a camada de <i>blockchain</i> que fornece funcionalidade para implementar e hospedar o banco de dados de <i>blockchain</i> ; uma camada de dispositivo IoT que consiste em dispositivos eletrônicos de dados, como vigilância de saúde, balança de pesagem e sensores de medição de pressão arterial; a camada de aplicativo desenvolvido para sistema <i>Android</i> que exibe os dados coletados e, por fim, uma camada adaptadora que integra os dados IoT com a camada de <i>blockchain</i> .	PAWAR et al. (2021)

Fonte: o autor (2023)

No

Quadro 2, são apresentadas arquiteturas IoT específicas para o contexto do Agro 4.0 de pesquisas realizadas no recorte temporal de 2019 até 2023.

Quadro 2 - Arquiteturas IoT para o Agro 4.0

Sub área do Agro	Objetivo da pesquisa	Arquitetura proposta	Autor(es)
------------------	----------------------	----------------------	-----------

Fazendas inteligentes	Propor uma arquitetura genérica de referência para uso em fazendas inteligentes	Uma arquitetura composta pelos módulos de captura e transmissão dos dados sensorizados, módulo de gerenciamento <i>web</i> e do aplicativo <i>Android</i> . Desenvolvido com o protocolo MQTT para comunicação entre microcontroladores e servidor, além de usar uma camada de serviço (<i>back-end</i>) de computação em nuvem.	TRIANTAFYLLOU; SARIGIANNIDIS; BIBI (2019)
Viticultura 4.0	Propor um sistema <i>web</i> para acompanhamento de dados meteorológicos da produção de uvas.	Utilizando o protocolo de comunicação HTTP os dados coletados em formato JSON são enviados a um servidor local e persistidos em um banco de dados não relacional. A camada <i>web</i> é provida por um serviço privado de visualização da informação que consulta os dados e disponibiliza em formato de gráficos e relatórios analíticos.	VOUTOS ET AL. (2019)
Agri 4.0	Projeto de um sistema de gerenciamento de Agricultura Inteligente usando uma arquitetura IOT de baixo custo que ajude os agricultores a aumentar a produtividade.	Usando uma rede <i>wi-fi</i> de longo alcance, os autores propõem uma arquitetura de coleta e transmissão de dados para um servidor em nuvem usando o NB-IoT (banda estreita e baixo consumo energético) para se comunicar com o servidor. O serviço de computação disponibiliza o acompanhamento temporal de temperatura, umidade e pH do solo que pode ser consultado através de um endereço <i>web</i> pelos agricultores.	VALLURU; KOTIKAM; HARIBABU (2021)
Agri 4.0	Apresentar um sistema de monitoramento de temperatura e umidade para o cultivo mais eficiente de folha de mostarda	O sistema fica alocado em um serviço privado de computação em nuvem que incluía os módulos de armazenamento, tratamento e disponibilização dos dados sensorizados em um endereço <i>web</i> no formato de <i>dashboard</i> .	ABIOYE ET AL. (2020)
Agri 4.0	Propor um sistema preditivo de geadas em estufas de produção de vegetais	O sistema monitora dados meteorológicos e por meio de um microcontrolador envia os dados, usando o protocolo HTTP, a um servidor <i>web</i> de terceiros que é integrado a uma inteligência artificial, e a mesma indica os riscos de gear. Foi construído um <i>dashboard</i> para o acompanhamento local dos dados coletados	CASTAÑEDA; CASTAÑO, (2020)
Pecuária 4.0	Este artigo valida a implementação da arquitetura IoT-A em cinco culturas agro distintas.	A estrutura compreende um modelo IoT-A. O modelo arquitetônico contempla a camada de serviços com uma API para coleta, armazenamento e tratamento dos dados. A camada <i>front-end</i> é uma aplicação <i>web</i> e é disponibilizada por uma empresa privada que processa os dados e entrega mapas georreferenciados com imagens coletadas por <i>drones</i> .	VERDUOW ET AL. (2021)
Modelos estatísticos para Avicultura	Revisar os modelos matemáticos/estatísticos para monitoramento animal (ambiente animal e saúde).	Nas pesquisas foram elencados modelos matemáticos para a previsão de volume de silos de ração, o comportamento dos custos de produção baseado em índices de ambiência animal, mortalidade utilizando regressão múltipla, e algoritmos de inteligência artificial para prever índices sensorizados que afetam o bem estar animal.	OLIVEIRA, L; TONIN, P. C; VICENZI (2020); KUHN ET AL. (2022); OJO ET AL. (2022); ARSHAD ET AL. (2022)

Fonte: o autor (2023)

Independente da área de aplicação em que se concentra a proposta não há um consenso sobre as tecnologias de *software* e *hardware* utilizadas e nem quais camadas o sistema deve incluir; além disso algumas propostas dos Quadros 1 e 2 fazem uso de ferramentas de terceiros, seja para o armazenamento, tratamento ou visualização dos dados.

Desse modo, a tese tem como característica de ineditismo a apresentação de uma arquitetura IoT desenvolvida de ponta a ponta com tecnologias gratuitas e/ou livres (*hardware* e *software*) para a gestão de dados direcionada para pequenos produtores de frango de corte; englobando módulos de comunicação que garantam a interoperabilidade entre o sistema (*web*, *mobile* e embarcados) e a segurança do sistema com *tokens* de autenticação para acesso aos dados, além da camada analítica de visualização de dados. A arquitetura IoT também inclui o atendimento a requisitos de experiência do usuário.

Por fim, a arquitetura da tese incluirá a criação de um algoritmo preditivo de tomada de decisão para os dados sensorizados relacionado com a ambiência e bem estar animal, por meio de análises comparativas com outros algoritmos encontrados na literatura dos autores Oliveira, Tonin e Vicenzi (2020), Kuhn et al. (2022); Ojo et al. (2022) e Arshad et al. (2022), conforme Quadro 2.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conteúdos que fundamentam este trabalho, iniciando com definições fundamentais de Agronegócio 4.0, Gestão de dados no Agronegócio, seguido para as seções de *Internet* das coisas (IoT), Tecnologias para desenvolvimento de *software*, o ciclo de desenvolvimento com design patterns, construção da *web*, *mobile* e de embarcados e por fim, modelos de algoritmos para a análise de dados.

2.1 AGRONEGÓCIO 4.0

A evolução da agricultura e da pecuária estão intimamente ligadas com as revoluções industriais que aconteceram no mundo, em um contexto histórico anteriormente a primeira revolução industrial todo o trabalho rural contava com o apoio de animais para a força e tração do arado no campo. E, por volta do século XVIII, com a primeira revolução industrial em curso, foi possível dar mais agilidade a algumas tarefas braçais e artesanais com as máquinas a vapor. (CANUTO, 2012; OLIVEIRA ET AL; 2022)

Mas somente no século XIX, com a segunda revolução industrial foi possível mecanizar as atividades agrícolas trocando máquinas de tração animal por motores a combustão, e junto disso avançaram também as biotecnologias de sementes, fertilizantes e defensivos agrícolas mais avançados e seus métodos de pulverização (CONWAY; GRIFFON; SABOURIN, 1998; SANTOS, 2012).

No século XX aconteceu a terceira revolução industrial acompanhada das tecnologias da informação e comunicação, com *softwares* capazes de otimizar a gestão do campo e permitir a mensuração mais apurada de produtividade além de automatizar processos (GEBBERS; ADAMCHUK, 2010), a chamada agricultura de precisão.

Além dela, a agricultura digital (Agricultura 4.0) e se caracteriza principalmente pelo uso de máquinas e tecnologias. Essa nova agricultura, até então, foi caracterizada por disponibilizar uma série de recursos que envolviam biotecnologia, adubação corretiva, defensivos agrícolas, técnicas de manejo, georreferenciamento (GPS), máquinas agrícolas cada vez mais modernas e automatizadas (ARTUZO; SOARES; WEISS, 2017). Em síntese, é possível que o agricultor controle a lavoura

por meio de informações e trate cada área como única, reconhecendo as diferenças de cada região.

Com a quarta revolução industrial, Medeiros (2019) caracteriza a hiperconectividade de todos os envolvidos na cadeia produtiva do agronegócio em tempo real como o Agronegócio 4.0; esse novo cenário possibilita a presença no campo sem a necessidade de um trabalhador *in loco*, graças à automação inteligente de coleta e monitoramento dos dados rurais. De forma pragmática, “a agricultura 4.0 é um conjunto de inovações voltadas para tecnologia avançada, a qual visa aprimorar, otimizar e rentabilizar a produtividade no campo” (SANTOS; ESPERIDIÃO; AMARANTE, 2019 p.21).

E a demanda por otimizações no uso dos recursos naturais e insumos tem feito que fazendas estejam migrando da agricultura 3.0 para a agricultura 4.0, ou seja, estão sendo massivamente monitoradas e automatizadas. Uma das primeiras adaptações é a disseminação de sensores dispersos por toda a propriedade e interligados à *Internet* gerando dados em grandes volumes que necessitam serem armazenados, tratados, analisados e disponibilizados para as tomadas de decisão no meio rural (SILVA; SILVA-MANN, 2020).

2.1.1 Avicultura de corte

As atividades relacionadas ao processo de produção e comercialização de produtos agroindustriais são nomeadas de acordo com a perspectiva utilizada. O termo cadeia produtiva ou cadeia de produção agroindustrial define as atividades a partir de um determinado produto final (BATALHA, 2021). Dentre as diversas atividades da produção avícola, o destaque é a engorda de aves para o abate, em especial o frango de corte, então, em virtude disso, esta pesquisa contempla a cadeia produtiva de frango de corte que está inserida na categoria da avicultura, a qual também inclui a produção de adubo, ovos e outros galináceos (BATALHA, 2021).

O apanhado histórico da cadeia produtiva do frango de corte tem início em meados da década de 1950, onde antes a criação de aves era basicamente uma atividade de subsistência. Os criadores de frango e agroindústrias iniciaram o processo de integração, que buscava aperfeiçoar o processo desde a criação até o abate (BATALHA, 2021). Com isso, as agroindústrias poderiam programar sua

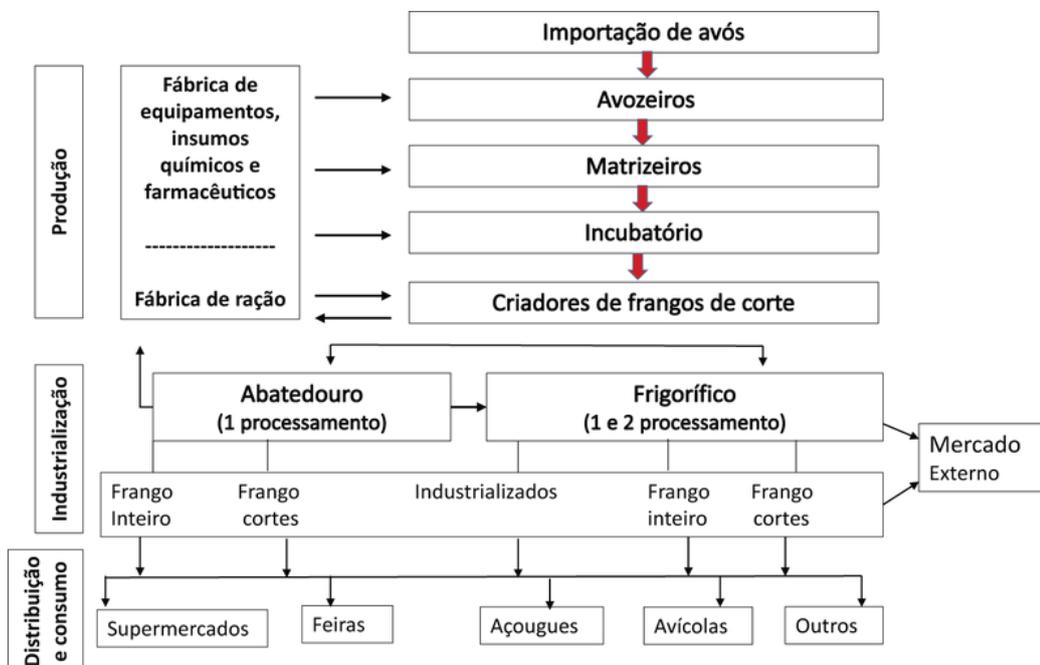
produção anual, obter maior aproveitamento das instalações e redução de custos (CARVALHO JÚNIOR; GIAROLA, 2020; PROCÓPIO ET. AL; 2020).

Posteriormente, na década de 1970 o setor agro passou por um processo de modernização que induziu mudanças na base técnica da produção e nos processos produtivos. Nesse período, as agroindústrias avícolas se destacavam como um segmento moderno que investia na atividade e firmava parcerias com empresas estrangeiras de melhoramento genético (BATALHA, 2021).

E foi a partir da década de 1990 que a cadeia produtiva de frango de corte ganhou maior visibilidade no cenário do agronegócio brasileiro e mundial, graças a uma diversidade de recursos tecnológicos e a eficácia no seu gerenciamento (PROCÓPIO ET. AL; 2020). Nos anos 2000 foi evidenciado um forte crescimento na produção brasileira fruto da melhoria dos processos e da qualidade sanitária dos rebanhos que propiciaram a conquista de mercados externos e também pela expressiva melhoria da renda da população brasileira que trouxe um significativo crescimento do consumo interno (PROCÓPIO ET. AL; 2020).

Batalha (2021) define a cadeia produtiva conforme apresentado na Figura 1.

Figura 1 - Cadeia Produtiva de Frango de Corte



Fonte: Batalha (2021, p.45)

As principais ligações da cadeia estão relacionadas a alguns termos técnicos utilizados nas fases da produção de frangos de corte como o avozeiro, matrizeiro,

incubatório, aviário, frigorífico, varejo e consumidor final e aos elos auxiliares que tem como objetivo prover insumos e outras demandas à cadeia principal como: a pesquisa e desenvolvimento genético, medicamentos, milho, soja e outras matérias-primas (BATALHA, 2021).

Segundo Batalha (2021), os principais elos da cadeia produtiva da avicultura de corte são:

- **Avozeiro:** São realizados por granjas que, a partir da obtenção de ovos das linhagens (bisavós), produzem as aves avós que passam pelo processo de cruzamento para a geração de matrizes.
- **Matrizeiro:** Espaços na granja onde as matrizes são cruzadas para gerarem os ovos que serão enviados aos incubatórios.
- **Incubatório/Nascedouro:** Nos incubatórios são chocados os ovos, dando origem às aves recém-nascidas que serão levadas aos aviários.
- **Aviário:** Local onde é realizado o processo de crescimento e engorda para a produção dos frangos; as aves chegam nos aviários com até três dias e ficam até a época de abate, que acontece dentro de 38 a 45 dias de engorda. O aviário é o quarto elo da cadeia produtiva e corresponde a uma etapa de produção, caracterizada pelos contratos de integração entre frigoríficos e produtores rurais (integrados). É no aviário que se dá o crescimento e a engorda das aves, que ali chegam com algumas horas depois de nascidos e ficam até a época de abate, aos 43 dias, aproximadamente.
- **Frigorífico:** Depois da engorda, os frangos são encaminhados para os frigoríficos e abatedouros, onde são abatidos e encaminhados para comercialização, seja como frangos inteiros, seja como cortes (coxas, peitos, carcaças, asas, miúdos). Cabe aos frigoríficos grande parte da coordenação do funcionamento desta cadeia produtiva.
- **Canais de comercialização:** São os canais responsáveis pela armazenagem, a distribuição e a comercialização, nos mercados local, nacional e internacional, passando pela embalagem e pelo tratamento técnico de conservação, que possibilitam a longevidade dos produtos. Os produtos da cadeia — frangos e ovos — também podem ser comercializados nos mercados atacadistas e varejistas — feiras,

açougues, supermercados, o que possibilita a participação, na cadeia, de grandes, médias e pequenas empresas

- **Consumidor Final:** Na sequência, está o último elo, o do consumidor final, representado tanto pelas pessoas do mercado nacional como do mercado internacional.

No tocante a questões socioeconômicas, a avicultura incentiva o emprego no campo para milhares de produtores que provêm a matéria-prima para a agroindústria. Portanto, as atividades dessa cadeia produtiva refletem a importância do setor nas questões econômicas e sociais, uma vez que no ano de 2020 foram contabilizados mais de 5 milhões de empregos (diretos e indiretos) e representou aproximadamente 2% do PIB brasileiro (ABPA, 2020). Sendo que o Brasil é o terceiro maior produtor de frango de corte do mundo, atrás de Estados Unidos e China; os dados do Relatório Anual de 2020 da Associação Brasileira de Proteína Animal (ABPA) indicam o crescimento superior a 133% na produção de frango de corte desde o início do século XXI, entre 2001 até 2020, passando de uma produção de 5,9 milhões de toneladas para 13,8 milhões de toneladas ao ano.

Ademais, segundo dados do Departamento de Agricultura dos Estados Unidos (USDA), no ano de 2020, o Brasil foi o maior exportador de carne de frango do mundo, cerca de 35% de toda a exportação mundial, seguido pelos Estados Unidos (29%) e União Europeia (14%) e a projeção até 2029 é a manutenção dessa liderança (USDA, 2021). Por fim, segundo a resolução número 4.174 de 2012 do Ministério da Agricultura, Pecuária e Abastecimento (MAPA) fica determinada que a classificação do produtor rural (pessoa física ou jurídica) é realizada de acordo com a receita bruta agropecuária anual auferida ou, na falta desta, a receita estimada nos seguintes critérios:

I - Pequeno produtor: receita até R\$500.000,00 (quinhentos mil reais);

II - Médio produtor: receita acima de R\$500.000,00 (quinhentos mil reais) até R\$2.400.000,00 (dois milhões e quatrocentos mil reais);

III - Grande produtor: acima de R\$2.400.000,00 (dois milhões e quatrocentos mil reais).

A resolução dispõe sobre a classificação de produtores rurais e sobre critérios para a apuração de saldos, para a fiscalização de financiamentos rurais e enquadramento tributário e regulatório das autoridades competentes (MAPA, 2012).

Na seguinte subseção é caracterizada a Avicultura de corte no estado do Paraná com um breve contexto histórico e seu posicionamento a nível nacional.

2.1.2 Ambiência animal na avicultura de corte

A preocupação com o bem-estar animal segundo relato de Nazareno et al. (2011) reflete de maneira positiva não somente para o mercado consumidor como também para organizações de proteção animal e demais entidades civis, uma vez que demonstra o comprometimento com as questões éticas por parte do produtor e da indústria de alimentos.

Cada vez mais os consumidores exigem produtos cárneos com atributos diferenciados, provenientes de animais criados e abatidos respeitando-se as normas de bem-estar animal (VIEIRA ET AL; 2016). A manutenção da adequada condição térmica em regiões de clima tropical é um desafio (NALENDRA ET AL; 2021; ARSHAD ET AL; 2022).

Na cadeia produtiva de frango, as preocupações com o bem-estar animal incluem fatores como temperatura, iluminação, qualidade do ar, da água, controle microbiológico e ventilação (NALENDRA ET AL; 2021; ARSHAD ET AL; 2022).

Mesmo com a evolução no desenvolvimento de pesquisas com aves geneticamente mais fortes e resistentes a fatores ambientais, isso não é suficiente para anular as condições e restrições microambientais - temperatura, umidade, intensidade da luz, velocidade do ar, controle microbiológico e etc. (ARSHAD ET AL; 2022).

A elevação da temperatura e da umidade relativa do ar podem aumentar as perdas por mortalidade, por exemplo. De acordo com VIEIRA ET AL; 2016, frangos de corte submetidos a condições ambientais desfavoráveis podem reduzir o consumo de ração, prejudicando a taxa de crescimento, o rendimento do peito e até mesmo a qualidade da carne. Desta forma, faz-se necessário o monitoramento constante de parâmetros ambientais nas instalações avícolas (DONKOH, 1989; SILVA; VIEIRA, 2010; VIEIRA ET AL; 2016).

Silva e Vieira (2010) citam que entre os anos 2002 e 2010 as perdas pré-abate por conta do estresse térmico eram próximas a 1% em toda a produção nacional. Corroborando aos autores supracitados, Abreu e Abreu (2011) relataram que durante o estresse térmico podem ocorrer mudanças comportamentais, fisiológicas,

hormonais e moleculares (ABREU; ABREU, 2011). A exposição de aves a temperaturas ambientes fora da zona termoneutra durante o curso da produção pode afetar adversamente a eficiência da produção e taxa de conversão alimentar, rendimento de carne, resposta imune e mortalidade (SILVA; VIEIRA, 2010; ABREU; ABREU, 2011). Frangos de corte acima dos dois quilos e meio podem precisar ser criados em temperatura ambiente mais baixa do que a relatada anteriormente devido ao seu maior peso corporal e taxas metabólicas (SILVA; VIEIRA, 2010).

A temperatura corporal de um frango adulto varia entre 40,6 a 41,7°C, e a zona termoneutra que permite que as galinhas mantenham sua temperatura corporal é de 18°C a 24°C., no entanto, sob alta temperatura ambiente, frangos de corte de crescimento rápido de alto peso corporal são particularmente suscetíveis ao estresse térmico devido ao seu alto nível de produção. Além das condições térmicas, índices de iluminação muito agudos (pouca luz e muita luz), concentração de gás amônia (principalmente causada pelas fezes dos animais), excesso de CO2 também influem no bem-estar e desempenho ideal (DONKOH, 1989; ABREU; ABREU, 2011).

No Quadro 3, a seguir, são discriminadas as condições ideais de dados como temperatura, umidade do ar, iluminação e amônia.

Quadro 3 - Valores ideais de ambiência para frangos de corte

Idade em semanas	Temperatura ambiente	Umidade do ar (%)	Iluminação (lux)	Gás amônia (ppm)
1	32 – 35	60 a 70	30-40	Abaixo de 50
2	29 – 32			
3	26 – 29			
4	23 – 26		10-20	
5	21 – 23			
6	20 - 22			
7	20 - 22			

Fonte: o autor (2023), adaptado de Gundim (2015)

A regulação incorreta de luminosidade pode acarretar em queda no rendimento da taxa de crescimento, consumo de alimentos e até mortalidade, Donkoh (1989), Abreu e Abreu (2011) citam que o período ideal para luminosidade na granja seja próximo de 23 horas diárias para frangos até 7 dias a partir disso ter 18 horas de luz e 6 horas de escuro, variando a intensidade da luz interna entre 10 a 40 lux de acordo com as semanas do frango mostradas no quadro acima. A conversão alimentar também demonstra estar relacionada com a iluminação adequada.

Há conforto no ambiente quando as aves possuem condições ideais para manifestar seu máximo desempenho. Porém, as diferenças que existem entre os

limites escolhidos como condições de conforto nos sistemas de criação de aves, tornam difícil sua adequada execução em campos produtivos (DONKOH, 1989; SCHWEAN-LARDNER; CLASSEN, 2010; ABREU; ABREU, 2011).

Os atributos ambientais de temperatura de bulbo seco, umidade relativa e velocidade do ar, os parâmetros aéreos de concentração de gases (dióxido de carbono, amônia, monóxido de carbono), a presença de poeira e microorganismos no interior das instalações são fatores utilizados para avaliar o ambiente em que as aves estão alocadas e que influenciam na qualidade de vida do frango de corte e, por consequência a sua produção e bem-estar (SILVA; VIEIRA, 2010; VIEIRA ET AL; 2016).

2.2 GESTÃO DE DADOS NO AGRONEGÓCIO

A gestão de dados rurais deve levar em conta elementos financeiros, produtivos, técnicos e sociais. Além de que, o administrador rural deve ter capacidade de agir de forma rentável com relação aos negócios, sempre com responsabilidade no âmbito socioambiental (FERRAZ; PINTO, 2017). Nesse contexto, no Quadro 4 foram descritas pesquisas no âmbito do uso direto da tecnologia para a gestão de dados em propriedades rurais.

Quadro 4 - Descrição de estudos sobre TI e GI no agronegócio

Descrição do estudo	Autor(es)
Investigaram o uso de um sistema <i>web</i> e <i>Open-Source</i> com informações de cunho geográfico e geoespacial. Por meio desse sistema foi possível integrar e gerenciar dados de pequenos produtores rurais por meio de um banco de dados centralizado de modo que era possível monitorar em tempo real os rebanhos bovinos e as condições climáticas dos próximos dias facilitando o planejamento dos agricultores cooperados, foco do estudo. Além disso, o sistema emitia relatórios pré-formatados e personalizados para cada propriedade aumentando ainda mais a gestão da informação individual para cada fazendeiro como a gestão coletiva da informação pela cooperativa.	(OLIVEIRA et al; 2014)
Houve a proposta de um modelo multidisciplinar para agricultura inteligente por meio de armazenamento de dados em nuvem. O modelo tecnológico proposto foi avaliado como benéfico para aumento da produção agrícola, contudo os autores apontam como ponto de atenção os custos e dependências do provedor do serviço em nuvem.	(CHANNE; KOTHARI; KADAM, 2015)
Com um recorte direcionado a aplicabilidade dos sistemas ERP (<i>Enterprise Resource Planning</i>) na governança da cadeia produtiva do agronegócio, investigaram as experiências de empresas agroalimentares que já utilizaram um sistema ERP. Os autores pesquisaram as barreiras e os incentivos para a adoção de ERP no setor hortícola e os resultados mostraram que o alinhamento desse sistema com as características e necessidades de uma empresa específica é um desafio pela complexidade dos processos e nos custos de implementação, no entanto, crucial pois a médio prazo apresenta capitalização maior para os fazendeiros.	(VERDOUW; ROBBEMOND; WOLFERT, 2015)

O artigo apresentou o desenvolvimento de um aplicativo móvel projetado para apoiar o consórcio de monitoramento de tempo/espaço de ocorrências da ferrugem asiática na soja. E sua avaliação foi tida como boa pelos usuários uma vez que forneceu uma valiosa quantificação da intensidade da ferrugem da soja em diferentes anos e áreas produtoras de soja no Brasil ajudando no mapeamento do problema e focando os esforços das políticas públicas com maior eficiência.	(ANDREIS et al; 2016)
Os autores tinham como proposta uma arquitetura da informação de baixo custo e baseada em IoT e usando <i>MIT app</i> inventor para monitorar a temperatura de torrefação. Como resultado o protótipo foi avaliado por parte do quadro funcional da empresa e tido como satisfatório devido a possibilidade de explorar as informações coletadas dos sensores apoiando a tomada de decisão.	(CALDERON et al; 2018)
O artigo descreve a arquitetura da informação de um aplicativo móvel para agricultores para apoiar nos cálculos agrícolas. E os resultados foram que um aplicativo finalizado facilitaria a formulação de políticas públicas nas atividades agrícolas e apoiariam os usuários com cálculos de previsões de lucro e de perdas da agricultura facilitando a gestão das informações agrícolas.	(ALZARLIANI et al, 2019)
A partir de uma abordagem de análise do sistema socioeconômico, o artigo explora o impacto das plataformas digitais no desenvolvimento de novas estratégias no setor do agronegócio. O artigo chama a atenção para a influência ambígua das plataformas digitais nos negócios agrícolas; por exemplo, quando o desempenho de uma empresa de alta tecnologia melhora com uma viabilidade desproporcional quando comparado aos médios e pequenos agricultores, isso é, a tecnologia melhora os processos de gestão no campo porém está acompanhada de uma exclusão digital no mesmo setor.	(REVENKO, L; REVENKO; 2021)

Fonte: o autor (2023)

Outros estudos, com uma perspectiva socioeconômica, como os de Canuto (2012), Castro de Paula, Chelotti (2018) e Dias (2020) também convergem na importância da gestão de dados rurais uma vez que o segmento se apresenta como um dos mais lucrativos na economia mundial e é tratado, segundo a organização das nações unidas (ONU), como fundamental para garantir a segurança alimentar de bilhões de pessoas no mundo (FAO, 2020; KUIPER, 2020).

2.3 INTERNET DAS COISAS

O conceito de *Internet* das Coisas está relacionado a uma entidade física de interesse individual (coisa) - objeto, como uma bicicleta, uma máquina industrial, um ar condicionado em uma sala, câmeras de segurança acionadas por detecção de movimento, lâmpadas com controle de iluminação e etc. (PIRES et al., 2015). Dependendo da natureza da “coisa” (dispositivo), diferentes tecnologias são utilizadas para conectá-los à *internet*, como por exemplo dispositivos de identificação (RFID, etiquetas ou códigos de barras), dispositivos de monitoramento como sensores

(temperatura e outros sensores, câmeras em veículos, porta fechadura ou aberturas de janela) e atuadores.

Segundo Albertin e Albertin (2017) a IoT consiste em uma rede global de bilhões de objetos exclusivamente identificáveis e endereçáveis, incorporados com sensores, atuadores e controladores) e estes são conectados à *Internet* no modo sem fio. A União Internacional de Telecomunicações determina a IoT como uma infraestrutura de rede global dinâmica que pode se autoconfigurar usando padrões e protocolos interoperáveis onde as coisas (físicas e virtuais) têm identidades, atributos e personalidades, usam interfaces inteligentes e podem se integrar perfeitamente na rede (KAFLE; FUKUSHIMA; HARAI, 2016).

Segundo Rajeswari, Suthendran e Rajakumar (2017) é possível classificar as tecnologias IoT em três tipos, conforme Quadro 5:

Quadro 5 - Classificação de Tecnologias para IoT

Classificação de tecnologias para IoT	Descrição
Tecnologias de detecção e coleta de dados	São responsáveis pela detecção e coleta de informações do ambiente físico (como, por exemplo, os sensores de temperatura, umidade e luminosidade) ou sobre objetos (identidade, estado, nível de energia).
Tecnologias de comunicação de dados	Configuram a forma como a transmissão de dados ocorre em uma determinada aplicação. Elas podem ser classificadas em duas principais categorias: com ou sem fio. As tecnologias com fio demandam uma infraestrutura física adequada para a passagem de cabos, podendo ser dispendioso e inviável em alguns casos. As tecnologias sem fio exigem menos conexões de <i>hardware</i> , facilitando a implementação em locais de difícil acesso e escassez de recursos elétricos. Os meios de comunicação sem fio incluem <i>wi-fi</i> , <i>bluetooth</i> e comunicação via dados móveis
Tecnologias de armazenamento e análise de dados	Englobam as aplicações IoT, a análise e o gerenciamento de dados e as plataformas de aplicativos. De forma geral, podemos dizer que as aplicações IoT são os <i>softwares</i> em execução que coordenam a interação entre pessoas, sistemas e dispositivos no contexto de um determinado propósito. Essas aplicações permitem a interação dispositivo-dispositivo e humano-dispositivo de forma confiável e robusta

Fonte: o autor (2023), adaptado de Rajeswari, Suthendran e Rajakumar (2017)

Existem diferentes tipos de arquiteturas, e independente de qual seja o que diferencia umas das outras são suas camadas de comunicação e segurança. Nas duas subseções a seguir serão abordados modelos de arquitetura IoT e protocolos de comunicação. Esses dois itens estão relacionados com a capacidade de um sistema IoT atender as três classificações de tecnologias do Quadro 5.

2.3.1 Arquiteturas IoT

Ao longo dos anos, vários projetos relacionados à *Internet* das Coisas (IoT) especificaram suas próprias versões de arquiteturas, baseando-se nos requisitos específicos aos quais os projetos estavam lidando (KRČO; POKRIĆ; CARREZ, 2014; AL-QASEEMI ET AL; 2016; SRINIVASAN ET AL; 2019). Dependendo do escopo do projeto ou do domínio do problema a ser abordado, as arquiteturas estavam se concentrando em diferentes aspectos ou em um subdomínio da IoT sem uma arquitetura consolidada e sempre adaptável (RAFIQUE ET AL; 2020).

De acordo com Tan et al. (2018), devido a uma grande heterogeneidade de domínios de aplicação e conseqüentemente dos requisitos, as abordagens para a especificação da arquitetura diferiram entre os projetos resultando em arquiteturas mais ou menos diferentes, compostas por uma série de componentes e protocolos. Ainda segundo o autor, isso resultou em uma interoperabilidade limitada entre os sistemas que também dificultou as discussões entre os domínios.

Uma pesquisa realizada por Sruthi e Kavitha (2016) indicou que a opinião da comunidade IoT era de que modelos de referência IoT são necessários, e os principais objetivos de um modelo comum são permitir a interoperabilidade entre as soluções, promover o entendimento comum da IoT e facilitar a integração com outros sistemas.

Os componentes mais importantes de um modelo de referência de IoT estão na terminologia, interface, modelo de interação, padrões, modelo de comunicação e modelos de segurança da informação.

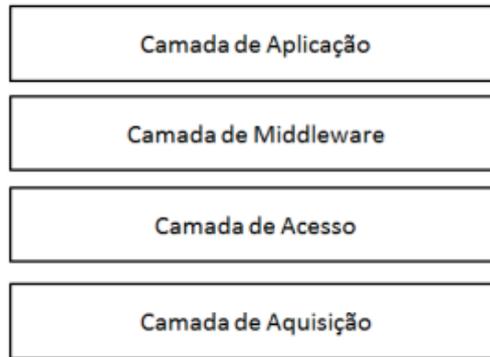
O estilo IoT consiste em um conjunto de componentes, portas, conectores e restrições, definidos com base na norma ISO/IEC 30141, esse padrão possui seis elementos fundamentais, sendo os sensores, atuadores, dispositivo que conecta a rede, controlador que possui o software interno responsável por ler os dados e enviar comandos aos atuadores (*firmware*), o *gateway* que funciona como uma ponte entre vários dispositivos e que se conecta a uma rede de longa distância e lida com a heterogeneidade de dispositivos e o armazenamento de dados, que armazena dados de contexto do IoT (SRUTHI E KAVITHA, 2016).

A seguir são apresentadas outras duas arquiteturas, a de quatro camadas e a arquitetura WSO2.

Arquitetura da IoT em Quatro Camadas: Tanto as arquiteturas propostas por Bandyopadhyay e Sen (2011), Bassl e Lange (2013), Chen. Et al. (2014), quanto

às sugeridas por Torkaman e Seyyedi (2016), Gokhale, Bhat e Bhat (2018), Ghade et al. (2020), Touqeer et al. (2021) e por podem ser organizadas em quatro camadas como mostra a Figura 2.

Figura 2 - Arquitetura IoT em quatro camadas



Fonte: o autor (2023), adaptado de Udoh e Kotonya (2018)

A camada de aquisição tem as funções básicas de obtenção de dados físicos; os sensores, atuadores, sistemas embarcados e etiquetas RFID estão nesta camada. Esses *hardwares* são responsáveis por prover identificação, coletar informações físicas do meio em que se encontram (sensores), atuar mecanicamente no meio (atuadores) e permitir o envio/recebimento de informações (BANDYOPADHYAY; SEM, 2011; BASSI; LANGE, 2013; GHADE ET AL, 2020; TOUQEER ET AL., 2021).

A Camada de Acesso é responsável pelo primeiro tratamento de dados recebidos e executa a comunicação com a *Internet*, normalmente utilizando Wi-Fi, cabo *ethernet*, *bluetooth* ou dados móveis (GHADE ET AL; 2020; TOUQEER ET AL., 2021).

Quanto à Camada de Middleware, Touqeer et al. (2021) relaciona três funcionalidades: i) interface com protocolos da camada de acesso; ii) abstração de dispositivos; iii) atuação sobre o contexto dos dados e oferecer abstração para a aplicação. Na primeira funcionalidade, a camada de middleware define os protocolos de troca de informações com as camadas inferiores. Na segunda, a camada atua sobre o contexto dos dados, ou seja, coleta as informações oferecidas pelos dispositivos e seleciona a informação que tem impacto sobre o processamento para tomar uma decisão. Na terceira, a camada utiliza uma interface REST para facilitar a interação entre cliente e dispositivos. Por fim, a Camada de Aplicação apresenta ao usuário de IoT os recursos oferecidos pelo sistema e permite atuações nos

dispositivos envolvidos. As aplicações se encontram em aplicativos *mobile*, *websites* e sistemas *desktop* (TORKAMAN; SEYYEDI, 2016, GOKHALE; BHAT; BHAT (2018); GHADDE ET AL, 2020).

Arquitetura de Referência da WSO2: O objetivo desta arquitetura é fornecer aos desenvolvedores um ponto de partida que contemple os requisitos de sistemas e tecnologias envolvendo IoT. Essa arquitetura possui três camadas, a saber:

- i) Camada de Dispositivos de IoT: Os dispositivos devem ter comunicação com a *Internet*. Além da capacidade de conexão, o dispositivo necessita de uma identificação única – endereço de IP.
- ii) Camada de comunicação: Deve suportar a conectividade entre os dispositivos com a rede; os dois protocolos mais conhecidos nesta camada são: (i) HTTP e (ii) MQTT. Além disso, a camada é responsável por gerar a interação dos dispositivos IoT com API's e com portais *web*.
- iii) Camada de processamento: provê capacidades para processar os dados, consistem, basicamente, em armazenar os dados em um banco de dados.

2.3.2 Protocolos de comunicação IoT

Nesta subseção foram descritos três protocolos mais utilizados em IoT segundo Krco, Pokric e Carrez (2014), MARJANI et al. (2017), Al-Sarawi et al. (2017), Çorak et al. (2018), Farooq, Wheelock, Pesch (2019) e mais recentemente com Sidna et al. (2020), Tightiz e Yang (2020) e Rubi e Gondi (2021), Costanzo e Flores (2021).

O primeiro protocolo de comunicação é o *HyperText Transfer Protocol* (em português, Protocolo de Transferência de Hipertexto) – ou HTTP que é utilizado para acessar informações na *Internet* usando requisição/resposta sobre o paradigma cliente/servidor. O HTTP foi desenvolvido para redes predominantemente formadas por computadores pessoais. O protocolo HTTP foi projetado principalmente para operar com o Protocolo de Controle de Transmissão (em inglês *Transmission Control Protocol* – TCP), sendo ineficiente para dispositivos com restrições.

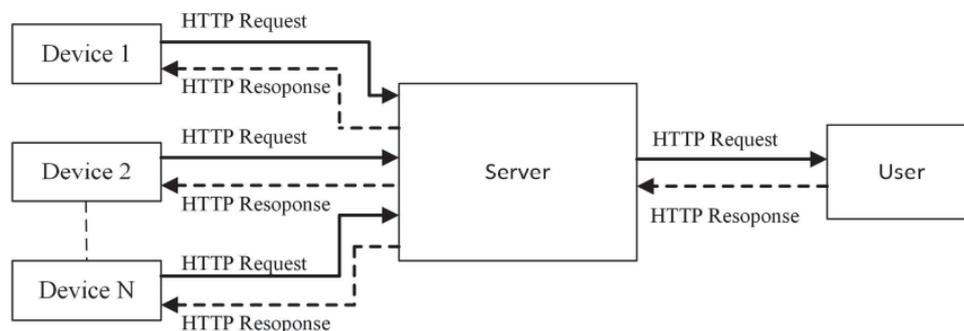
De acordo com Costanzo e Flores (2021) nos primórdios dos micros controladores haviam limitações nas capacidades de processamento e então o protocolo HTTP foi uma dificuldade para a transmissão de dados via *internet*, contudo, com os avanços de *hardware* isso foi superado e o protocolo, juntamente com o MQTT

é amplamente utilizado em sistemas IoT usando PLC's ou placas como ESP, Arduino e outras.

Uma desvantagem do protocolo HTTP é que quando ele está sendo usado como protocolo de comunicação IoT, o mesmo precisa transferir um grande número de pequenos pacotes com dados e pode ocorrer uma sobrecarga gerando consumo de recursos de rede e atrasos nas entregas (RUBI; GONDI, 2021).

Por outro lado, como o HTTP é operado por TCP / IP, pode-se obter uma comunicação confiável. Os dados acessados em HTTP são transferidos com base no endereço IP e URL. Depois de muitas vezes estabelecer a liberação de uma conexão, a comunicação é concluída (ALIWARGA; SATRIATAMA; PRATAMA, 2020). Além disso é possível implementar *WebSockets* na comunicação HTTP, a qual possui transmissão de dados em tempo real mitigando problemas de um grande número de requisições ao mesmo tempo (COSTANZO; FLORES, 2021). A comunicação HTTP é ilustrada conforme a Figura 3.

Figura 3 - Arquitetura do protocolo HTTP



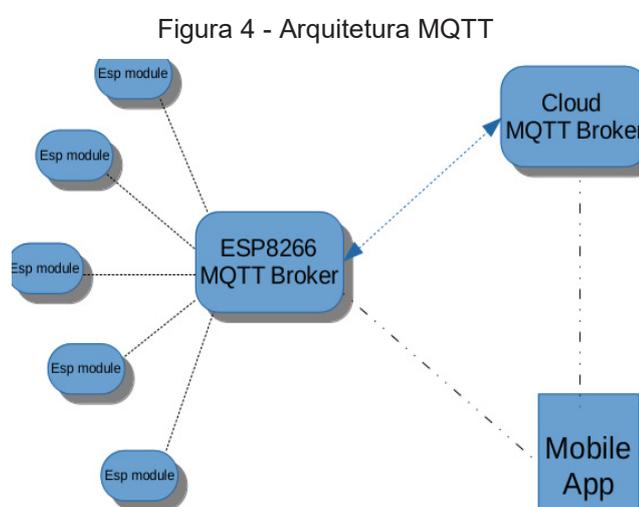
Fonte: ALIWARGA; SATRIATAMA; PRATAMA (2020, p.3)

Quando se utiliza *WebSockets*, Silva et al. (2018) e Rodriguez e Julcapoma (2020) afirmam que a latência e uso de rede ficam reduzidos, solucionando o problema da grande carga de dados para efetuar as requisições. Em outras palavras, o protocolo HTTP consegue entregar performance, segurança e confiabilidade para os dispositivos IoT como outros protocolos IoT, principalmente o MQTT, protocolo desenvolvido especificamente para a *Internet das Coisas*.

O segundo protocolo é o MQTT (*Message Queue Telemetry Transport*), e foi projetado para dispositivos limitados em *hardware* e utiliza a estratégia de *publish/subscribe* e um *broker* para transferir mensagens. Um dos objetivos do MQTT é minimizar o uso de largura de banda da rede e recursos dos dispositivos.

O MQTT também provê mecanismos para garantir a entrega das mensagens e faz uso dos protocolos TCP/IP nas camadas de transporte e rede respectivamente. O cabeçalho MQTT pode assumir tamanho fixo ou variável.

A Figura 4 mostra o funcionamento do MQTT com os seus três componentes básicos: *subscriber*, *publisher* e *broker*. O *subscriber* (computador ou dispositivo móvel) se inscreve num tópico (Esp8266 MQTT) no *broker* e fica aguardando a informação ser publicada. O *publisher* (Esp module) coleta a informação do ambiente e publica num tópico o valor correspondente e o *broker* envia essa informação para os *subscribers* que se inscreveram neste tópico.



Fonte: Aliwarga; Satriatama; Pratama (2020, p.5)

Como terceiro protocolo que também é utilizado em arquiteturas IoT está o *Constrained Application Protocol* (CoAP). Definido como um protocolo baseado na arquitetura *Representational State Transfer* (REST) é projetado para atender às necessidades específicas, tais como simplicidade e baixo custo operacional (ALSARAWI ET AL; 2017). Além disso, os recursos controlados pelo servidor são identificados e acessados por meio de identificadores de recursos universais (TIGHTIZ; YANG, 2020). O CoAP logicamente tem uma abordagem de duas camadas: uma camada de mensagens CoAP usada para lidar com o Protocolo de Datagrama de Usuário (em inglês *User Datagram Protocol* - UDP) e a segunda camada trata a natureza das interações requisição/resposta semelhante ao protocolo HTTP (RUBÍ; GONDIM, 2021).

Dentre outros protocolos, Constanzo e Flores (2021) citam alguns menos comuns, porém não serão aprofundados nessa tese por não estarem no escopo de

interesse da pesquisa, (i) *Advanced Message Queuing Protocol* (AMQP); (ii) *Extensible Messaging and Presence Protocol* (XMPP); (iii) *Universal Plug-and-Play* (UPnP); (iv) *Data Distribution Service* (DDS).

2.3.3 Microcontroladores, Sensores e Atuadores

A prototipação de dispositivos IoT são baseados na utilização de plataformas de *hardware* economicamente acessíveis que fornecem uma maneira simples e rápida de programar, denominada de microcontrolador (MAMAT ET. AL. 2021). O microcontrolador é um dispositivo semicondutor em forma de circuito integrado, que integra as partes básicas de um microcomputador (processador, memórias e portas de entrada e saída) para aplicações específicas. Suas vantagens giram em torno de baixo custo e o baixo consumo de energia e suas limitações estão em processamento com reduzida memória RAM e de armazenamento (KHALID ET. AL. 2021).

Na perspectiva de Marques (2021), os microcontroladores são circuitos eletrônicos que dão ao usuário acesso a interfaces de entrada/saída de dados, como portas digitais e analógicas e uma interface de comunicação, que podem ser conectados a um computador para serem programados (KHALID ET. AL. 2021). E graças aos pinos GPIO (*General Purpose Input/Output*) essas placas de *hardware* podem, por sua vez, ser conectadas com diversos tipos de sensores, atuadores e demais módulos de comunicação com ou sem fio para criação de projetos (MARQUES, 2021).

Dentre as plataformas de prototipagem em IoT, com *hardware* livre e de placa única se encontra o Arduino, projetada inicialmente para a linguagem C e C++. O objetivo do Arduino é a criação de ferramentas acessíveis, flexíveis e fácil entendimento para pessoas não familiarizadas com o desenvolvimento de *software*.

Outra família de microcontroladores são os modelos ESP, criado em 2014 pela empresa Espressif System se difundiu rapidamente no mercado de microcontroladores por ter um baixo custo, tamanho reduzido e a capacidade de conexão via WiFi (KHALID ET. AL. 2021). O primeiro módulo, esp-01, fabricado tinha a capacidade de operar via WiFi e possuir duas portas GPIO para conexão de sensores sua interface de gravação não era integrada ao módulo o que dificultava a adequação e praticidade em projetos, além disso a maior limitação era com a memória

interna de 512Kb, onde o *software* interno (*firmware*) ocupa 400Kb, praticamente ocupando toda a memória do dispositivo (KHALID ET. AL. 2021).

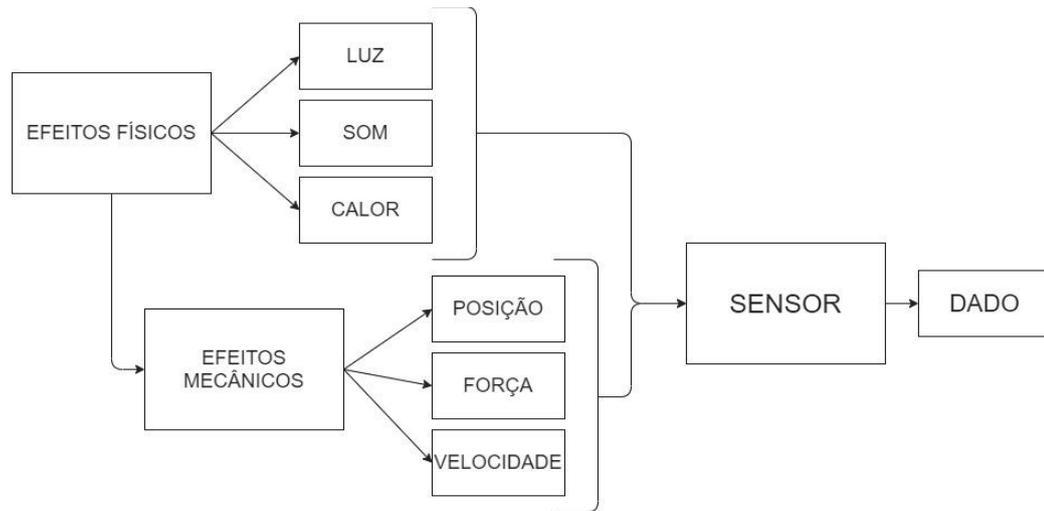
De modo a atender projetos mais robustos em IoT com uma maior capacidade de processamento e número de portas GPIO, o NodeMCU, uma plataforma *open source* traz acoplado o chip ESP8266; esse modelo apresentado conta com conversor Serial e regulador de tensão próprio. Também há pinos próprios para I2C, SPI e Analógico (MAMAT ET. AL. 2021). Esse módulo é capaz de atender projetos de automação residencial, redes de sensores, robótica, comunicação *Wireless* entre MCU's (*Micro controller Unit*), hospedagem de páginas *WEB* para controle de portas GPIO dentre outras aplicações (MAMAT ET. AL. 2021).

Além dos microcontroladores, nos estudos acerca de automação e digitalização de sistemas de informação, seja de cunho industrial, comercial, domésticos, é preciso determinar as variáveis a serem acompanhadas. Então é necessário obter os valores das variáveis físicas do ambiente a ser monitorado, e este é o trabalho dos sensores (THOMAZINI; ALBUQUERQUE, 2020).

E para ser possível mostrar a diferenciação entre alguns elementos presentes em uma automação de qualquer natureza é requerido o uso de sensores e atuadores, pois eles verificam e interferem o ambiente de acordo com a programação do sistema (THOMAZINI; ALBUQUERQUE, 2020; ZEYMER; SILVA; SCHNEIDER, 2021).

O termo “sensor” é empregado para designar dispositivos sensíveis a alguma forma de energia, que pode ser luminosa, térmica ou cinética. O objetivo é relacionar informações sobre uma grandeza que precisa ser medida, como temperatura, pressão, velocidade, corrente, aceleração, posição, entre outras. A Figura 5 alude o modelo de captura de dados de um sensor

Figura 5 - Modelo de captura de dados de um sensor



Fonte: o autor (2023), adaptado de Thomazini e Albuquerque, 2020

Existem dois tipos de sensores, os sensores digitais e os sensores analógicos, sua distinção se dá uma vez que um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle.

Os sensores analógicos podem assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que esteja dentro da sua faixa de operação. Além disso, várias das grandezas físicas podem assumir qualquer valor ao longo do tempo (pressão, temperatura, velocidade, umidade, vazão, força, ângulo, distância, torque, luminosidade). Essas variáveis são mensuradas por elementos sensíveis associados a circuitos eletrônicos com saídas não digitais.

Um sensor digital pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, que podem ser interpretados como zero ou um (valor booleano): desligado – OFF ou ligado – ON. Não existem naturalmente grandezas físicas que assumam esses valores, mas eles são assim mostrados ao sistema de controle após serem convertidos pelo circuito eletrônico de sensores (pressostato, termostato, chave de nível) ou transdutores (usado para determinação da posição ou velocidade), em que a saída se apresenta por meio de pulsos ou códigos binários (ZEYMER; SILVA; SCHNEIDER, 2021).

Porém é possível converter um sinal analógico em um sinal digital ou vice-versa, por meio dos conversores analógico-digital ou digital-analógico. Contudo é preciso estar ciente que quando uma conversão é realizada, parte do sinal é perdida e pequenas distorções (erros) são introduzidas no valor da grandeza realmente

medida. Ao transformar um sinal analógico em digital, o número de bits utilizado pelo conversor deve ser previamente escolhido pelo projetista para não obter valores falsos da grandeza física para não tornar o processo desnecessariamente caro, além de poder torná-lo mais lento durante a conversão (THOMAZINI; ALBUQUERQUE, 2020; ZEYMER; SILVA; SCHNEIDER, 2021).

Normalmente os sensores não são isolados, são sistemas compostos denominados transdutores. Que se conceituam por receber um dispositivo completo, que contém o sensor, usado para transformar uma grandeza qualquer em outra que pode ser utilizada nos dispositivos de controle (THOMAZINI; ALBUQUERQUE, 2020; BICA ET AL. 2021).

Um transdutor pode ser considerado uma interface às formas de energia do ambiente e o circuito de controle ou, eventualmente, entre o controle e o atuador. Os transdutores transformam uma grandeza física (temperatura, pressão etc.) em um sinal de tensão ou corrente que pode ser facilmente interpretado por um sistema de controle. Neste caso, o transdutor é o instrumento completo que engloba sensor e todos os circuitos de interface capazes de serem utilizados numa aplicação industrial (BICA ET AL. 2021). Os sensores possuem uma série de características que devem ser levadas em consideração ao selecionar um conjunto de sensores como mostrado no Quadro 6:

Quadro 6 - Características dos sensores

Característica do sensor	Descrição
Sensibilidade	Sensibilidade é a razão entre o sinal de saída e de entrada para um dado sensor ou transdutor. No caso de sensores analógicos, a sensibilidade refere-se à relação entre uma variação na grandeza em questão e à variação na medida fornecida pelo instrumento, ou seja, um sensor muito sensível é aquele que fornece uma variação na saída para uma pequena variação da grandeza medida.
Exatidão	Exatidão expressa a proximidade da concordância entre um valor tido como verdadeiro e o valor encontrado na análise. Consiste no erro da medida realizada por um transdutor em relação a um medidor padrão (valor de referência).
Precisão	É a característica relativa ao grau de repetitividade do valor medido por menor, conforme o tipo de grandeza medido, as condições de operação do sensor, estado de conservação e, é claro, a sua qualidade. Um exemplo de grandeza difícil de ser medida com precisão é a umidade, por isso a maioria dos sensores comerciais apresenta um erro de leitura entre 2% e 10%.
Alcance	Representa toda a faixa de valores de entrada de um transdutor. Por exemplo: a distância mínima e máxima que um sensor de movimento vai capturar o dado; Ou um sensor ultrassônico vai conseguir aferir a distância entre ele e um objeto.
Estabilidade	Está relacionada com a flutuação da saída do sensor. Se a flutuação for muito alta, ou seja, se o sensor possuir uma baixa estabilidade, a atuação do

	controlador que utiliza esse sinal pode ser prejudicada e apresentar valores imprecisos ou até errados.
Velocidade de resposta	Trata-se da velocidade com que a medida fornecida pelo sensor alcança o valor real do processo. Em sistemas realimentados o ideal é que o sensor utilizado tenha uma resposta instantânea, pois uma resposta lenta pode prejudicar muito a eficiência do sistema de controle e até impedir que o sistema funcione a contento.

Fonte: o autor (2023), adaptado de Thomazini e Albuquerque, 2020

Além dessas características, deve-se levar em consideração na especificação dos sensores as seguintes características como a facilidade de manutenção, custo, calibração, vida útil e etc. (THOMAZINI; ALBUQUERQUE, 2020). No Quadro 7 estão categorizados os sensores:

Quadro 7 - Categoria de tipos de sensores

Categoria dos sensores	Tipos de sensores de acordo com sua categoria
Sensores de presença	Sensores ópticos, Sensor infravermelho, sensor ultrassônico, senso a laser
Sensores de posição	Sensor de proximidade indutivo, sensor de proximidade capacitivo, sensor de proximidade magnético
Sensores ópticos	Sensor fotoresistor, sensor ultravioleta, sensor de cor, sensor fotovoltaico
Sensores de velocidade	Sensor dínamo taquimétrico, sensor de bobina, acoplador óptico
Sensores de aceleração	Acelerômetros de deslocamento, Acelerômetros de deformação, Acelerômetros a balanço de força, Acelerômetro integrado do tipo capacitivo
Sensores de temperatura	Sensor Termo resistência, sensor termopar, pirômetros, sensores termistores
Sensores de pressão	Sensores de pressão ópticos, Sensor de pressão integrado, Sensor de pressão capacitivo
Sensores de nível	Sensor nível vibratório (piezoelétrico), Sensor laser, Sensor célula de carga, Sensor Ultrassônico, Sensor Alfassônico, Sensor Fotoelétricos, Medidor de nível por borbulhamento, Medidor de nível por onda guiada, Sensor de nível por espectroscopia de impedância
Sensores de vazão	Sensor óptico, Magnético, magnético indutivo, Ultrassônicos, Medidor de vazão em válvulas
Sensores de tensão, corrente e potência	Sensor de corrente, sensor de efeito hall, Sensor de tensão CA e CC, Transdutores de potência
Sensores de umidade, gases e pH	Sensor de umidade, Sensores de gases, sensor medidor de pH

Fonte: o autor (2023), adaptado de Thomazini e Albuquerque, 2020

No tocante aos atuadores Thomazini e Albuquerque, 2020 os definem como dispositivos que modificam uma variável controlada. Recebem um sinal proveniente do controlador e agem no sistema controlado. De forma geral trabalham com potência elevada. Exemplos de alguns atuadores: i) Válvulas (pneumáticas, hidráulicas). ii) Módulos Relés (estáticos, eletromecânicos). iii) Cilindros (pneumáticos, hidráulicos). iv) Motores (Step-motor, Syncro motor, Servo motor). v) Solenóides.

Os atuadores podem atuar de forma independente aos sensores ou programados para responderem a uma ação de acordo com os dados que os sensores registam.

2.4 TECNOLOGIAS PARA DESENVOLVIMENTO

A seção aborda os assuntos sobre Banco de dados (relacionais e não relacionais) e Linguagens de programação que são necessárias para o desenvolvimento da arquitetura IoT proposta no escopo desta tese.

2.4.1 Banco de dados

Elmasri *et. al.* (2019) definem banco de dados como uma coleção de fatos registrados que refletem o estado de certos aspectos de interesse do mundo real. O conteúdo de um banco de dados representa uma visão instantânea do mundo real. E cada mudança em algum item reflete uma mudança ocorrida na realidade que se está representando. Na perspectiva de Silberschatz, Sundarshan, Korth (2020) banco de dados é uma coleção de dados utilizada pelo sistema de informação de uma determinada organização.

Existem duas categorias de bancos de dados, os relacionais e os não relacionais, os bancos relacionais são os mais utilizados no mundo dos negócios e são sucessores dos bancos hierárquicos e em rede (MEIER; KAUFMANN, 2019).

Nos bancos de dados relacionais define-se o esquema antes de adicionar os dados, por exemplo, um banco com nome, endereço e número de telefone possui a estrutura base para receber seus dados e se for adicionado um outro campo como idade ou sexo será necessário adicionar uma nova coluna, ou até mesmo construir um novo esquema com os novos relacionamentos (ELMASRI ET. AL; 2019).

Os bancos de dados não relacionais os esquemas são dinâmicos, ou seja, conjuntos de dados diferentes podem ser armazenados juntos, não sendo necessário criar um esquema para cada tipo de dado (PERKINS; REDMOND; WILSON, 2018); esse esquema também pode ser chamado de NoSQL; que denota um *software* de banco de dados que não requer o uso de SQL (*Structured Query Language*) para interagir com o banco de dados (PERKINS; REDMOND; WILSON, 2018).

Uma das características dos bancos NoSQL são a alta performance, sem tabelas, esquemas, necessidade de SQL ou até mesmo linhas; não requerem dados que possuam Atomicidade, Consistência, Isolamento e Durabilidade (ACID), além de *joins* e chaves estrangeiras (PERKINS; REDMOND; WILSON, 2018). Os dados armazenados neste tipo de modelo são constituídos por duas partes: “*string*” que representa a chave, e os dados a serem armazenados, que representam o valor, assim criando um par “chave-valor” (MEIER; KAUFMANN, 2019).

2.4.2 Linguagem de programação para web, mobile e embarcados

As linguagens de programação são um conjunto de convenções e regras que especificam como transmitir comandos e informações de pessoas para máquinas (SEBESTA, 2018; BERTOLINI ET. AL. 2019). As linguagens possuem diferentes níveis de abstração, sendo as de baixo nível e de alto nível. As linguagens com baixo nível de abstração são comumente chamadas de linguagens de máquina, e tratam-se de programação que compreendem as características da arquitetura do computador; assim, utiliza-se de instruções do processador (comunicação binária). Um exemplo é a linguagem Assembly, que trabalha diretamente com os registradores do processador, manipulando dados.

Com um nível maior de abstração, e mais próximo da sintaxe humana, as linguagens de alto nível se definem como códigos que passam por um compilador ou interpretador que os converte a linguagem de máquina, não sendo necessário o programador ter conhecimentos de código binário (BERTOLINI ET. AL. 2019).

De acordo com Sebesta (2018) diversas linguagens de programação foram criadas para os mais variados fins, como exemplo: aplicações científicas, militares, para sistemas desktop de cunho comercial, sistemas *web* diversos, desenvolvimento *mobile* e sistemas embarcados.

A programação *web* refere-se especificamente ao desenvolvimento de *software* para acesso via *internet*; os sites podem ser estáticos apenas informações *front-end* ou dinâmicos - com, por exemplo, integrados a banco de dados e algoritmos (SILVA, 2018; DUCKETT, 2016).

Dentre as linguagens para *web*, segundo Silva (2018) o *JavaScript* é uma das principais tecnologias da *World Wide Web (WWW)* juntamente com HTML e CSS. No caso do HTML e CSS ambas são utilizadas para especificar o conteúdo das páginas

da *Web* e o *JavaScript* é usado para especificar sua estrutura e comportamento. Além da interação dinâmica nas páginas *Web* com os usuários, a linguagem é capaz de interagir de forma assíncrona com os servidores, realizando tarefas de *back-end* (MORAES, 2021).

Isso se tornou possível graças ao *Node.js*, um ambiente de execução *JavaScript* baseado no motor *Google Chrome*, que fornece uma camada servidor completa para a criação de programas escaláveis e de alto desempenho. Além disso, com o *Node.js*, é possível criar aplicações *JavaScript* que não requerem um navegador para execução, ou seja, são autossuficientes na criação e manutenção de API's e micro serviços independentes (MORAES, 2021).

Além do ambiente *server-side*, a linguagem também está sendo executada em dispositivos móveis com aplicativos desenvolvidos utilizando bibliotecas (*frameworks*) híbridas como por exemplo *Xamarin* e *VueJS*, criados em 2011 e 2013 respectivamente (NOVICK, 2017); uma desvantagem de utilizar *frameworks* híbridos é a perda de desempenho quando o usuário executa o aplicativo, justamente por não serem aplicações nativas dos sistemas operacionais Android e iOS (NOVICK, 2017).

E para resolver o problema de desempenho supracitado, em 2015, o grupo *Facebook* lançou um *framework* chamado *React Native*, para o desenvolvimento nativo de aplicações Android e iOS, o qual converte código *JavaScript* para a linguagem nativa do sistema operacional *mobile*. As principais vantagens do *React Native* são: i) experiência do usuário fluída; ii) carregamentos de telas mais rápido; iii) melhor integração entre funções do celular como câmera, giroscópio, etc; iv) módulos de segurança mais robustos (LUCKENBAUGH, 2020).

Além dessas vantagens, antes do surgimento do *React Native*, desenvolver para Android e iOS era requerido utilizar as linguagens *Objective-C/Swift* (iOS) e *Java/Kotlin* (Android), no caso, eram necessários dois códigos com sintaxe diferentes para executar o mesmo aplicativo em sistemas operacionais distintos. Ou seja, com o *React Native*, o código pode ser reaproveitado em até 100% entre as plataformas *mobile* (LUCKENBAUGH, 2020).

A área de programação de microcontroladores e microprocessadores para controlar seus periféricos e manipular circuitos externos é denominada linguagem de programação de embarcados (DENARDIN; BARRIQUELLO, 2019). Esse tipo de programação se diferencia de programação para desktop, *web* e *mobile* pelas limitações de recursos, como falta de memória, ausência de um sistema operacional

completo, limitações nas formas de comunicação com o mundo externo e requisitos limitados de tempo de resposta para realizar uma tarefa que deve ser tratado pelo sistema embarcado (ALMEIDA; MORAES; SERAPHIM, 2017).

De acordo com Denardin e Barriquello (2019) as linguagens mais utilizadas e recomendadas em microcontroladores são C, C++ e Assembly; mas também é possível desenvolver em *Java*, *JavaScript* e *Python* graças a bibliotecas e *frameworks*, contudo, essas possuem limitações na comunicação com os sistemas embarcados além de causarem perda de performance por não serem linguagens nativa para embarcados.

2.5 CICLO DE DESENVOLVIMENTO DE UMA APLICAÇÃO

A seção apresenta as características do ciclo de desenvolvimento de uma aplicação, percorrendo toda a pilha de desenvolvimento; iniciando pelo design pattern, camada de serviço (*back-end*) e a integração com a camada *front-end web* e *mobile*, respectivamente. Por fim, a seção aborda o ciclo de desenvolvimento para sistemas embarcados.

2.5.1 Desenvolvimento back-end

A camada de Serviços (*back-end*), também conhecido como “*server-side*”, onde estão alocadas as regras de negócio da aplicação, a conexão com o banco de dados, a comunicação com *webservices* e também determina a autenticação de usuário e toda a parte de criptografia e segurança de uma aplicação (BERTOLINI, 2019).

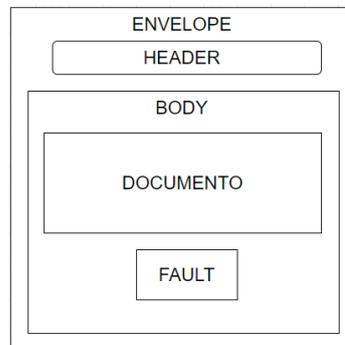
Existem duas principais formas de implementação de um *back-end* em uma aplicação *web*: O padrão SOAP (*Simple Object Access Protocol*) que consiste realizar o envio de pacote de dados com um documento no formato *XML* que carrega internamente o conteúdo da mensagem; e a arquitetura REST (*REpresentational State Transfer*), uma coleção de princípios e restrições arquiteturais para o desenvolvimento de aplicações distribuídas na *Web/Mobile* (DINIZ; SILVA, 2021).

No padrão SOAP as trocas de dados ocorrem no formato XML entre as entidades *Requester* (solicitante) e *Provider* (provedor) por meio da comunicação HTTP e possui três elementos, a saber:

- **Envelope:** Elemento raiz do documento XML. Toda mensagem SOAP deve contê-lo.
- **Header:** Um cabeçalho opcional. Quando utilizado, o header deve ser o primeiro elemento do envelope.
- **Body:** É um elemento obrigatório e contém a informação a ser transportada para o seu destino final. Pode conter um elemento opcional “*fault*”, usado para carregar mensagens de status e erros retornados ao processarem a mensagem

A Figura 6 ilustra a estrutura da mensagem de um envelope padrão SOAP.

Figura 6 - Estrutura de arquitetura SOAP



Fonte: o autor (2023), adaptado de Soni e Ranga (2019)

Por outro lado, a arquitetura REST, segundo Raemy (2020), adota o paradigma cliente-servidor, onde as requisições partem inicialmente do cliente e são respondidas pelo servidor utilizando métodos HTTP.

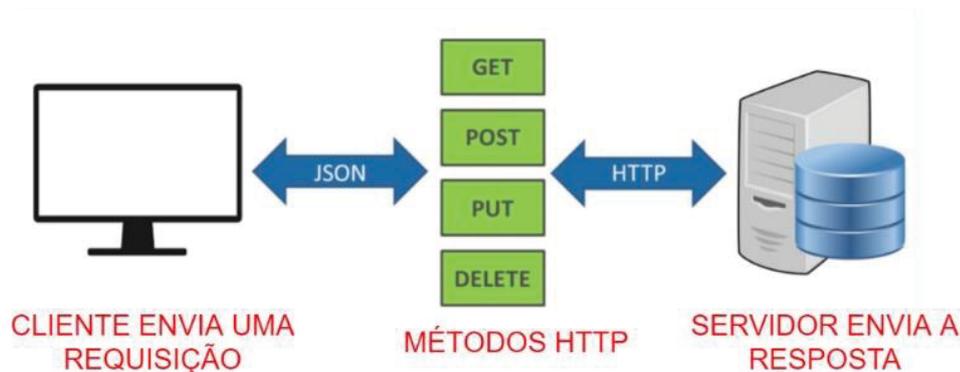
O princípio dessa arquitetura é a separação de responsabilidades, uma vez que ao separar as responsabilidades de interface das de armazenamento de dados, a utilização da arquitetura cliente-servidor permite que os componentes, o cliente e o servidor, se desenvolvam de forma independente. Por isso essa arquitetura é construída em camadas, contendo níveis hierárquicos, seu uso provê uma maior flexibilidade e simplicidade na comunicação entre sistemas diversos. Embora as interações entre cliente e servidor ocorram sempre nos dois sentidos, cada troca de dados pode ser tratada como um fluxo independente (RAEMY, 2020).

Ou seja, na arquitetura REST o servidor interpreta as requisições e retorna uma resposta *HTTP* ao navegador; essas respostas podem ser por meio de uma representação visual em formato *HTML*, ou em formato de erro afirmando que o recurso solicitado não foi encontrado. Esse processo é repetido diversas vezes no

período de navegação, a cada nova URL aberta ou formulário submetido são realizadas novas requisições para o servidor (DINIZ; SILVA, 2021).

Na Figura 7 é possível verificar a interação entre *server-side* e *client-side* em uma arquitetura REST.

Figura 7 - Estrutura da arquitetura REST



Fonte: o autor (2023), adaptado de Raemy (2020)

Diferentemente das demais abordagens, os serviços *web* REST estão focados nos recursos disponíveis e não nas chamadas a serviços. Logo, as requisições destinam-se a manipular representações de recursos, realizando operações para recuperar, inserir, alterar ou excluir (MODUGU; FARHAT, 2020).

Nesta proposta, a troca de mensagens utiliza o HTTP diretamente, como camada de aplicação. Logo, não existe empacotamento adicional dos dados intercambiados, além do próprio HTTP. Segundo Raemy (2020) os serviços *web* REST são conhecidos pela sua simplicidade, visto que as tecnologias empregadas são normas estabelecidas pelo *W3C/IETF* como HTTP, XML, URI, etc.

Adicionalmente, os serviços *web* podem ser construídos com ferramentas e linguagens de programação existentes no mercado. Logo, as barreiras para adoção desta proposta são mínimas (RAEMY, 2020). Por tratar-se de uma solução simples, que permite a utilização de formatos leves para troca de mensagens, tal como JSON (*JavaScript Object Notation*) (MODUGU; FARHAT, 2020), ou até mesmo texto simples, essa arquitetura proporciona escalabilidade para suportar um grande volume de clientes e acessos.

O fluxo de dados na arquitetura REST ocorre de maneira bilateral, e é realizada por meio dos métodos de comunicação do HTTP, descritos no Quadro 8.

Quadro 8 - Métodos HTTP

MÉTODO	DESCRIÇÃO
<i>GET</i>	O método <i>GET</i> solicita a representação de um recurso específico. Requisições utilizando o método <i>GET</i> devem retornar apenas dados.
<i>POST</i>	O método <i>POST</i> é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.
<i>PUT</i>	O método <i>PUT</i> substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.
<i>DELETE</i>	O método <i>DELETE</i> remove um recurso específico.
<i>HEAD</i>	O método <i>HEAD</i> solicita uma resposta de forma idêntica ao método <i>GET</i> , porém sem conter o corpo da resposta.
<i>CONNECT</i>	O método <i>CONNECT</i> estabelece um túnel para o servidor identificado pelo recurso de destino.
<i>OPTIONS</i>	O método <i>OPTIONS</i> é usado para descrever as opções de comunicação com o recurso de destino.
<i>TRACE</i>	O método <i>TRACE</i> executa um teste de chamada <i>loop-back</i> junto com o caminho para o recurso de destino.
<i>PATCH</i>	O método <i>PATCH</i> é utilizado para aplicar modificações parciais em um recurso.

Fonte: o autor (2023), adaptado de Modugu e Farhat (2020)

De acordo com Raemy (2020) esses métodos são os responsáveis por transportarem informações na *web*, por padrão é sempre o cliente que inicia a comunicação via requisição e conseqüentemente o servidor responde e isso acontece via texto, ou seja, as mensagens são e uma fração de bits de dados, mas o corpo da mensagem pode conter outros arquivos de mídia. As mensagens enviadas pelos métodos HTTP são feitas a partir de um cabeçalho (*header*) e o corpo da mensagem (*body*). Para haver o intercâmbio desses dados entre cliente e servidor em uma aplicação *web* é requerido um formato de dados universalmente interpretado pelos navegadores e compatível com qualquer linguagem de programação moderna.

Outro protocolo de comunicação para uso em arquiteturas REST é o *Websocket*, uma tecnologia que permite a comunicação bidirecional por canais sobre um único soquete TCP - *Transmission Control Protocol* (WANG; SALIM; MOSKOVITS, 2013). A interação do protocolo consiste em estabelecer uma conexão entre clientes e servidores inicialmente com cabeçalho HTTP utilizando um único *socket* TCP. Após conectado, é removido o cabeçalho HTTP e trocado para o cabeçalho *Websocket*, permitindo a simultaneidade na troca de mensagens em ambas as direções, de modo a oferecer uma conexão persistente e dedicada. Isso significa que o servidor também tem permissão de enviar mensagens para o cliente a qualquer momento sem necessitar de uma requisição do cliente (MIU ET AL; 2020).

Segundo Wang, Salim e Moskovits (2013), para se estabelecer uma conexão *WebSocket*, é requerido que o cliente e servidor iniciem usando o tradicional protocolo

HTTP, e então realizem a migração para o protocolo *WebSockets*; esse processo é conhecido como *Handshake*.

Na subseção seguinte será apresentado o JSON, um formato de dados que atende os requisitos de comunicação entre o servidor-*web*, servidor-*mobile* e servidor-embarcados.

2.5.2 JavaScript Object Notation (JSON)

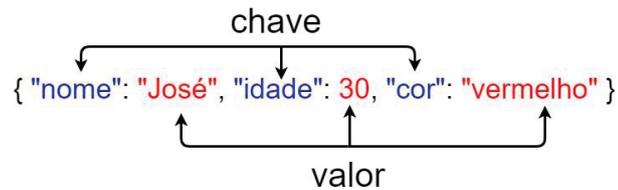
O *JavaScript Object Notation* (JSON) é um formato padrão de representação de dados largamente utilizado para dados semiestruturados independente da linguagem de programação (BREJE ET AL; 2018). Esse formato para intercâmbio de dados é considerado leve e simplificado, com uma sintaxe próxima à linguagem natural. Essa notação possibilita a composição de artefatos sem redundâncias de *tags*, como ocorre com o XML, tornando menor o volume de dados e, conseqüentemente, reduzindo a quantidade de memória necessária para o seu armazenamento, manipulação e transmissão.

Por possibilitar instanciar documentos menores, com estruturas simples e de fácil leitura, a notação JSON é adotada principalmente como formato escolhido para a transmissão de dados entre aplicações via *Internet* (SMITH, 2020).

A sintaxe é formada basicamente por uma estrutura hierárquica composta pelos artefatos estruturais de vetores, matrizes e objetos, e pares de elementos nome e valor, estes dois tipos primários de cadeias de caracteres, números, booleanos e nulos. Assim como no *JavaScript*, o JSON representa os valores de forma simples, atribuindo a eles nomes que os identificam. Embora se assemelhe com a sintaxe da linguagem *JavaScript*, JSON é independente da linguagem de programação e a estruturação de um documento na notação JSON é composta de forma hierárquica por uma sequência de vetores, matrizes e/ou objetos (TIWARY; STROULIA; SRIVASTAVA, 2021).

As informações são representadas por um conjunto de pares chave e valor, sendo que a chave é delimitada pela abertura e fechamento de aspas duplas, seguido de dois pontos e do respectivo valor. Como no exemplificado na Figura 8

Figura 8 - Elementos chave e valor JSON



Fonte: o autor (2023)

A estrutura de dados em JSON é flexível e permite representar dados em diversos formatos, como: *array*, *boolean*, *integer*, *number*, *null*, *object* e *string*. É o principal formato de dados adotados por bancos NoSQL. As estruturas dos dados JSON são flexíveis não sendo necessário definir um esquema rígido e definitivo como os utilizados para dados no formato “.sql”, por exemplo (SMITH, 2020).

2.5.3 Desenvolvimento front-end Web e Mobile

O objetivo inicial de aplicações *web* é facilitar a criação e o compartilhamento de informações entre usuários, sejam em atividades científicas, governamentais ou comerciais (AQUINO; GANDEE, 2016). Além disso, uma aplicação *web* compreende desde um simples *website* estático até um portal de comércio eletrônico com intenso processamento de informações. Nesse sentido, QUEIRÓS, R; PORTELA (2020) classificaram e detalharam cada tipo de sites para *web*, a saber:

- a) Site estático: é o tipo de aplicação mais simples baseada na *Web*, corresponde a uma coleção de páginas estáticas conectadas por hiperlinks (links) que permitem a navegação. Como exemplo dessas aplicações temos os documentos ou informações editados e publicados em HTML e CSS
- b) Site estático com formulários de entrada: este tipo de site possui um nível de interação básico implementado por meio de formulários de entrada. Tal funcionalidade permite uma pequena interação por parte dos usuários. Exemplo de aplicações deste gênero, podemos citar sites que possuem formulários de contatos, questionários, livros de visita, comentários e sugestões, cadastros, entre outros.
- c) Site com acesso a dados dinâmicos: por meio de consultas o usuário pode ter acesso a base de dados. As respostas das consultas são geradas

dinamicamente e geralmente apresentadas em formato HTML, XML, entre outros. Exemplo disso são *websites* que possuem sistemas de notícias, cadastro de informações em banco de dados.

- d) Aplicação *Web*: este tipo de aplicação baseada na *Web* herda todos os atributos das categorias citadas anteriormente e acrescenta a lógica dos negócios e operações, para promover a interação do usuário com os dados armazenados. Além disso, possuem estruturas similares aos *softwares* tradicionais (off-line). Exemplo disso podemos citar os sistemas bancários, *e-commerce*, *intranets* corporativas, entre outras. Para o desenvolvimento de aplicações *Web*, o conhecimento de sua arquitetura é um requisito fundamental. De um modo geral, as aplicações *Web* possuem três componentes básicos: um servidor *Web*, uma conexão de rede e um cliente. O servidor *Web* é um *software* executado em um computador remoto que responde a solicitações de outro *software* chamado cliente, via uma conexão de rede previamente estabelecida. Estes componentes podem estar estruturados em três camadas. Essas camadas separam as funções de interface do usuário (*front-end*), as funções de acesso ao banco de dados e às de lógica de negócio existentes em uma api para fazer o serviço de integração do *back-end* com o *front-end*.

O desenvolvimento de aplicativos móveis começou a se popularizar em 2008, com a chegada dos primeiros *smartphones* no mercado de tecnologia. Esses aplicativos abriram uma infinidade de possibilidades de auxílio na resolução de atividades diárias, funcionando como facilitadores em tarefas e no aumento da produtividade (PAUL; NALWAYA, 2019; BARUA; DOSHI; HIRAN, 2020).

Em síntese, os aplicativos móveis são *softwares* instalados em sistemas operacionais de *smartphones* e/ou *tablets*. Após instalados, eles podem ser acessados a qualquer instante, exigindo ou não uma conexão de *internet* disponível para sua utilização, dependendo das suas funções. Para o desenvolvimento de aplicações *mobile* são requeridas as linguagens nativas de seus sistemas operacionais, contudo existem linguagens e frameworks que substituem essa necessidade por um conhecimento nativo de cada linguagem para o desenvolvimento da mesma aplicação para diferentes sistemas operacionais (PAUL; NALWAYA, 2019).

No tocante ao desenvolvimento de uma aplicação *Full Stack* para *web* e *mobile* é necessário um processo bem definido, onde haja uma integração entre as

camadas de servidores, API indo até a camada visual e interativa com o usuário - *front-end* (BIØRN-HANSEN et. al. 2020).

De acordo com Pinheiro et al. (2020) os dois atributos chaves do desenvolvimento *front-end* são i) atenção aos requisitos e ii) a contínua revisão do conteúdo da informação que será disponibilizada. Em grande parte dos casos, no início do desenvolvimento, não é possível mapear o que um *website* ou aplicativo *mobile* irá conter por completo, uma vez que sua estrutura e funcionalidades evoluem com o tempo. Além disso, as informações que a aplicação vai fornecer ao usuário estão atreladas à camada de *back-end* (BIØRN-HANSEN et. al. 2020).

Sendo assim, o projeto *front-end* de uma aplicação demanda balanço entre o conteúdo da informação, estética e performance, o que envolve o conhecimento multidisciplinar do desenvolvedor.

Apesar de não existir um consenso sobre um modelo geral de ciclo de vida de uma aplicação *web* e *mobile* – o ponto em comum é ser evolutivo e incremental – algumas atividades típicas envolvidas na construção da aplicação são consideradas na maioria dos modelos: análise de requisitos, conceituação/abstração, prototipação, projeto, implementação e evolução ou manutenção (PINHEIRO ET AL; 2020); conforme descritas no Quadro 9.

Quadro 9 - Atividades do ciclo de desenvolvimento Web/Mobile

Atividade	Descrição
Análise dos requisitos	O desenvolvedor coleta as necessidades em termos de conteúdo, estruturação, acesso e estética. Requisitos de conteúdo definem a informação do domínio específico que deve estar disponível através da aplicação. Requisitos de estruturação especificam como o conteúdo deve ser organizado. Requisitos de acesso definem o estilo de acesso da informação que deve ser providenciada, o que inclui prioridades na apresentação da informação, indexação dos conteúdos, facilidades de pesquisas e suporte. Requisitos de estética definem as propriedades da aparência em geral.
Conceituação/Abstração	A aplicação é representada através de um conjunto de modelos abstratos que conduzem aos componentes principais da solução. No contexto da <i>Web</i> , o foco da abstração está na captura de objetos e relacionamentos de como irão aparecer para o usuário, mais do que como serão representados dentro do sistema. No <i>mobile</i> a maior preocupação é a forma como o usuário vai conseguir interagir com a tela e a adequação da aplicação em diferentes modelos de dispositivos.
Prototipação	É uma versão simplificada e funcional, realizada para uma avaliação prévia. Tipicamente, protótipos são construídos numa arquitetura simplificada com um conjunto de páginas implementadas manualmente contendo amostras do conteúdo da aplicação e da estética, simula a expectativa da aparência, comportamento e interação.
Projeto	As atividades de projeto devem destacar as constituições fundamentais de um site, abstraindo-se dos detalhes de implementação de baixo nível, e permitir ao projetista identificar estruturas recorrentes e padrões de navegação para serem reusados.

Implementação	A fase de implementação cria a aplicação a partir do projeto. Os elementos e relacionamentos destacados durante o projeto são mapeados na construção com a tecnologia de implementação escolhida. Entidades, componentes, links e estruturas de acesso são associados com outros elementos fornecidos pela tecnologia.
Evolução e manutenção	Depois de entrega, mudanças nos requisitos ou o conserto de erros podem requerer a revisão da estrutura, navegação, apresentação ou conteúdo. As mudanças são aplicadas no nível mais alto possível no ciclo de desenvolvimento e propagadas até o nível de implementação.

Fonte: o autor (2023), adaptado de Pinheiro et al., 2020

2.5.4 Desenvolvimento de Embarcados

Os sistemas embarcados são definidos por Denardin e Barriquello (2019) como sistemas para controlar uma função ou um conjunto de funções, baseados em microprocessadores, e que não são projetados para serem programados como os computadores pessoais.

Embora não sejam reconhecidos pela maioria das pessoas, esses sistemas estão presentes em vários dispositivos eletrônicos, tais como centrais telefônicas, drives de armazenamento, impressoras, equipamentos médicos e de redes de computadores (roteadores, hubs, switches), além de videogames, eletrodomésticos (micro-ondas, máquinas de lavar, *smart tv*), carros (controladores de tração, de motor, freios ABS), aeronaves (sistemas de controle inercial, controle de voo), dentre outros (GAMA ET AL; 2021).

Em função de sua vasta aplicabilidade em variados ramos da atividade humana, o processo de desenvolvimento de *softwares* voltados para o controle de sistemas embarcados possui diversas características que diferem ao processo de desenvolvimento de outros tipos de *software* de propósito geral. Diniz e Silva (2021), apontam algumas limitações e necessidades inerentes ao processo de desenvolvimento de sistemas embarcados como: a) capacidade de processamento e espaço em memória limitados; b) Necessidade de redução de consumo de energia; c) elevado grau de confiabilidade requerido por aplicações desse tipo; d) escolha de componentes de forma a otimizar a aplicação desenvolvida, dentre outras.

Denardin e Barriquello (2019) acrescentam que os sistemas embarcados em muitos casos são concebidos para que possam manter-se em funcionamento de maneira independente durante toda a sua vida útil, sem a necessidade de intervenções e reprogramação. Por isso, o desenvolvimento de sistemas embarcados

é uma tarefa bastante complexa em função principalmente do elevado grau de exigência em termos de qualidade e confiabilidade de seus produtos.

2.5.4.1 Ambientes de desenvolvimento para sistemas embarcados

O desenvolvimento de sistemas embarcados é abrangente uma vez que é possível programá-los em diversas plataformas. No caso de *hardwares* proprietários, o desenvolvedor disponibiliza *software* próprio para a sua programação como o exemplo dos Programadores lógicos controláveis (PLC); para os *hardwares* livre existem diversas opções de plataforma para a programação. Por exemplo, o ambiente de desenvolvimento do Arduino (ARDUINO IDE), a qual facilita a escrita de código e o upload para as placas de diversos microcontroladores (ARDUINO, 2021). Também existe a linguagem de programação Lua para embarcados, a mesma oferece bibliotecas e *frameworks* para desenvolvimento utilizando a linguagem por meio de IDE's como *Visual Studio Code*, *Eclipse*, *ZeroBrane Studio* (PATEL; DEVAKI, 2019).

Uma outra linguagem utilizada para o desenvolvimento de soluções de sistemas embarcados é a linguagem de programação *Node-RED*, a qual consiste em uma ferramenta voltada principalmente para o desenvolvimento de aplicações relacionadas ao conceito de *Internet* de Coisas (NODE-RED, 2021). O *Node-RED* utiliza uma abordagem de programação gráfica, ou seja, através do mesmo, torna-se possível elaborar uma aplicação por meio do estabelecimento de conexões entre blocos que possuem códigos predefinidos (conhecidos como nós) para a realização de determinadas tarefas. O *Node-RED* tem como objetivo auxiliar na elaboração de aplicações propondo uma maneira mais simples para o usuário que não domina muito as ferramentas de programação e também não possui conhecimento suficiente sobre estruturas utilizadas no desenvolvimento de sistemas embarcados (NODE-RED, 2021).

Outra alternativa para o desenvolvimento de aplicações de sistemas embarcados com uma interface gráfica é o MIT *App Inventor*, também conhecido como *App Inventor for Android*, essa solução possui código aberto originalmente criado pela *Google* e mantido pelo *Massachusetts Institute of Technology* (MIT). Com ela é possível desenvolver aplicações Android e integrá-las a placas de microcontroladores com bibliotecas próprias (MIT APP INVENTOR, 2021).

2.5.5 Segurança de dados em aplicações Web, Mobile e Embarcados

Na perspectiva da tecnologia da informação a segurança de dados visa cuidar da proteção de dados e informações contra ameaças, ataques cibernéticos que visam o furto de dados ou alteração de sua integralidade (FERREIRA, 2021). A segurança dos dados deve seguir quatro princípios básicos, a saber:

- Autenticidade: relacionada com a identificação de um usuário ou computador. A autenticação em um sistema deve assegurar a quem recebe a mensagem que ela é realmente procedente do emissor que está sendo informada em seu conteúdo (NBR ISO/IEC 27001:2019).
- Integridade: é a garantia da exatidão das informações e visa assegurar que pessoas não autorizadas tenham acesso e modifiquem, adicionem ou destruam essas informações (NBR ISO/IEC 27001:2019).
- Confidencialidade: é a garantia de que somente pessoas autorizadas tenham acesso a informação (NBR ISO/IEC 27001:2019).
- Disponibilidade: é a garantia de que as pessoas autorizadas possam acessar a informação sempre que necessário (NBR ISO/IEC 27001:2019).

E para que esses princípios de segurança de dados sejam atendidos é necessário modelar um mecanismo de segurança que atenda a necessidade de cada negócio (AUTOR). O uso de senhas de acesso são os mecanismos lógicos utilizados para impedir o acesso de pessoas não autorizadas a sistemas de informação. O *firewall* como sistema para controle de acesso na rede local também é amplamente utilizado como mecanismo de segurança (PAUFERRO; PAIVA; LESSA, 2020).

O mecanismo de criptografia consiste na transformação da informação em uma forma aparentemente ilegível, a fim de garantir a sua privacidade, ocultando a informação de pessoas não autorizadas (FERREIRA, 2021). Para que a criptografia se realize é utilizado um algoritmo com uma chave criptográfica, de modo que a decifragem somente será possível se o receptor possuir o acesso à chave.

Um outro mecanismo de segurança é a assinatura digital, método utilizado para que o registro de fatos ocorridos na *web* possa ter a mesma confiabilidade de um documento formal. A legitimidade dos documentos está intimamente ligada à sua originalidade e à certeza de que ele não foi alterado ao chegar no destinatário (PAUFERRO; PAIVA; LESSA, 2020).

2.6 EXPERIÊNCIA DO USUÁRIO (UX)

O capítulo apresenta os conceitos e definições de “Experiência do Usuário” - “*User eXperience* (UX)”, abordando sobre o objetivo de cada atributo na avaliação de qualidade. São apresentadas definições, métodos e parâmetros relacionados à avaliação de usabilidade, interface e experiência de uso relacionado a aplicações (*software*). O conceito de *User eXperience* (UX) pode ser comumente entendido como subjetivo e dependente de contexto (NORMAN; NIELSEN, 2010; MENDOZA; ET AL; 2020). Segundo a ISO 9241-210:2019 (2019, p.2), a UX é definida como “As percepções e respostas de uma pessoa que resultam do uso de um produto, sistema ou serviço”. Com o desenvolvimento de novas tecnologias, os usuários não estão procurando apenas realizar uma tarefa, mas também se entreter (MARQUES ET. AL; 2019).

A usabilidade na perspectiva do *design* de interface não é suficiente para definir a qualidade de um *software* e conseguir a aceitação do usuário (MENDOZA; ET AL.; 2020). O produto pode ser útil na questão técnica, porém desagradável e não aceitável pela maioria dos usuários (INAL, ET. AL; 2020). É neste contexto que a UX e usabilidade se encaixam, abrangendo detalhes sobre as interações entre usuários e produto, desde a percepção de como o produto funciona e se cumpre suas metas, necessidades e expectativas em qualquer contexto que eles usem o produto (BAÑUELOS, ET AL. 2021). De acordo com MENDOZA; ET AL.; 2020, UX está focada em abordar as necessidades humanas além do instrumental, ou seja, além dos aspectos voltados à realização de tarefas, para enriquecer a qualidade do produto e criar uma interação holística.

Segundo Ferrari e Gonçalves (2017) o método de Garrett para planejamento de produtos digitais que melhorem a experiência do usuário são exigidos elementos que devem ser levados em consideração, sendo eles ilustrados na Figura 9:

Figura 9 - Elementos de UX



Adaptado de GARRETT (2010, p.24)

Na camada de objetivos do site e necessidades do usuário estão as metas do negócio, metas criativas ou outras metas de origem interna para o site. Juntamente com a estratégia para atender as necessidades do usuário.

A segunda camada, requisitos de conteúdo e especificações funcionais, requer a definição dos elementos do conteúdo necessários ao site para ir ao encontro às necessidades do usuário, também o conjunto de funcionalidades detalhadas que a aplicação deve conter para atender aos usuários. O *design* de informação é a apresentação da informação para facilitar a compreensão. E a arquitetura de informação se refere à estrutura das informações na tela de modo a facilitar o acesso intuitivo ao conteúdo.

O *design* da informação se refere a como as informações vão se comportar na tela de modo a facilitar a compreensão por parte do usuário, nesse elemento estão o *design* de navegação: elementos da interface para facilitar a movimentação do usuário em meio a arquitetura da informação. E *design* de interface que visa facilitar

a interação do usuário com as funcionalidades do sistema. Por fim, o *design* visual se refere ao tratamento visual do texto, elementos gráficos da interface e componentes de navegação.

Uma vez identificado o conceito da experiência do usuário e os benefícios do *design* centrado no usuário e suas características, questiona-se como operacionalizar a avaliação da UX. Assim como toda ciência, a experiência do usuário apresenta métodos e métricas que possibilitam avaliar como se dá e se caracteriza a experiência do usuário com um sistema. De acordo com a ABNT NBR ISO 9241-171:2018, a avaliação centrada no usuário é uma atividade necessária em um projeto centrado no ser humano (ABNT, 2018). Para avaliar os processos de interação e as mensagens transmitidas a partir dos produtos para os usuários em relação às formas de uso, são realizados experimentos da experiência do usuário (dentre estes, estudos de usabilidade), que utilizam métodos, dimensões e métricas específicas, com intuito de mensurar e compreender diversos fatores durante a UX, Rivero e Conte (2017) realizaram um mapeamento sistemático fornecendo uma visão geral dos tipos de métodos de avaliação de UX.

O primeiro método SAM (*Self Assessment Manikin*) que avalia as emoções nas dimensões de felicidade/tristeza, excitação e domínio/controle. Esse método permite reunir informações sobre o grau de aceitação do *software*, as emoções que os usuários sentiram ao usá-lo e as informações sobre o grau de atributos retratados como: beleza, identificação e estímulo (LIAO, ET. AL; 2019).

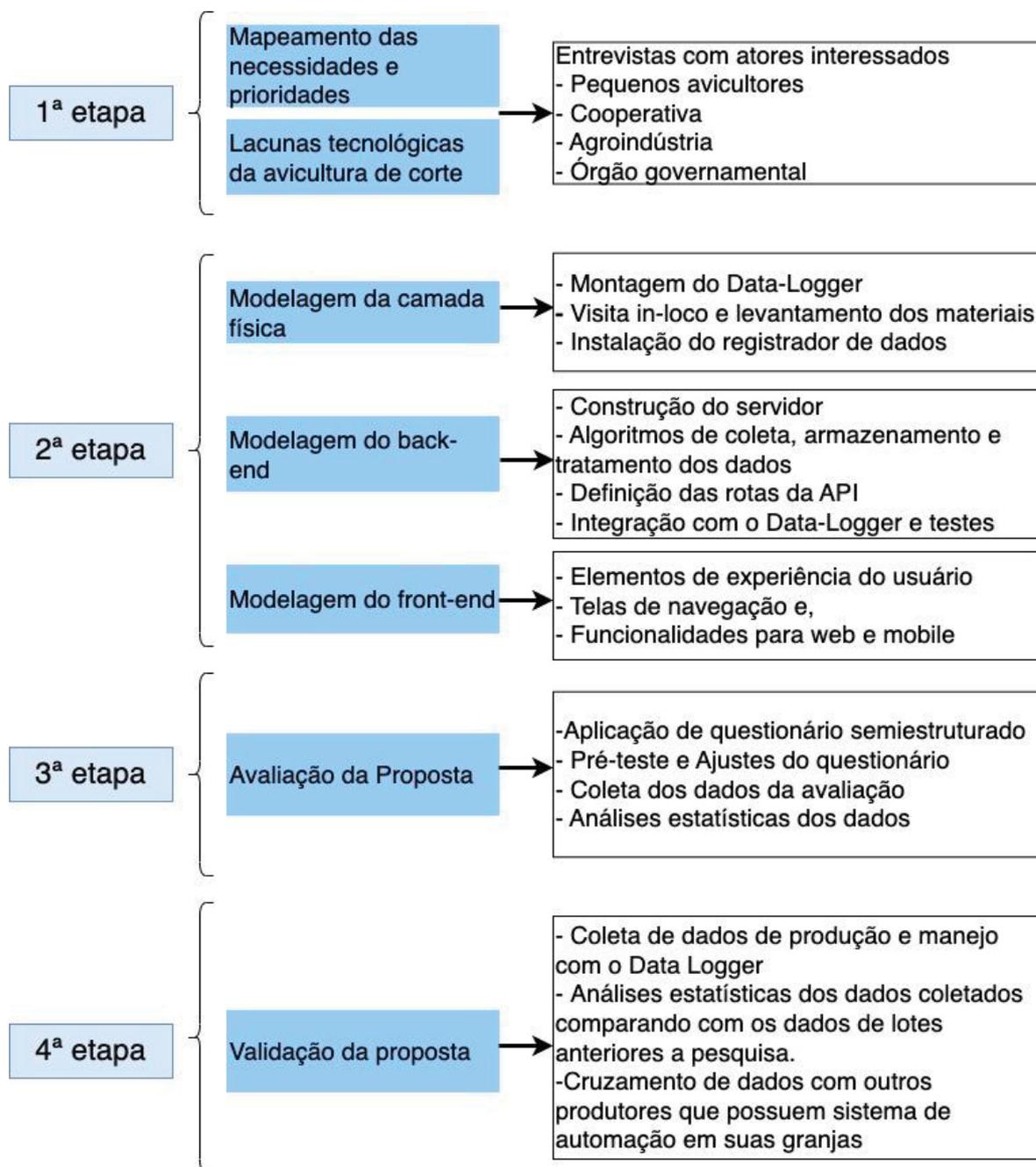
O segundo método é chamado de UEQ (*User Experience Questionnaire*) e é composto por 26 pares de atributos opostos entre si que medem a atratividade considerando os aspectos de: familiaridade, eficiência, confiabilidade, estímulo e novidade. Nessa técnica cada par corresponde a um adjetivo e seu antônimo, consistindo de uma escala diferencial semântica de 7 pontos entre eles, onde o avaliador deve marcar o ponto mais próximo do adjetivo que melhor descreve sua experiência com o usuário (HINDERKS ET. AL; 2020).

Outro método é a Análise Retrospectiva, onde os usuários são solicitados a relembrar informações sobre a experiência que tiveram. Nesse método os participantes são solicitados a esboçar como sua opinião sobre uma determinada qualidade do produto mudou com o passar do tempo. (MAIA; FURTADO, 2014).

3 METODOLOGIA

Os encaminhamentos metodológicos para o atingimento dos objetivos propostos seguiram as etapas explicitadas na Figura 10.

Figura 10 - Fluxo metodológico



Fonte: o autor (2023)

Etapa 1 (OBJETIVO ESPECÍFICO A) – Entrevistar produtores de frango de corte de pequeno porte, para mapear suas necessidades e prioridades. Além de encontrar as lacunas tecnológicas que o sistema pode atender.

Etapa 2 (OBJETIVOS ESPECÍFICOS B e C) – Estruturação da camada física da arquitetura, com o levantamento de materiais para a preparação do Data Logger. Modelagem da camada back-end e front-end (web e mobile) considerando os aspectos da experiência do usuário.

Etapa 3 (OBJETIVO ESPECÍFICO D) – Avaliar a proposta da plataforma web/mobile com os produtores, verificar sua aceitação e questões de sua experiência em utilizar o sistema.

Etapa 4 (OBJETIVO ESPECÍFICO E) – A última etapa consiste em validar estatisticamente a eficiência da arquitetura, cruzando os resultados de antes da implementação e depois dela. Além de comparar os dados com dados cedidos por produtor que possui um sistema de automação comercial e dados produtivos de pequenos produtores cooperados.

3.1 DELIMITAÇÃO DA PESQUISA E SELEÇÃO DOS ATORES

O campo da pesquisa são os pequenos produtores criadores de frango para corte (conforme classificados na seção 2.1.1) do estado do Paraná, que exercem as atividades contempladas no Cadastro Nacional de Atividade Econômica (CNAE) 0155-5/01, que por definição são atividades de avicultura, produção de esterco de frango, criação de frango para corte e criação de galináceos para corte (IBGE, 2021).

Por meio de busca documental no cadastro interno do Sindicato das Indústrias de Produtos Avícolas do Estado do Paraná (SINDIAVIPAR) no ano de 2022 estavam ativas 16.350 unidades de aviários de corte, sendo 70% delas pertencentes a pequenos produtores. Conforme detalhado na Tabela 1.

Tabela 1 - Número de aviários de corte por região

Região do Estado do Paraná	Número de aviários para frango de corte	Número de pequenos aviários
Noroeste	1.718	1.433
Norte Central	1.830	1.589
Norte Pioneiro	829	697
Oeste	6.040	3.710
Centro Ocidental	910	610
Centro Oriental e Metropolitano	945	634

Sudoeste	2.782	1.870
Centro Sul	97	65
Sudeste	199	134
TOTAL	16.350	10.742

Fonte: o autor (2023)

Na seleção dos atores, houveram o aceite de dois produtores de frango de corte. Ambos são cooperados na mesma cooperativa que possui operações em todo o oeste paranaense e até em outras regiões do país.

Contudo um dos produtores colocou restrições sobre detalhes de seu aviário e seu sistema de automação. O produtor é de médio porte e será chamado de “PRODUTOR 1”. O mesmo possui oito granjas (150m x 16m) divididas em 2 propriedades e com capacidade para até 25 mil frangos por lote cada galpão.

Para colaborar com a pesquisa o PRODUTOR 1 permitiu a instalação de um pequeno registrador de dados com um sensor de temperatura e umidade e um sensor de gás dióxido de carbono (MQ-135) o qual permite verificar a qualidade da ventilação. A instalação foi feita por um técnico de eletrônica que presta serviços para o próprio aviário e o mesmo notificou que deixou o *Data Logger* próximo a uma estação dos sensores já utilizados na granja. Como não foi possível instalar todos os sensores da arquitetura proposta, os sensores instalados serviram como parâmetro de calibragem dos sensores. Comparando os dados coletados pelo sistema de automação comercial e os sensores do pesquisador.

Como não foi possível ter acesso irrestrito ao aviário do PRODUTOR 1 para realizar a pesquisa, todos os detalhamentos da montagem dos registradores de dados são com o produtor de pequeno porte (PRODUTOR 2).

O outro participante da pesquisa (PRODUTOR 2), é de pequeno porte (foco da pesquisa) e possui um aviário com capacidade de até 10 mil aves por lote; seu aviário é do tipo convencional e utiliza a circulação de ar de pressão positiva. Seu aviário possui dimensões de 100m x 10m, e pode ser visto nas fotos da Figura 11 cedida pelo próprio proprietário.

Figura 11 - Fotos do galpão da pesquisa



Fonte: o autor (2023)

O sistema de climatização é composto por ventiladores e aspersores de água, as janelas laterais são teladas e aramadas para ventilação providas de cortinas de lona retráteis, sistema de abertura manual por manivelas distribuídas em volta ao aviário.

Ambos os aviários (PRODUTOR 1 e PRODUTOR 2) são de pressão positiva, onde a ventilação é forçada por meio de ventiladores de fora para dentro renovando o ar, inclusive acionando nebulização para aumentar a umidade e ajudar no controle da temperatura

3.2 PROCEDIMENTOS DE COLETA E ANÁLISE DE DADOS

A fim de especificar questões relacionadas às necessidades, prioridades, problemas e limitações no tocante às tecnologias IoT para gestão do manejo de granjas de frango de corte optou-se pela técnica de entrevistas semiestruturadas para a coleta de informações advindas de produtores, gestores agroindustriais, cooperativas, e outras organizações interessadas.

Devido às restrições da Lei Geral de Proteção de Dados (LGPD - Redação dada pela Lei nº 13.853, de 2019) não foi possível obter uma lista de contato dos produtores diretamente com algum órgão do setor, que dificultou aumentar o número de candidatos a participar da coleta dos dados.

Devido a restrições por conta da pandemia de Covid-19 as entrevistas foram realizadas em formato digital, sendo os entrevistados da região oeste do Paraná, próximo a cidade de Foz do Iguaçu-PR.

Participaram das entrevistas sete pessoas, sendo quatro avicultores, um gestor de agroindústria de abate de frangos, um coordenador de cooperativa e um servidor público municipal de secretaria de Agropecuária e Meio Ambiente.

As entrevistas foram descaracterizadas de modo a preservar a identidade dos respondentes e seus respectivos dados sensíveis, as respostas foram organizadas e apresentadas em forma textual de acordo com o roteiro de entrevista do Quadro 10.

A escolha por entrevistar produtores da região oeste paranaense se deu uma vez que essa região é a maior produtora de frango de corte do Brasil, conseqüentemente onde se concentram as maiores quantidades de granjas, cooperativas e agroindústrias do segmento.

As entrevistas dispõem de um resultado discursivo e seus apontamentos são categorizados de modo a permitir o mapeamento das necessidades e prioridades que uma arquitetura IoT necessita para ser eficiente no contexto em que está sendo implementada. Para delinear as entrevistas foi proposto um roteiro subdividido em quatro seções, conforme Quadro 10.

Quadro 10 - Roteiro de entrevistas

Nº	Sessão	Objetivo	Suporte Teórico
1	Caracterização dos respondente (mantendo o anonimato)	Descrever o perfil do respondente com relação a sua formação, experiência profissional, tempo no ramo e número de funcionários.	-
2	Relação do produtor com a agroindústria	Investigar as questões relacionadas à negociação entre produtor e agroindústria. Verificar o aporte de apoio técnico, matéria prima e dos frangos em estágio de incubação. Também questões relacionadas a forma de remuneração.	VOGADO ET. AL. (2016); GALLO (2015);
3	Questões práticas de produção e manejo	Compreender questões relacionadas aos processos produtivos e fluxos de entrada, manejo e logística. Etapas de pré-engorda, de engorda e crescimento e por fim, pré-abate.	ROWE; DAWKINS; GEBHARDT (2019); HAFEZ; YOUSSEF; ATTIA (2020)
4	Questões tecnológicas, problemas, demandas, prioridades	Mapear as necessidades, problemas, demandas relacionado aos sistemas automatizados existentes para granja, custo, manutenção, viabilidade e também coletar requisitos e <i>insights</i> dos interessados de funcionalidades não existentes ou inviáveis para sistemas de baixo custo	SALAM (2020); ASTILL EL. AL (2020); HAFEZ; ATTIA (2020)

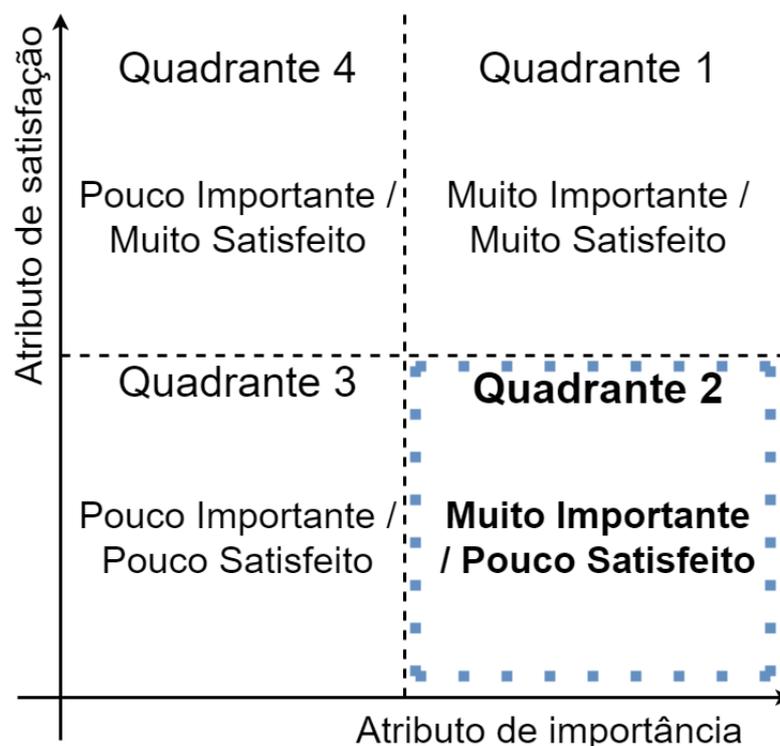
Fonte: o autor (2023)

Com a técnica de entrevistas semiestruturadas não houve a realização de um pré-teste, uma vez que essa técnica permite uma flexibilização do roteiro; e caso o entrevistado não entenda algum questionamento ou haja algum tipo de ambiguidade o pesquisador tem a liberdade de corrigir e/ou explicar (BONI, 2005). As questões norteadoras para as entrevistas estão no “Guia de Entrevista” no Apêndice 1.

De modo a viabilizar a análise e classificação das informações advindas das entrevistas optou-se pela “Matriz de *needs*” ou Matriz de Importância e Satisfação. Para esclarecimento, a Matriz de *needs* mede a intensidade das dores do usuário (cliente) quando ele está executando seu trabalho/tarefa; além disso, a matriz calcula o tamanho da oportunidade em cada uma das funcionalidades de um produto, mostrando as “dores do cliente” não resolvidas (OLIVEIRA; MAGALHÃES; FIALHO, 2021). Dessa forma a análise é direcionada de forma lógica e sistemática, levando em consideração o conjunto de critérios que servirão de parâmetro para orientar a construção da aplicação (DE SÁ, ET AL; 2019).

Pela Matriz (Figura 12) serão mapeadas as informações base para a construção da arquitetura IoT que irá subsidiar a gestão de dados de granjas pelos produtores via aplicação *web* e *mobile* contemplando aspectos de experiência de usuário e atendimento das regras de negócio.

Figura 12 - Matriz de *Needs*



Fonte: o autor (2023)

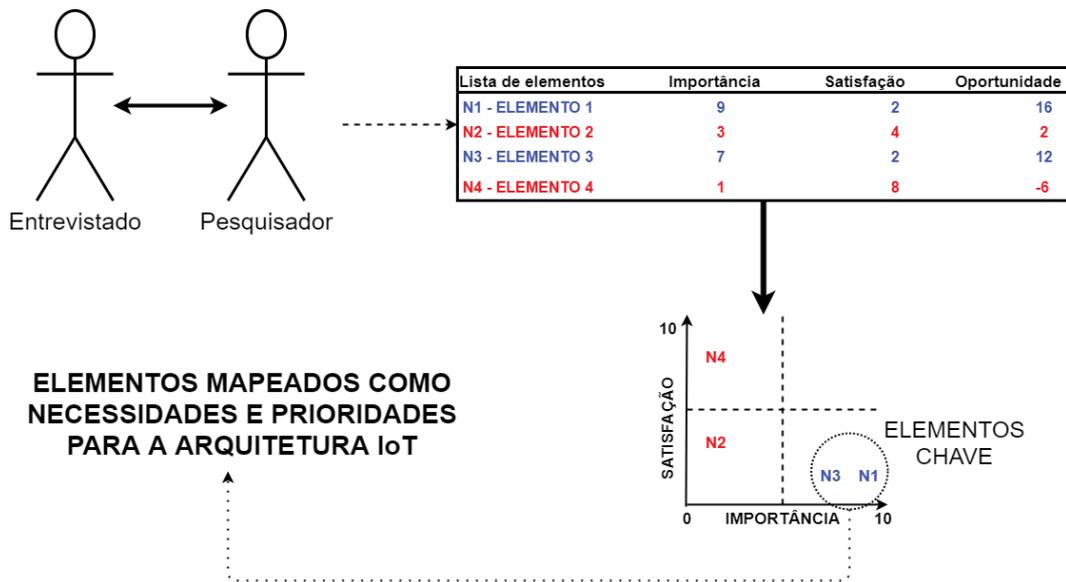
O foco de concentração nas funcionalidades são os elementos que se encontram no Quadrante 2, uma vez que será classificado como muito importante e que os usuários estão pouco satisfeitos, indicando uma oportunidade e necessidade por parte dos usuários. O cálculo de oportunidade é:

$$\text{OPORTUNIDADE} = \text{IMPORTÂNCIA} + (\text{IMPORTÂNCIA} - \text{SATISFAÇÃO})$$

Sendo que quanto maior o valor, maior a oportunidade, eventualmente o valor da oportunidade será negativo, devido a fórmula de cálculo, contudo, não há uma relação matemática posicional no preenchimento do gráfico de *needs*, uma vez que as coordenadas dos elementos são advindas da importância (x) e satisfação (y).

A construção da matriz vai ser guiada pelos elementos que surgirem durante as entrevistas, a Figura 13 esclarece o processo de construção.

Figura 13 - Preenchimento da matriz de *needs*



Fonte: o autor (2023)

Com a organização da matriz e o posicionamento dos elementos, é possível consolidar a base de construção da solução IoT. Os elementos classificados com nota entre 0 e 2 são considerados de baixa necessidade e prioridade, entre 3 e 4 – pouca necessidade e prioridade, 5 e 6 – prioridade e necessidade média, 7 e 8 – alta prioridade e necessidade e 9 e 10 – prioridade e necessidade máxima. A próxima subseção descreve os materiais utilizados para o desenvolvimento.

3.3 MATERIAIS

A construção da arquitetura IoT para pequenos agricultores rurais que engloba todo o ciclo de gestão de dados depende, primeiramente, da camada física, ou seja, são requeridos microcontroladores, sensores, atuadores e demais componentes eletrônicos conectados entre si para garantir a comunicação integrada com a camada de *software*. Assim, nessa subseção serão apresentados os dispositivos e componentes eletrônicos utilizados para a construção do projeto proposto. Ao final, no Quadro 12 é apresentada uma síntese do conjunto de eletrônicos selecionados com uma breve descrição do objetivo do uso e de suas funções.

3.3.1 Microcontroladores

A seleção da placa microcontroladora exige o conhecimento das características da mesma, tendo em vista que a sua programação será baseada nas especificações físicas da estrutura do *hardware* e do restante da arquitetura proposta.

A escolha da placa eletrônica embarcada NodeMcu baseou-se nos seguintes requisitos: pequena dimensão (5,3cm x 3 cm); compatibilidade com a IDE do Arduino e Linguagem C e C++; capacidade de acesso à rede WiFi e baixo custo de aquisição. O modelo possui 11 pinos digitais e 1 porta analógica, conversor analógico/digital e 3 portas de alimentação externa (portas Vin e 3V); para o protótipo da arquitetura utilizou-se o modelo fabricado pela Wemos, o Lolin V3; conforme Figura 14.

Figura 14 - NodeMCU placa Wemos Lolin



Fonte: Wemos (2021)

Esse modelo de placa é composto basicamente por um chip controlador (ESP8266 ESP12E), uma porta USB mini para alimentação e programação, conversor USB serial integrado e já possui WiFi nativo. A única restrição é que ele opera com pinos em nível lógico de 3.3V, porém possui um regulador de tensão aceitando tensão

até 9V. Possui também conversor USB Serial integrado, podendo ser programável via USB. Compatível com a IDE, módulos, sensores e atuadores também utilizados nas placas Arduino (KHALID ET. AL. 2021).

3.3.2 Sensores

Os sensores foram escolhidos a partir da revisão teórica sobre os aspectos de ambiência animal, qualidade e ganhos de produção para a avicultura de corte. Sendo assim, conseguindo mensurar esses dados é possível programar de maneira mais eficaz os parâmetros para o acionamento dos atuadores e até estimar o melhor momento para realizar a intervenção dos atuadores (ligando e/ou desligando).

A utilização dos sensores DHT11 e BME280 serão importantes para conseguir monitorar a temperatura e umidade ambiente, uma vez que esses parâmetros estão diretamente ligados com o desenvolvimento dos frangos. Sendo temperaturas e umidade ambiente diferentes das faixas recomendadas podem causar perda de massa, facilitar doenças e ocasionar até mortes. E a pressão barométrica ambiental de onde os animais estão alojados influenciam no valor do índice de entalpia de conforto térmico (SCHWEAN-LARDNER; CLASSEN, 2010).

Os sensores de gases (MQ2 – MQ135 e MQ-137) ajudam no monitoramento da qualidade do ar, a presença em excesso de dióxido de carbono pode ocasionar perdas na conversão alimentar e reduzir o conforto dos animais alojados nas granjas. Henn et al (2013) alertam que a cama dos frangos pode reter até 30% do CO₂ produzido pelos frangos, reforçando a necessidade da higienização correta entre os ciclos, assim sendo, a presença de sensor compatível pode ajudar a verificar a qualidade da ventilação do ambiente. A concentração de amônia acima de 30 ppm pode começar a dar problema na no ganho de peso, e até interferir no consumo de ração consequentemente nas taxas de conversão alimentar, contudo é citado por Vieira et al (2016) que se torna crítica a concentração do NH₃ quando os valores superam as 50ppm.

O sensor de iluminação BH1750 utiliza os dados de luminosidade do galpão juntamente com temperatura e umidade uma vez que a alta luminosidade pode causar stress nos frangos, principalmente nas últimas semanas de criação. E os frangos com até uma semana necessitam de maior temperatura e luminosidade (SCHWEAN-LARDNER; CLASSEN, 2010).

No tocante ao monitoramento da água ingerida pelas aves, a arquitetura IoT utiliza dois sensores: ST100 e Ntc10k de turbidez e temperatura, respectivamente. A turbidez da água possui dois pontos críticos sendo o primeiro a saúde física da própria ave e o segundo à saúde dos potenciais consumidores. A temperatura da água por sua vez é relacionada a hidratação da ave e da ajuda na regulagem corporal da ave (VIEIRA ET AL; 2016).

A seguir são descritas as características, faixas de medição, precisão e demais detalhes dos sensores utilizados na pesquisa:

BME280 (Sensor de umidade, temperatura e pressão barométrica): desenvolvido pela Bosch (BOSCH, 2020), esse sensor (Figura 15) monitora a umidade relativa do ar, temperatura do ambiente e pressão barométrica. Na pesquisa de Carranco et. al. (2017), a comparação entre o sensor BME280 e uma estação meteorológica profissional calibrada apresentou assertividade de 90% nas coletas sendo considerado pelos autores confiável e de um excelente custo-benefício.

Figura 15 - Sensor BME280

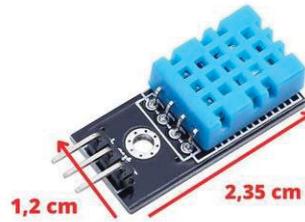


Fonte: o autor (2023)

O sensor opera em 5V, entre 0°C e 65°C, mensura a pressão em Pascal em uma faixa entre 300~1100hPa e com precisão $\pm 0,50$ hPa.

DHT11 (Sensor de temperatura e umidade relativa do ar): Trata-se de um sensor digital de baixo custo que afere a temperatura e umidade relativa do ar. Ele usa um sensor capacitivo de umidade e um termistor para medir o ar ao redor e envia um sinal digital no pino de dados. O sensor DHT11 (Figura 16) ainda possui um coeficiente de calibração o que confere estabilidade e precisão nos dados (JINKUANG, ET. AL., 2015; ADAFRUIT, 2019).

Figura 16 - Sensor DHT11



Fonte: o autor (2023)

O sensor consegue trabalhar entre 3.3V e 5V e consegue mensurar temperaturas entre 0°C e 50°C e umidade de 20% a 90% com precisão de $\pm 1^\circ\text{C}$ e $\pm 1\%$. O DHT11 serve como complemento ao BME280, com a captura dos mesmos dados de temperatura e umidade ambiente é possível fazer uma média dos valores capturados, aumentando a consistência dos dados (JINKUANG, ET. AL., 2015; ADAFRUIT, 2019).

MQ-2 (Sensor de gás inflamável e fumaça): com a capacidade de detectar a concentração de vários gases combustíveis e fumaça em um ambiente, tais como: GLP, Metano, Propano, Butano, Hidrogênio, Álcool, Gás Natural e outros inflamáveis. Quando o Detector de Gás MQ-02 detectar a presença e/ou a concentração de gases a saída digital fica em estado alto e é possível capturar o valor numérico da concentração do gás detectado (Figura 17).

Figura 17 - Sensor MQ-2



Fonte: o autor (2023)

O sensor auxilia no monitoramento do ar, identificando outros gases tóxicos ou nocivos e por ser um detector de fumaça também pode ser integrado a um sistema de detecção de chamas. O funcionamento do sensor é de 5V e mensura partículas entre 300-10.000ppm, por meio da biblioteca do fabricante é possível visualizar o quantitativo de gases de forma independente (JINKUANG, ET. AL; 2015; ADAFRUIT, 2019; ZHENGZHOU, 2020).

MQ-135 (Sensor de gases tóxicos - amônia, dióxido de carbono, benzeno, óxido nítrico e também fumaça e álcool.): O sensor (Figura 18) é capaz de detectar concentrações entre 10 e 300 ppm (partes por milhão) de amônia e álcool etílico e entre 10 e 1000 ppm de benzeno no ar. A sensibilidade do sensor pode ser ajustada através de um trimpot localizado atrás da placa. Quando algum gás ultrapassa o nível ajustado no trimpot, a saída digital do sensor (DO) fica em nível alto. Em concentrações normais, a saída digital permanece em nível baixo.

Figura 18 - MQ-135



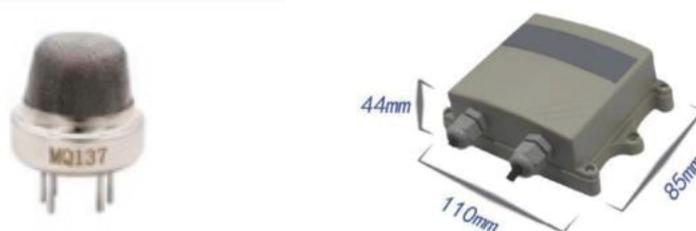
Fonte: o autor (2023)

O módulo também possui uma saída analógica (AO), que permite medir a concentração de gases no ar com precisão (ZHENGZHOU, 2020).

Para mapear qual gás o sensor está identificando os usuários podem converter a alteração da condutividade para corresponder ao sinal de saída da concentração de gás por meio de um circuito simples. O sensor opera em 5V com range de alcance de 10ppm a 1.000ppm e o fabricante garante funcionamento preciso de até 48 meses (ZHENGZHOU, 2020).

MQ-137 (Sensor de gás amônia NH₃.): O sensor de Gás MQ-137 tem capacidade de verificar a concentração de Gás Amônia entre 5 ppm a 500 ppm, o sensor opera entre -40°C e +80°C. Assim como o MQ-135 a sensibilidade do sensor pode ser ajustada através de um trimpot localizado atrás da placa. A Figura 19. Ilustra o sensor e sua case fornecida pelo fabricante.

Figura 19 - Sensor NH₃



Fonte: o autor (2023)

Para calcular o valor em partes por milhão do gás amônia se utiliza a seguinte equação:

$$\text{Valor NH}_3 = (\text{NH}_3 \text{ máx} - \text{NH}_3 \text{ min}) / (\text{Voltagem máx} - \text{Voltagem min}) * \text{valor da saída digital}$$

Desse sensor, especificamente, a voltagem é de 0-10V. Assim como o MQ-135, o tempo de vida útil com precisão de 95% com acurácia de ± 5 ppm são de 48 meses, segundo o fabricante (ZHENGZHOU, 2020). O valor tolerável não nocivo de gás amônia é de até 50ppm, sendo que o olfato humano não percebe a presença ou não com níveis de 20ppm ou menos (VIEIRA ET AL; 2016).

BH1750 (Sensor de luminosidade): Com o sensor é possível determinar a quantidade de luz (medida em lux) que está incidindo sobre o sensor, o sensor opera entre 3 e 5V com uma faixa de medição de 1 a 65.535 lux O sensor funciona corretamente com 95% de precisão em temperaturas entre -40°C e $+80^\circ\text{C}$ - Figura 20.

Figura 20 - Sensor BH1750



Fonte: o autor (2023)

Com os dados do sensor é possível regular a luminosidade de forma a ser mais eficiente possível e até economizar energia elétrica.

ST100 (Turbidez da água): O sensor é capaz de detectar e verificar a qualidade da água, fazendo a medição da turbidez, verificando os dados por meio de sinal digital ou. O ST100 opera em 3.3V a 5V, seu funcionamento consiste na emissão de uma luz infravermelha, imperceptível à visão humana, capaz de detectar partículas que estejam em suspensão na água. A Figura 21 ilustra o sensor

Figura 21 - Sensor ST100



Fonte: o autor (2023)

A faixa de alcance do sensor é de 0 - 1000NTU - Unidade de Turbidez Nefelométrica; que significa que o instrumento está medindo a luz dispersa da amostra

em um ângulo de 90 graus em relação à luz incidente. O valor máximo recomendado de turbidez é de 100 NTU (ZHENGZHOU, 2020).

NTC10k (Temperatura da água): O Sensor de Temperatura NTC10K MF58 é um sensor utilizado para medição de temperatura em locais úmidos e até submerso em líquidos devido ao encapsulamento. O funcionamento do NTC é explicado da seguinte forma: quando a temperatura do ambiente aumenta, a resistência do componente diminui, permitindo então realizar a leitura da temperatura conforme varia a resistência do NTC (Figura 22).

Figura 22 - Sensor NTC10K



Fonte: o autor (2023)

O sensor opera em 3V – 5V com precisão de $\pm 1^{\circ}\text{C}$ e sua faixa de medição varia entre -20°C e $+105^{\circ}\text{C}$, ou seja, consegue mensurar a temperatura da água também em estado sólido.

O Quadro 11 sintetiza as grandezas e unidades de medida dos sensores.

Quadro 11 - Descrição das grandezas e unidades de medida dos sensores

Modelo Sensor	Grandeza Física	Unidade de Medida	Símbolo da unidade.	Faixa de alcance
BME280	Temperatura ambiente	Graus celsius	$^{\circ}\text{C}$	-40 a +85
	Umidade do ar	Umidade relativa	%	0 a 100
	Pressão Barométrica	Hectopascal	hPa	300 - 1100
DHT11	Temperatura ambiente e Umidade do ar	Graus celsius e Umidade relativa	$^{\circ}\text{C}$ %	0 a 50°C 0 a 100%
MQ-2	Quantidade da substância	Partes por milhão	ppm	0 a 2000 e 0/1 binário
MQ-135	Quantidade da substância	Partes por milhão	ppm	10 a 1.000
MQ-137	Quantidade da substância	Partes por milhão	ppm	5 a 500
Bh1750	Intensidade luminosa	Lux	lx	1 a 65.535 0/1 binário
ST100	Turbidez da água	Unidade de Turbidez Nefelométrica	NTU	0 a 1000NTU
NTC10K	Temperatura da água	Graus celsius	$^{\circ}\text{C}$	-20°C a 105°C

Fonte: o autor (2023)

Todos os sensores supracitados possuem bibliotecas compatíveis com a Linguagem C e C++ e suporte para a IDE do Arduino.

3.3.3 Demais componentes eletrônicos

Para completar a montagem do hardware também foram utilizados os demais componentes:

Cabos Jumper: Utilizados para realizar as conexões elétricas entre os microcontroladores, sensores, atuadores e demais módulos IoT.

Cabos JST Estanhado: Fios de cobre para realizar a conexão dos componentes e conseguir maior alcance do que os cabos jumper.

Conector de fio tipo presilha: Utilizado para conectar os cabos jumper, cabo flat, e cabo PP duas vias aos componentes eletrônicos sem a necessidade de soldagem. Em outras palavras, utilizado para realizar as conexões elétricas entre os próprios cabos, sensores, atuadores e demais módulos IoT.

Módulo relé de 4 canais: Para realizar acionamentos elétricos de cargas até 127V, como lâmpadas, equipamentos eletrônicos, motores, ou usá-lo para fazer um isolamento entre um circuito e outro. O módulo é equipado com transistores, conectores, *leds*, diodos e relés.

A descrição dos componentes utilizados e seus respectivos preços estão no Apêndice 2.

3.3.4 Registrador de Dados (*Data Logger*)

O registrador de dados (*Data Logger*) é a ponte de interação entre o mundo físico (placa de desenvolvimento) e o mundo virtual do computador – *software* – (MABROUK ET. AL;2021). O desenvolvimento do *Data Logger* baseou-se em acoplar o NodeMCU a uma placa circuito em uma caixa case de forma que seja possível conectar os sensores/atuadores nas portas GPIO disponíveis, protegê-lo do ambiente externo dando independência ao protótipo para implementação em locais abertos, nesse caso, nos aviários.

Além dos sensores (descritos na subseção 3.3.2) e demais componentes eletrônicos (subseção 3.3.3), o Quadro 12 detalha os outros itens utilizados para a montagem do *Data Logger*.

Quadro 12 – Conjunto de materiais utilizados para montar o Data Logger

Item	Nome	Ilustração	Função/Objetivo	Preço estimado
1	Caixa de junção elétrica, à prova d'água, ip66		Vedar, organizar e facilitar a fixação dos componentes do data logger.	US\$ 8,00
2	Case de impressora 3D para montagem		Acondicionar os componentes do data logger para a fixação na granja	US\$ 5,00
3	Fonte 3V-5V		Fonte que regula a tensão para o funcionamento do microcontrolador.	US\$ 0,99

Fonte: o autor (2023)

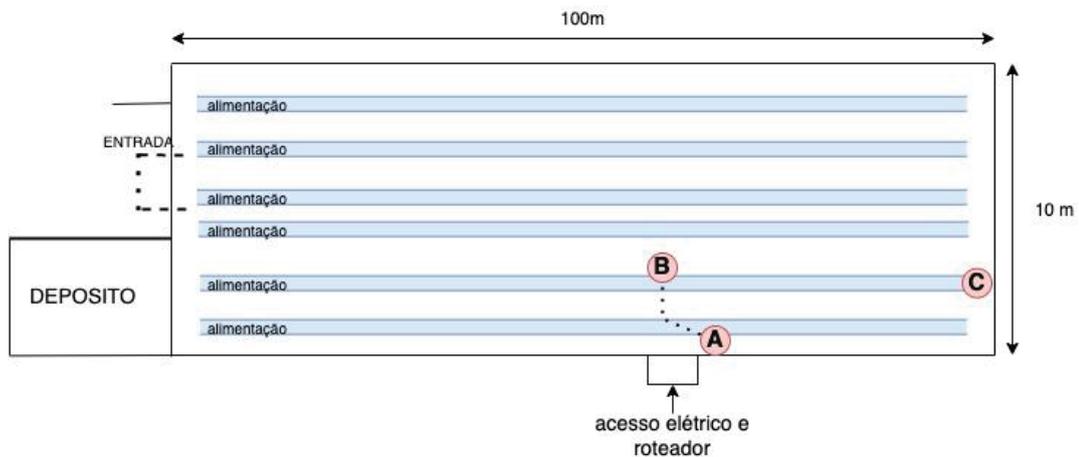
O Item 1 (Caixa de junção elétrica, à prova d'água, ip66) foi utilizado no aviário do participante que restringiu a utilização de equipamentos externos. Por ser uma caixa menor com relação ao Item2 e hermeticamente fechada e mais resistente, optou-se em utilizá-la nessa ocasião. O Item 1 também foi utilizado para o monitoramento da turbidez e temperatura da água, uma vez que precisou ser um data logger isolado por ter que ficar próximo a tubulação de água do aviário.

O preço estimado em fevereiro de 2023 do microcontrolador, sensores e demais componentes eletrônicos, em dólar americano (US\$), encontra-se no 2. Optou-se por colocar o preço em dólar uma vez que são componentes importados e os preços em reais (BRL) flutuam de acordo com o câmbio e o dólar mantém os preços mais estáveis.

O manual da montagem do *data logger* está no repositório público do pesquisador e pode ser acessado pelo *link*: <https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas> .

O croqui visto na Figura 23 mostra a localização dentro da granja onde foram instalados os registradores de dados para a coleta de dados.

Figura 23 - Croqui da granja com o registrador de dados



Fonte: o autor (2023)

Na posição (A) foi instalado o *Data Logger* com a caixa do “item 2” do Quadro 12, e estavam conectados diretamente os sensores BME280 e DHT11, também estava acoplado o módulo relé.

Os sensores: MQ2, MQ135, MQ137 e BH1750 ficaram na posição (B) e se conectaram ao Datalogger do (A) por meio de um cabo JST estanhado de aproximadamente 1 metro.

Próxima a parede e de um dos canos que chegam a água dos frangos foi instalado mais um Registrador de Dados. Utilizando a caixa “item 1” do Quadro 12 ficaram acondicionados, além do microcontrolador, os sensores ST100 e NTC10K para o monitoramento da água.

No Apêndice 3 foram anexadas fotos dos *Data Loggers* utilizados para coleta de dados.

3.4 TECNOLOGIAS E FERRAMENTAS

As tecnologias utilizadas para o desenvolvimento deste projeto são gratuitas e de código aberto, o que permite a liberdade de realizar as alterações necessárias em seu código-fonte. Além disso, não existem custos com licenças de *softwares* ou recursos adicionais. O desenvolvimento do Sistema Integrado proposto por essa tese exige a utilização de linguagens de programação que possibilite:

- Comunicação entre o *data logger* e a camada de aplicação (API);
- Manipulação do banco de dados NO-SQL

- A criação do servidor que aloque a API, Banco de dados e os *end-points* para o *front-end*
- Desenvolvimento do *front-end Web e Mobile* (iOS e Android)
- Interpretar e plotar gráficos interativos e análises estatísticas;

Devido à complexidade dos requisitos para a proposta de arquitetura serão necessários um conjunto de ferramentas, bibliotecas e linguagens de programação com documentação disponível em repositórios oficiais.

Conforme indicado na seção 2.4 - “Tecnologias para desenvolvimento”, especificamente na seção de Linguagem de programação para *web*, *mobile* e embarcados a linguagem C e C++ possibilitam a criação de algoritmos para a comunicação entre os microcontroladores, sensores, atuadores e o servidor *web*. Sendo assim, os procedimentos de coleta de dados são realizados nessa camada; onde o microcontrolador envia, por meio de requisição HTTP, os dados dos sensores no formato JSON para a API REST, que, por sua vez, faz o serviço de comunicação entre NodeMCU e servidor tanto para consulta como para a persistência dos dados no banco de dados.

Para o desenvolvimento da camada *server-side* e *client-side* da aplicação a linguagem a ser utilizada é o *JavaScript* (JS) por meio do *Node.js* no *back-end* e o *React.js* e *React Native* para o *front-end*:

- 1- No *back-end* o *JavaScript* possibilita o desenvolvimento completo de uma API REST por meio do *Node.js* um compilador e interpretador de scripts *JavaScript*. O resultado dessa interpretação resulta em um código de máquina *server-side*. Ademais, permite a implementação de um projeto leve e em servidores abertos com qualquer Sistema Operacional (SO), diminuindo o custo com *hardwares* (Máquinas servidores) e *Softwares* - Licenças de servidores Windows (TILKOV; VINOSKI, 2010; NODE, 2019). Portanto, para o desenvolvimento da nossa API utilizaremos o *Node.js*, o qual vai realizar a comunicação entre *client-side* e *server-side*.
- 2- Para o *front-end* serão utilizadas duas bibliotecas do *JavaScript* ambas *open source* e utilizadas para construir as interfaces. A primeira, chamada de *React.JS* e voltada para o desenvolvimento *Web*. E vai apresentar as telas de navegação *Web* do sistema. Para o desenvolvimento *Mobile* será utilizada a biblioteca *React Native* com o *framework* “*Expo*”. O qual permite

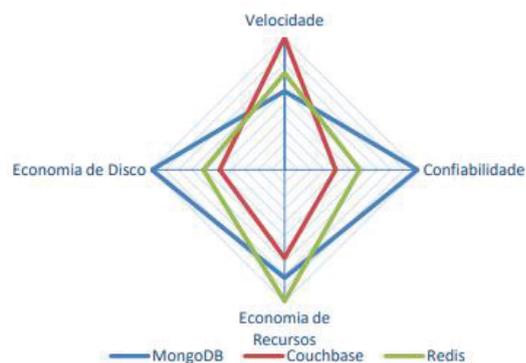
o desenvolvimento simultâneo para os sistemas operacionais Android e iOS. Além de ser gratuito, com o Expo não é necessário um computador da marca Apple ® para o desenvolvimento e testes de aplicações iOS.

No tocante a banco de dados, o modelo escolhido para o desenvolvimento foi o NO-SQL (*Not Only SQL*) – banco de dados não relacional. Esse modelo tem como características a arquitetura flexível, a alta velocidade, gravação de dados e escalabilidade, que os tornam muito compatíveis com aplicações de alto desempenho e que priorizam a performance (MEIER; KAUFMANN, 2019).

De acordo com Matos e Soares (2017) e Hoddie e Prader (2020) a *internet das coisas* geralmente está atrelada a aplicações que funcionam em tempo real e necessitam de grande velocidade de manipulação de dados, uma vez que nesse ambiente há uma grande quantidade de dados de múltiplas fontes sendo trafegados.

Existem diversos Sistemas Gerenciadores de Banco de Dados NOSQL, cada um com suas características em relação a velocidade de consulta, confiabilidade e estabilidade, economia de armazenamento e economia de recursos de máquina (MEIER; KAUFMANN, 2019). Matos e Soares (2017) desenvolveram um estudo e avaliaram por meio de comparação três dos mais robustos, gratuitos e amplamente utilizados bancos de dados No-SQL, conforme a Figura 24 ilustra:

Figura 24 – Comparação entre os bancos de dados NOSQL



Fonte: Matos e Soares (2017)

Os autores relataram que o *MongoDB* é recomendado para aplicações IoT onde há restrições de confiabilidade e segurança bem como de espaço para armazenamento em disco, além de ter um resultado satisfatório nas demais métricas; o *Couchbase* é recomendado para aplicações que demandam de alta velocidade sem limitações de processamento e armazenamento e por fim o *Redis* é recomendado

para aplicações onde há restrições de consumo de processamento, tendo um resultado mediano nas demais métricas avaliadas (MATOS; SOARES, 2017).

Devido a necessidade de segurança dos dados (confiabilidade) e de economia de armazenamento pois serão coletados uma grande quantia de dados o banco de dados escolhido para o projeto é o *MongoDB*, pois o mesmo também tem um desempenho satisfatório em relação a velocidade e economia de recursos, principalmente quando comparado ao modelo relacional tradicional de banco de dados SQL.

Para a visualização dos gráficos, indicadores e análises estatísticas serão utilizadas bibliotecas que fornecem aporte para o desenvolvimento de *dashboards* interativos e com filtros personalizáveis para o usuário final.

O Quadro 13 lista as tecnologias utilizadas no desenvolvimento da arquitetura IoT proposta.

Quadro 13 - Tecnologias utilizadas no desenvolvimento

Tecnologia	Logo	Utilizada para:	Autor/Fonte
C, C++		Comunicação entre os microcontroladores, sensores e atuadores com a API REST para o armazenamento dos dados no <i>MongoDB</i>	KERNINGHAN B.; RITCHE, D. C LANGUAGE. ANSI (1988). ISO. ISO/IEC 14882: C++ (2017).
Node.js		Desenvolver a API REST e integrar a arquitetura IoT	NODE.JS - OPENJS FOUNDATION (2020).
React.js		Desenvolvimento da plataforma web	REACTJS (2019)
React Native		Desenvolvimento da plataforma mobile (iOS e Android).	REACT NATIVE DOCUMENTATION (2019)
MongoDB		Sistema de Gerenciamento de Banco de Dados SGBD	MONGODB FOUNDATION (2014)

Fonte: o autor (2023)

Com as tecnologias definidas, a seguinte subseção apresenta os procedimentos de desenvolvimento da arquitetura.

3.5 PROCEDIMENTOS DE DESENVOLVIMENTO

A construção desta seção é sequência das seções de Materiais e Tecnologias e Ferramentas, subseções 3.3 e 3.4 respectivamente.

Detalhando o passo a passo de como foi construído o sistema IoT, tanto na perspectiva do hardware (parte física) quanto na perspectiva de software (parte digital). A subseção tem como aporte os conceitos estabelecidos no referencial teórico, como: Bancos de dados não relacionais (No-SQL), Estrutura e formato de dados (JSON), API REST e as Linguagens de Programação para o desenvolvimento.

3.5.1 Procedimentos de desenvolvimento do sistema embarcado

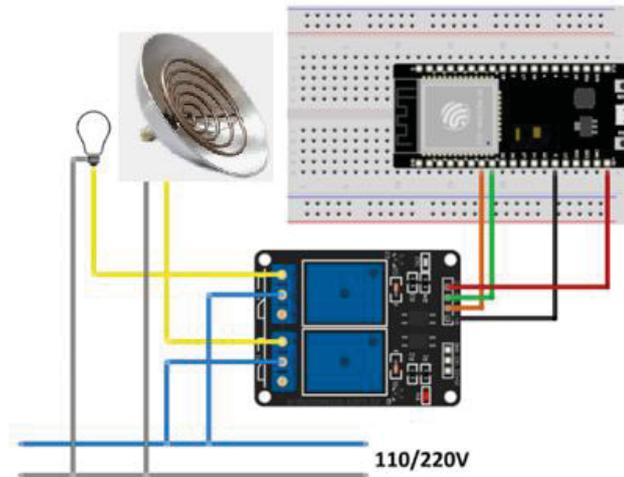
Com o Registrador de dados montado (subseção 3.3.4), é possível digitalizar dados de um ambiente físico (real) e transferi-los para o ambiente virtual (software). Os dados extraídos são advindos de duas formas: analógicos (valores quantitativos) e digitais (do tipo binário - 0/1). Por exemplo, o sensor MQ-137 com a saída analógica de dados é possível quantificar o valor de ppm do gás amônia presente no ambiente. E se utilizado a leitura dos dados digitais, o sistema identifica se há ou não a presença do gás no ambiente.

Assim sendo, para extrair os dados é requerido um algoritmo (conjunto de instruções e técnicas computacionais aplicadas em um determinado contexto). No caso dessa pesquisa, a extração dos dados sensorizados faz uso da Linguagem C e C++; através da IDE Arduino que compila o script no microcontrolador.

São utilizadas as bibliotecas dos fabricantes dos sensores, assim, os dados são coletados separadamente de acordo com sua documentação oficial, garantindo a fidedignidade dos dados uma vez que a comunicação entre sensor e microcontrolador ocorre em frequência elétrica conhecida e pré-determinada – A lista das bibliotecas utilizadas são encontradas no repositório publico do pesquisador no endereço: <https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas> .

No tocante aos atuadores, optou-se por módulos relés pois é um facilitador no acionamento de cargas por meio de sinais digitais que podem ser enviados ao microcontrolador remotamente. A Figura 25 exemplifica a conexão dos módulos relés conectados a um sistema de iluminação com o microcontrolador.

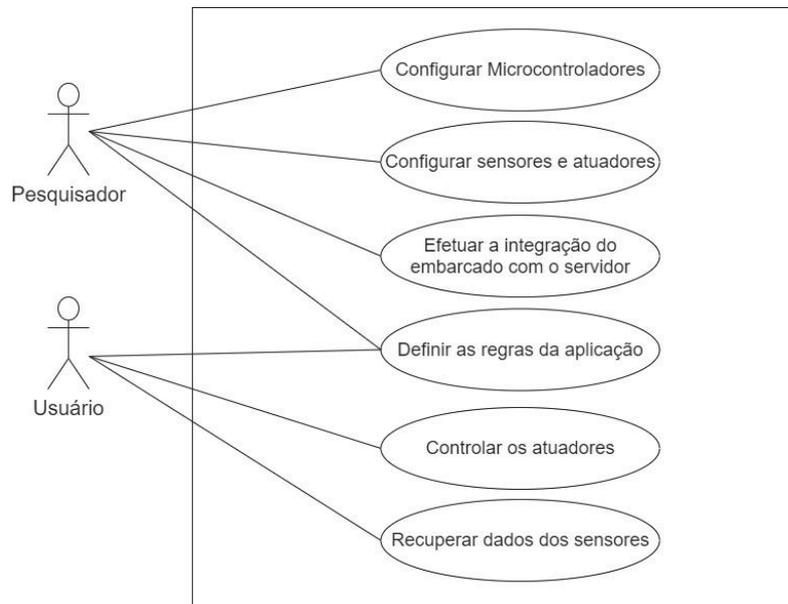
Figura 25 – Exemplo de conexão dos módulos relés com microcontrolador



Fonte: o autor (2023)

A Figura 26 apresenta o diagrama de caso de uso do sistema (levando em consideração o Data Logger).

Figura 26 - Diagrama de caso de uso do sistema embarcado



Fonte: o autor (2023)

O pesquisador compilou o script via ArduinoIDE no microcontrolador e definiu as rotas para envio dos dados. Também configurou o servidor para que o banco de dados seja acessado via requisições HTTP advindas dos microcontroladores.

No tocante às regras do negócio, pesquisador e usuário (avicultor) apoiados na literatura e nas características da espécie dos frangos da granja definem as regras e parâmetros que os sensores devem monitorar e atuar. Essas regras então são

convertidas em algoritmos que rodam no servidor para que seja possível consultar pela aplicação *web/mobile*. O usuário pode controlar os atuadores e acompanhar a leitura dos sensores em tempo real no lado do cliente (utilizando um computador ou o próprio *smartphone*) – Uma vez que o *Data Logger* é o módulo de captura e envio dos dados e o *back-end* são os módulos de persistência e recuperação dos dados.

3.5.2 Desenvolvimento do back-end

Para a construção do servidor foi utilizada a biblioteca “*Express*” do NodeJS que fornece um conjunto robusto de recursos para configuração de servidor para integração de aplicações servidor/cliente. A biblioteca possui um sistema de rotas; possibilita o tratamento de exceções dentro da aplicação; permite a integração de vários sistemas e facilita a criação de páginas *web/mobile* para suas aplicações; além de gerenciar diferentes requisições HTTP com seus métodos.

No tocante a modelagem e manipulação no banco de dados *MongoDB* utilizou-se a biblioteca “*Mongoose*”; a qual manipula objetos (JSON) de um banco de dados não relacional e permite realizar as operações e manipulações do mesmo, por exemplo o CRUD (*Create, Read, Update, Delete*).

A comunicação do servidor com o *front-end* e com o sistema embarcado é mediada pela API que é responsável por enviar as requisições ao servidor utilizando os métodos HTTP.

3.5.3 Procedimentos do desenvolvimento front-end Web e Mobile

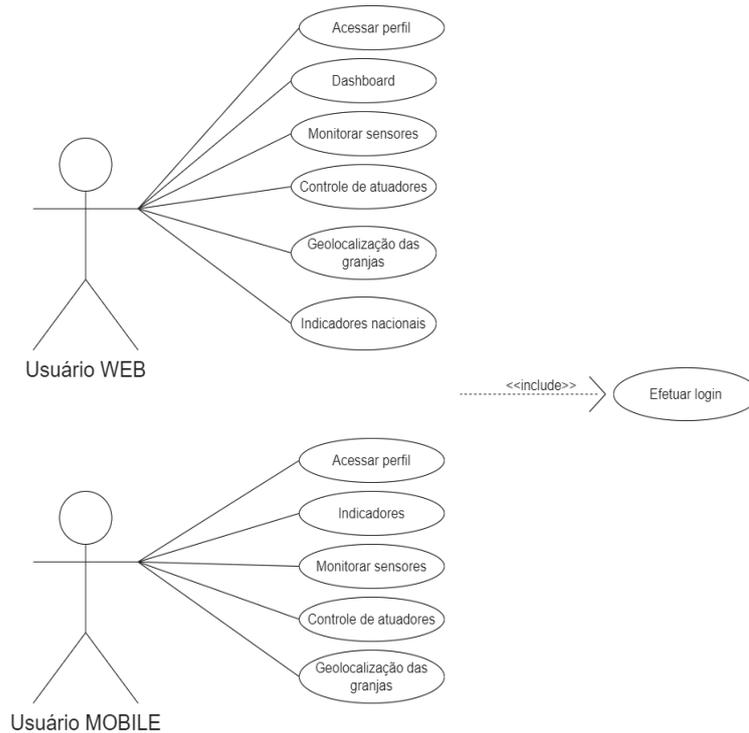
O desenvolvimento da etapa de *front-end* engloba os tópicos de experiência de usuário (UX), então, os procedimentos foram divididos em três etapas, a saber:

- Criar um protótipo funcional com elementos teóricos da cadeia produtiva de frango de corte atendendo aos requisitos de UX.
- Ajustar o protótipo de acordo com o mapeamento das necessidades e prioridades advindas dos dados coletados da entrevista
- Integrar o *front-end* com a API

Foram utilizadas as bibliotecas *React Native* para o desenvolvimento da interface *mobile* (Android e iOS) e da biblioteca *React* para a interface *web*.

A Figura 27 apresenta o diagrama de caso de uso da aplicação *web* e *mobile*

Figura 27 - Diagrama de caso de uso das aplicações web e mobile



Fonte: o autor (2023)

Após efetuar o *Login* na aplicação o usuário consegue ter acesso às suas informações, sendo a geolocalização de seus aviários, dados compartilhados de produção e rendimento com outros produtores cadastrados na plataforma, dados em tempo real de sensores, controle dos atuadores, e vínculo com indicadores nacionais do agronegócio.

3.6 PROCEDIMENTOS DE AVALIAÇÃO

As etapas de todo o processo da avaliação da arquitetura IoT estão sintetizadas no Quadro 14.

Quadro 14 - Etapas de coleta dos dados

Etapa	Objetivos	Materiais ou procedimentos
Formulação do Questionário	Formular e organizar as questões com o objetivo de mapear as dimensões de avaliação UX com base na literatura correlacionada aos temas.	Suporte teórico (Quadro 13)
Aplicação do pré-teste	De forma a consolidar o questionário, enviá-lo para potenciais respondentes integrantes da população pesquisada de modo a identificar falhas na redação, complexidade de questões, desnecessidade de questões e demais aspectos (GIL, 2009)	Ação do pesquisador

Ajustes e correções	Correções e alterações do questionário conforme o resultado do pré-teste para o envio definitivo.	Ação do pesquisador
Envio dos questionários	Contatar os respondentes por e-mail para o envio individualizado do questionário.	E-mail (<i>Google Forms</i>)
Tratamento dos dados coletados	Aplicação dos testes estatísticos aos dados obtidos por meio do questionário.	<i>Software R</i>

Fonte: o autor (2023)

A coleta de dados se deu por meio de questionário, uma vez que há a garantia do anonimato dos participantes, e que podem responder no momento que lhe for mais conveniente, além de proteger os respondentes da influência das opiniões e aspectos pessoais do pesquisador (GIL, 2008).

Houve a participação de 23 respondentes, sendo 18 proprietários de aviário, 1 gestor de cooperativa e 4 auxiliares de produção que estão no dia a dia da granja. Todo o processo durou cerca de quatro semanas (entre novembro de 2022 e janeiro de 2023).

Antes da aplicação do questionário foi realizado um pré-teste, com seis participantes, sendo: i) dois profissionais, um da área de *Design* e outro da Engenharia Agrônômica; ii) um funcionário de cooperativa responsável pelo setor de TI; e iii) três avicultores do universo pesquisado. Junto com o formulário de avaliação foi apresentada a interface da aplicação, permitindo que os respondentes conseguissem contribuir para o ajuste do questionário.

O questionário para avaliação (Apêndice 4) foi dividido em sete dimensões distribuídas em 13 questões (duas abertas e onze fechadas), nas quais os respondentes indicaram o grau de concordância. A estrutura do questionário, com suas respectivas dimensões, pode ser verificada no Quadro 15.

Quadro 15 – Dimensões de avaliação UX da arquitetura IoT

Dimensão	Descrição	Suporte Teórico
Classificação do respondente	A seção visa classificar o respondente do questionário com relação a sua categoria (Produtor, agroindústria ou outros).	-
Estético/Visual	Identificar se a interface da aplicação é intuitiva e amigável	MAIA; FURTADO (2014); REGADO; PINTO (2021)
Estímulo/Engajamento	Verificar se o usuário se identifica com a aplicação	RIVERO; CONTE (2017); MARQUES ET. AL. (2019);

		REGADO; PINTO (2021)
Usabilidade	Detectar se o usuário consegue utilizar as funcionalidades de modo facilitado	LOPES; VALENTIM (2019); FERREIRA; SANTOS; PORTELA (2021)
Aprendizado	Constatar a dificuldade de aprendizado do usuário ao utilizar a aplicação	FERREIRA; SANTOS; PORTELA (2021); OLIVEIRA; NÓBREGA (2021)
Utilidade	Mensurar a utilidade da aplicação e se ela possui valor agregado pelo serviço oferecido	MAIA; FURTADO (2014); MARQUES ET. AL. (2019); REGADO; PINTO (2021)
Experiência geral	Identificar o sentimento geral do usuário sobre a aplicação	MARQUES ET. AL. (2019); OLIVEIRA, F. L.; NÓBREGA (2020)

Fonte: o autor (2023)

Os dados coletados foram tabulados e analisados utilizando o *Software R*, a planilha com as respostas pode ser consultada no apêndice 6 – dados tabulados da avaliação. As análises dos dados coletados foram realizadas conforme protocolo de avaliação visto no Quadro 16.

Quadro 16 – Protocolo de análise da avaliação

Etapa	Procedimento	Testes	Objetivo
1	Estatística descritiva: Médias, mediana, desvio padrão, mínimos e máximos.	Médias, desvios padrão, mínimos e máximos	Apresentar uma visão geral das características dos respondentes em relação às dimensões avaliadas.
2	Classificação dos dados: Frequências, percentuais e gráficos	Gráfico boxplot e gráfico de barra	Categorizar e visualizar graficamente as variáveis que representam as dimensões exploradas.
3	Análise de correspondência múltipla (ACM)	Cálculo de massas, inércias e gráficos das categorias	Identificar as dimensões e os agrupamentos das respostas. Visualizar graficamente as relações entre os respondentes e suas impressões sobre a arquitetura IoT.

Fonte: o autor (2023)

O script das análises da avaliação está no Apêndice 5, a versão do *Software R* utilizada para as análises é a 4.2.2 e a do *R Studio* é 12.0.353(2022), também foi utilizado o *Microsoft Excel 2016* para armazenar os dados. E a base de dados analisada no Apêndice 6.

A etapa 1 é a estatística descritiva e tem o objetivo de fornecer uma visão geral das características das dimensões avaliadas. As medidas das estatísticas descritivas apresentadas são as médias, mediana, desvio padrão, valor mínimo e valor máximo de cada variável componente do questionário.

Na classificação dos dados (etapa 2) foram apresentadas as tabelas de contingência e gráficos descritivos (histogramas). Essa etapa é também conhecida como análises exploratórias, descritas como um conjunto de técnicas para tratamento de dados, que não utiliza fundamentação matemática rigorosa e permite um primeiro contato com a informação disponível (HAIR, ET. AL; 2009).

Na terceira etapa foi realizada a ACM uma vez que o número de respostas não ultrapassou 30 observações e os dados eram de origem categórica (HAIR ET AL; 2009).

Segundo Abdi e Valentin (2007) e Hair et al. (2009) a ACM torna-se um método confiável e recomendado uma vez que seu princípio é produzir um gráfico bidimensional no qual os objetos dentro da mesma categoria são plotados próximos e objetos de diferentes categorias são plotados distantes, dessa forma não necessitando de uma quantidade maior de dados (ABDI; VALENTIN, 2007).

Para esclarecimentos, Abdi e Valentin (2007) ressaltam que método da ACM requer que sejam identificados os agrupamentos (categorias) que exercem maior influência nos dados, ou seja, realizar o cálculo de massas e inércias das categorias.

Assim é possível gerar o mapa (gráfico) das coordenadas principais de linha e/ou de coluna nas dimensões com maior inércia - que é a medida de maior contribuição para o conjunto.

Logo após esse primeiro mapa, há então a criação de outro mapa onde é possível identificar associações entre categorias de objetos, possibilitando identificar agrupamentos exibidos de maneira simultânea (HAIR ET AL. 2007).

3.7 PROCEDIMENTOS DE VALIDAÇÃO

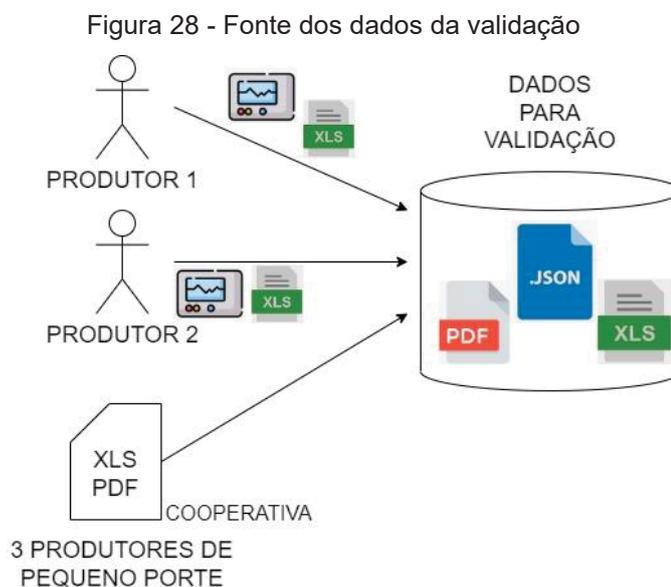
Para consolidar a proposta desta tese e validar estatisticamente o funcionamento, qualidade e eficiência da arquitetura foram utilizados testes estatísticos com o cruzamento de dados anteriores e posteriores à implementação da arquitetura IoT.

A fonte primária dos dados foi a da própria pesquisa, com a coleta direta dos dados usando o *Data Logger* nas granjas dos produtores. Foram utilizados também dados de relatórios de produção do PRODUTOR 1 e PRODUTOR 2, fornecidos pela cooperativa de todo o ano de 2022.

Além disso, o PRODUTOR 1 forneceu dados sensorizados de temperatura, umidade e dióxido de carbono de seu sistema comercial de automação do mesmo período em que o *Data Logger* do pesquisador esteve implementado em sua propriedade, permitindo uma comparação dos dados entre os sistemas.

A Cooperativa, que ambos produtores pertencem, também forneceu dados de produção de outras três granjas de localização próxima às do PRODUTOR 1 e PRODUTOR 2, os dados da cooperativa são do ano de 2022.

Esses dados são de aviários de pequeno porte, com características semelhantes ao do PRODUTOR 2. Os dados foram descaracterizados e entregues ao pesquisador em formato “.xls” (planilhas) e “.pdf”. Na Figura 28 pode ser observada as fontes dos dados para a validação.



Fonte: o autor (2023)

Ambos os produtores (1 e 2) tiveram dados do *Data Logger* enviados direto ao servidor em formato JSON via rede *wi-fi*. Os dados em formato XLS e PDF da cooperativa foram fornecidos por meios digitais.

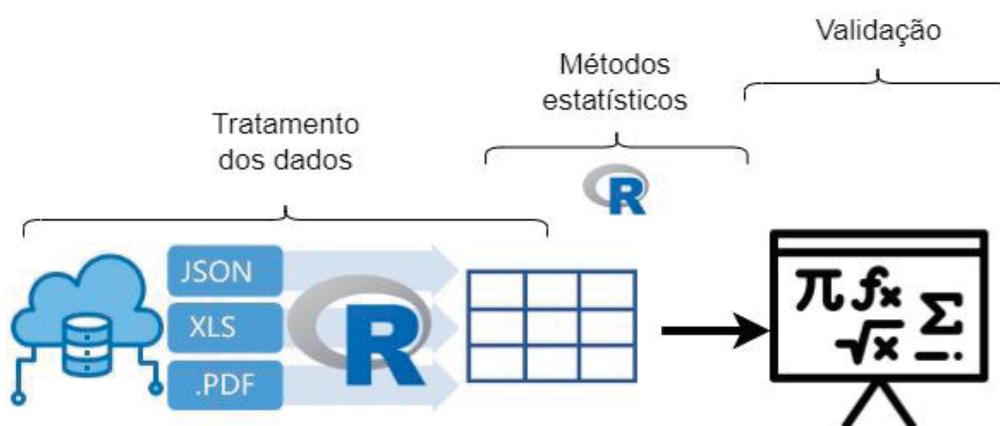
Os dados foram coletados entre 02 de novembro de 2022 até 12 de janeiro de 2023, totalizando pouco mais de 70 dias. O processo de implementação e testes

iniciou no no quarto lote, contudo só foram utilizados os dados coletados pelo *Data Logger* no quinto lote do ano.

Os dados foram armazenados e organizados em um repositório único para seguir com as análises, todas as etapas de tratamento dos dados e análises foram utilizando o Software R na versão 4.2.2 e o R Studio 12.0.353 (2022).

A Figura 29 sintetiza o processo de tratamento, aplicação dos métodos estatísticos e então a análise dos resultados para verificar ou não a validade da arquitetura.

Figura 29 – Fluxo dos dados para a validação



Fonte: o autor (2023)

O processo de tratamento dos dados teve por objetivo tabular e organizar os dados de modo a garantir a consistência e resultados corretos oriundos das análises.

Nos relatórios de produção foram identificadas as informações relevantes para a análise, sendo: o período de produção (datas), o código da granja, identificador do lote, o Quadro 17 detalha as demais informações que se tornaram as variáveis utilizadas nas análises.

Quadro 17 - Descrição da base de dados

Fonte dos dados	Nome da variável na base de dados	Descrição da variável	Polaridade	Unidade de medida
Relatórios cooperativa	Total_aves	Nº de aves no galpão	-	Unidade
	Mortalidade	Nº de aves mortas	Quanto menor melhor	Unidade
	Refugo	Nº de aves retiradas	Quanto menor melhor	Unidade
	Viabilidade	Diferença do total de aves do início com o final	Quanto maior melhor	%

	Peso_med	Peso médio dos frangos	Quanto maior melhor	g
	Gpd_medio	Ganho peso diário (média semanal)	Quanto maior melhor	g
	Conv_alimentar	Índice de conversão alimentar	Quanto menor melhor	-
	Iep	Índice de eficiência produtiva	Quanto maior melhor	-
	Pag_ave	Pagamento por unidade de frango entregue	Quanto maior melhor	R\$
	Temp_med	Temperatura média no galpão	-	°C
	Umid_med	Umidade média no galpão	-	%
Data Logger	Bme_280_t dht11_t	Temperatura Ambiente	-	°C
	Bme_280_umi Dht11_umi	Umidade Relativa	-	%
	Bme_280_p	Pressão Barométrica	-	Pa
	Mq-2	Gases inflamáveis e fumaça	Quanto menor melhor	ppm
	Mq-135	Dióxido e Monóxido de carbono	Quanto menor melhor	ppm
	Mq-137	Gás amônia (NH3)	Quanto menor melhor	ppm
	Bh1750	Luminosidade	-	lux
	St100	Turbidez da água	Quanto menor melhor	NTU
	Ntc10k	Temperatura da água	-	°C
Data Logger	Bme_280_t dht11_t	Temperatura Ambiente	-	°C
	Bme_280_umi Dht11_umi	Umidade Relativa	-	%
	Mq-135	Dióxido e Monóxido de carbono	-	ppm

Fonte: o autor (2023)

Além das variáveis dos relatórios de produção, as análises contaram com mais 12 variáveis de sensores, divididas entre os data loggers.

A base de dados utilizada na validação está disponível em formato “.csv” no repositório público da tese que pode ser acessado pelo link: [“https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas”](https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas) e também no Apêndice 7.

Todas as análises seguiram o protocolo de análise do Quadro 18.

Quadro 18 - Protocolo de análise da validação

Etapa	Procedimento	Objetivo
1	Estatísticas descritivas (Médias, desvio padrão, máximos e mínimos)	Descrever o conjunto de dados
2	Shapiro-Francia e Teste de Bartlett	Checar a normalidade e homogeneidade do conjunto de dados respectivamente
3	Teste t de <i>student</i> ou U de Mann-Whitney	Verificar se há diferenças significativas entre dois grupos
4	ANOVA ou H de Kruskal-Wallis	Verificar diferenças significativas entre mais de dois grupos

Fonte: o autor (2023)

Para o protocolo de validação foi considerado o nível de significância abaixo de 0,05. E é necessário pontuar que o Quadro 18 deve ser interpretado da seguinte maneira:

As estatísticas descritivas (etapa 1) foram detalhadas aviário por aviário e de todos os aviários juntos, ou seja, foram descritos os dados gerais de cada aviário no ano de 2022 e uma média geral entre esses aviários, baseando-se nos relatórios de produção da cooperativa.

Na etapa 2 são realizados testes para verificar a normalidade e homogeneidade do conjunto de dados, esses procedimentos são feitos em cada grupo separado e também para toda a base de dados.

Se na etapa 2 for confirmado que os dados seguem uma distribuição normal e variância homogênea na etapa 3 utiliza-se o Teste t de *student*, caso uma das duas premissas não seja atendida utiliza-se o U de Mann-Whitney para uma distribuição não normal e/ou não homogênea.

Os testes estatísticos da etapa 3 foram utilizados para realizar três cruzamentos de dados, os quais consideraram a variável monetária “**pag_ave**” como variável dependente. A escolha dessa variável se dá pela importância aos produtores, que na última linha o que importa é se toda a tecnologia foi convertida em retorno financeiro. O Quadro 19 aborda quais dados foram cruzados e seus respectivos cruzamentos:

Quadro 19 - Cruzamento de dados

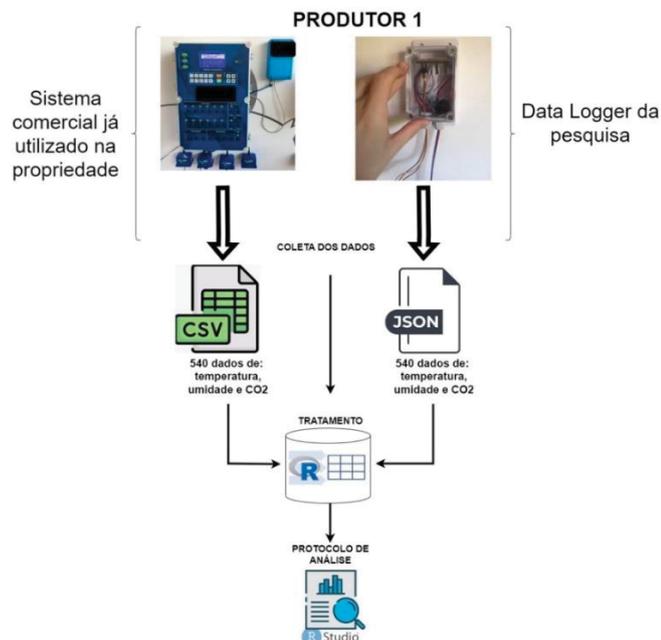
Nº	CRUZAMENTO DE DADOS REALIZADO	JUSTIFICATIVA
(1)	Lote sensoriado x Lotes não sensoriados somente do PRODUTOR 2	Verificar se houve diferença estatisticamente significativa entre os resultados do lote com a arquitetura IoT implementada comparada a mesma granja em lotes em que o pequeno produtor não tinha aporte tecnológico de monitoramento da granja
(2)	Resultados sensoriados do PRODUTOR 2 x Resultados não sensoriados de todos os produtores de pequeno porte	Validar a eficiência da arquitetura somente entre pequenos produtores, cruzando dados do lote que teve suporte tecnológico da arquitetura IoT com todos os demais que não tiveram.
(3)	Sistema próprio de sensoriamento do PRODUTOR 1 x <i>Data Logger</i> da pesquisa instalado no PRODUTOR 1	Validar a qualidade da dos dados coletados do data logger comparado a um sistema comercial, calibrado, e utilizado em grande escala por médios e grandes produtores.

Fonte: o autor (2023)

A primeira validação busca verificar se há diferenças significativas nas aferições de temperatura, umidade e dióxido de carbono (qualidade da ventilação) do sistema comercial já utilizado na granja com a arquitetura IoT de baixo custo proposta nessa tese.

Com o *Data Logger* implementado próximo aos sensores da empresa, foram coletados dados a cada 2 horas durante todo o período do lote, desde a chegada das aves com dois dias de vida até o dia do abate (neste lote 45 dias), totalizando 540 aferições. A Figura 30 explica o processo de validação da qualidade dos dados coletados na propriedade do PRODUTOR 1.

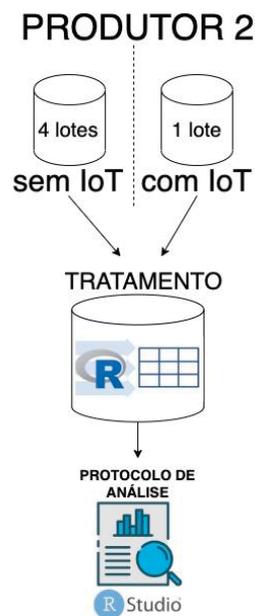
Figura 30 - Sequência do protocolo de análise



Fonte: o autor (2023)

O segundo cruzamento de dados é a comparação entre dados somente do PRODUTOR 2. Para verificar se houveram diferenças estatisticamente significativas, ou seja, se a implementação surtiu efeito positivo (melhorando a eficiência e remuneração). A Figura 31 ilustra o cenário para as análises.

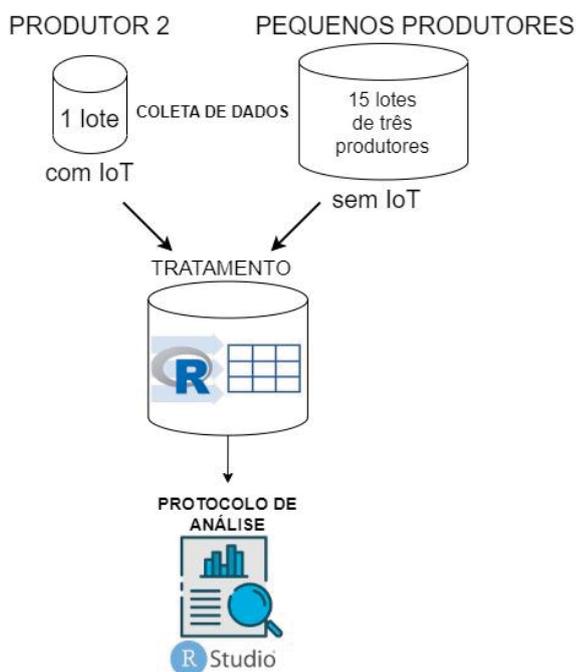
Figura 31 - Cruzamento de dados somente do PRODUTOR 2



Fonte: o autor (2023)

Para o terceiro cruzamento de dados, foram utilizados dados fornecidos pela cooperativa dos lotes do ano de 2022 dos três pequenos produtores e comparados com desempenho do lote do PRODUTOR 2 que utilizou a arquitetura IoT, conforme Figura 32.

Figura 32 - Cruzamento de dados do PRODUTOR 2 x pequenos produtores



Fonte: O autor (2023)

Em todos os cruzamentos de dados foram utilizadas análises bivariadas comparando dois grupos. Sendo que o primeiro cruzamento a comparação do grupo “Sistema comercial” x “Data logger da pesquisa”.

O segundo cruzamento de dados também comparou dois grupos, os dados dos “aviário antes da pesquisa (sem IoT)” x “os dados com a pesquisa (com IoT)”. E o último cruzamento de dados foi do “lote com IoT” x “lotes sem IoT” todos com dados advindos de pequenos produtores.

O script R das análises de validação pode ser encontrado no Apêndice 8 e também no repositório público da tese (<https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas>).

4 MAPEAMENTO DAS NECESSIDADES E PRIORIDADES

De modo a sintetizar as principais abordagens nas pesquisas sobre IoT na 1.3 foram sintetizados um apanhado de propostas encontradas na literatura acadêmica.

Para explorar as o contexto prático empresarial/comercial, foi adotada a técnica de entrevistas, a qual, além de verificar quais as necessidades e prioridades dos atores da cadeia produtiva de frango de corte com relação ao auxílio da tecnologia IoT em seus processos, também se verificou as ofertas de soluções encontradas no mercado, bem como as lacunas tecnológicas que ainda não foram exploradas por empresas privadas (**OBJETIVO ESPECÍFICO A**).

No tocante às questões das relações dos produtores com a agroindústria, os entrevistados citam que recebem consultorias de técnicos agropecuários, veterinários e zootecnistas com informações referente às aves, doenças, rações, bem-estar animal e outras informações técnicas de produção.

Os entrevistados citam que a parceria com a agroindústria, também citada como integração, ocorre mediante a prestação de serviços, as agroindústrias fornecem frangos com até três dias de vida aos avicultores, que cedem sua mão de obra e o espaço físico (granja) para realizar o trabalho de crescimento e engorda. O ciclo completo leva de 43 a 49 dias. Para a remuneração, os elementos elencados pelos entrevistados foram: **índice de mortalidade, taxa de conversão alimentar, ocorrência de doenças** e a inspeção após o abate que calcula o **tamanho e peso médio dos frangos por lote**. Assim, a remuneração para o produtor, depende do bom manejo do aviário.

Os elementos supracitados para a remuneração têm relação direta com as questões práticas de produção e manejo, sendo a maior oportunidade da implementação de dispositivos IoT, uma vez que as fases de crescimento e engorda dos frangos são as mais relevantes para aumento de produtividade e lucratividade para o produtor. São três etapas que acontecem dentro do aviário, que são destacadas a seguir:

A fase de alojamento é a primeira etapa, citada como mais crítica uma vez que as aves chegam ao galpão com no máximo três dias de vida e precisam ser divididas em círculos de proteção, onde ficam pelos primeiros 28 dias, pois não possuem a capacidade de produzirem seu próprio calor através dos nutrientes ingeridos. **Sendo totalmente dependentes fontes de aquecimento (campânulas)**

para controle da temperatura correta (cerca de 32°C para a primeira semana, 30° para a segunda semana, 28° na terceira semana e 26° na quarta semana) e umidade do ar variando entre 50 e 70% para todas as etapas nos círculos de proteção. Se a temperatura da cama e do ar ambiente estiverem baixas, a temperatura corporal interna das aves irá cair, levando à aglomeração dos mesmos, diminuição da ingestão de ração e água, menor crescimento e suscetibilidade a enfermidades (RODRIGUES; ET. AL, 2014). É citado também que nessa fase não é recomendado o uso de ventilação natural ou mecânica, e gradualmente os círculos de proteção devem ser expandidos de tamanho uma vez que as aves triplicam de peso nos dez primeiros dias.

Os fatores mais importantes para garantir um bom desenvolvimento é o consumo de ração específica para essa fase, e a **qualidade da água oferecida**, a qual deve ser limpa, fresca (entre 20 e 24°C) e clorada, com bebedouros de fácil acesso. **Água quente reduz a ingestão de ração, com perdas no desenvolvimento dos animais. Galpões de frangos não devem ter incidência direta de luz solar.** Os bebedouros não devem ter sua capacidade máxima em fluido, uma vez que **podem molhar o piso e gerar diminuição de temperatura corporal dos frangos.**

Fase de crescimento é o período em que as aves apresentam maiores ganhos de massa e tamanho, e que o avicultor precisa ter maior preocupação com **temperatura, umidade e o controle de ventilação**, e de reabastecimento de água e ração; além disso, os processos de higienização também são considerados cruciais pelos avicultores. Essa fase dura entre 21 a 35 dias e a disponibilidade de ração deve ser constante, os comedouros e bebedouros devem ser regulados na altura do dorso da ave, o que evita desperdícios e dificuldades de ingestão. Nesta fase, os frangos sentem mais calor do que frio, portanto é importante as **instalações manterem sombreadas e com ventilação adequada**, além de água fresca. Os materiais utilizados no piso (como serragem ou casca de arroz), deve **manter-se seco**, quando molhados devem ser trocados evitando desenvolvimento de agentes patogênicos como fungos e bactérias (RODRIGUES; ET. AL, 2014).

A fase de pré-abate, também conhecida por alguns entrevistados como fase de acabamento, nesse período o consumo de ração é aumentado significativamente, chegando a 120g por frango, e **a principal causa de mortalidade é o calor, enquanto que na fase inicial é o frio.** Na fase pré-abate também foi citado pelos

entrevistados que com muito calor as aves soltam muitas penas, o que pode deixar o trabalho de sanitização posterior mais moroso.

Por fim, demais considerações sobre os processos produtivos são em relação ao desperdício de insumos (ração, água e energia elétrica para os ventiladores mecânicos); o desempenho e a conversão alimentar dependem da qualidade genética dos animais e dos cuidados com a nutrição e manejo. Na medida que a idade aumenta, a ave piora sua conversão alimentar, ou seja, gasta mais ração para converter em kg de peso vivo. O índice “Viabilidade” indicado por uma agroindústria e por uma cooperativa é que com 49 dias tenha uma **viabilidade de 93 %**, ou seja, **7% de mortalidade**.

O Quadro 20 apresenta a lista de elementos da “Matriz de Needs” para as questões práticas de manejo na fase de alojamento.

Quadro 20 - Matriz de Needs para fase de alojamento

Elemento	Lista de elementos	Importância	Satisfação	Oportunidade
N1	Umidade e temperatura ambiente	10	4	16
N2	Umidade e temperatura das camas	9	2	16
N3	Temperatura e turbidez da água	9	2	16
N4	Temperatura das campânulas	9	5	13
N5	Níveis de ração e água nos comedouros	8	6	10

Fonte: dados da pesquisa (2023)

Os elementos N1, N2 e N3 se posicionaram como requisitos prioritários e necessários na fase de alojamento, o elemento N4 ficou na divisa de satisfação, indicando média satisfação por parte dos produtores, porém elencado como importância alta no controle. E o elemento N5, embora considerado de alta importância indicou média satisfação.

Além dos elementos da fase de alojamento, foram elencados três novos elementos para as fases de crescimento e pré-abate, conforme Quadro 21.

Quadro 21 - Matriz de needs para fase de crescimento e pré-abate

Elemento	Lista de elementos	Importância	Satisfação	Oportunidade
N6	Peso médio semanal das aves	8	2	14
N7	Consumo diário de ração	7	6	8
N8	Luminosidade do aviário	7	6	8

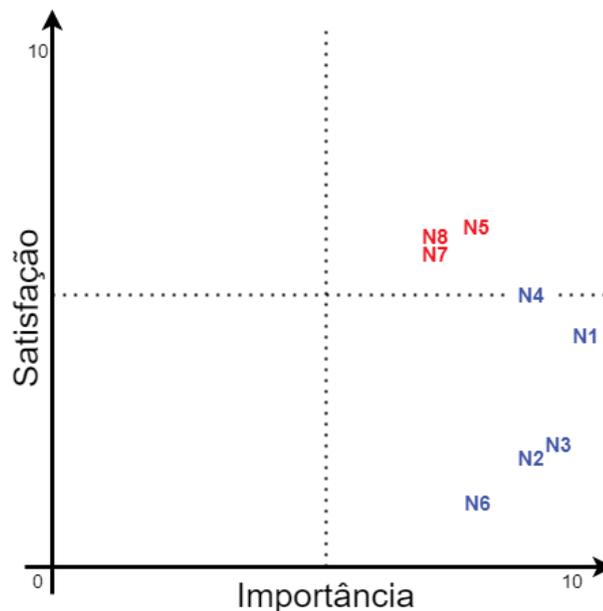
Fonte: dados da pesquisa (2023)

O elemento N6 foi classificado como alta prioridade uma vez que se mostrou importante pelos entrevistados para garantir uma alta remuneração e que se torna uma necessidade visto que o ganho de peso é considerado elemento chave no

desempenho produtivo e o manejo atual encontra-se em um estágio operacional artesanal. Os elementos N7 e N8 apresentaram-se como importantes, contudo não foram considerados necessidades uma vez que o cálculo de consumo diário de ração é realizado pelo próprio sistema de pedidos de ração, fornecido pela própria agroindústria; e a luminosidade do aviário é controlada pelo uso de cortinas e sua construção já é projetada para ficar no melhor ângulo possível. Um dos entrevistados ressaltou que orçou um sistema autônomo de cortinas para controle de iluminação, porém o custo não compensará o benefício.

A Figura 33 ilustra o posicionamento dos oito elementos presentes na matriz.

Figura 33 - Posicionamento dos elementos na matriz de needs



Fonte: dados da pesquisa (2023)

Seguindo o guia das entrevistas, para identificar as lacunas tecnológicas, o pesquisador direcionou as perguntas sobre quais tecnologias da informação possuem acesso. De pronto, os entrevistados responderam que possuem um aplicativo desenvolvido pela cooperativa para pedirem ração e outros insumos para a produção.

Pelo aplicativo é possível também estimar os custos com matéria prima e verificar seu histórico de compras; também tem acesso pelo aplicativo da cotação diária dos frangos, milho e soja e conseguem consultar a previsão do tempo.

Quando indagados sobre quais tecnologias de cunho IoT existem para a gestão em tempo real dos processos de manejo, os avicultores citam que são serviços oferecidos por empresas terceiras, mas nenhum deles contratou os serviços e o

empecilho citado foi o alto custo de implementação. Um respondente cita que não sente a necessidade uma vez que sua produção é pequena e a própria família consegue gerenciar, mas ressaltou que se interessa em modernizar sua granja pensando em expandir a produção no futuro. Embora nenhum dos produtores entrevistados tenha um sistema implementado, dois deles citaram o nome de duas empresas que prestam esse tipo de serviço e dizem conhecer produtores vizinhos que tenham contratado tal tecnologia.

Nas entrevistas com representantes da cooperativa do oeste paranaense e da agroindústria da mesma região, foi afirmado que, eventualmente, a cooperativa acompanha a contratação e implementação de sistemas de automação em granjas. Sendo que são produtores maiores de médio e grande porte que se interessam.

A principal justificativa foi a da dificuldade de financiamento para investir em tecnologias, uma vez que o dinheiro adquirido por financiamento e empréstimos são para insumos básicos da criação, manutenções em geral e até com seguro em alguns casos, citou um dos respondentes.

Ademais, o representante da cooperativa diz que o investimento em modernização dos galpões se paga, em média, depois de 40 meses. Também foi esclarecido que essas empresas oferecem pacotes distintos de soluções; o mais contratado segundo dados internos do representante da cooperativa são os de controle e monitoramento ambiental (temperatura, umidade e pressão). Um entrevistado afirmou que existem os sistemas que, além do controle ambiental, integram sensores de iluminação, pesagem de ração, detector de fumaça integrado a alarme de incêndio.

Durante a entrevista foram também citados aplicativos e sites desenvolvidos pela EMBRAPA como o "*Conforcalc*", aplicação indicada para produtores de frango de corte, técnicos, professores e estudantes. O aplicativo determina o nível de conforto térmico ambiental das aves pelo cálculo do ITGU - Índice de Temperatura do Globo negro e Umidade.

Outro aplicativo citado pelo representante da Agroindústria foi o "*Custo Fácil*" que estima os custos de produção. Voltado para a alimentação de aves, a EMBRAPA também disponibiliza o aplicativo "*Energcalc*" que calcula a energia metabolizável por aves, determinando valores referência para a formulação de rações.

Quando o pesquisador questionou os entrevistados sobre a quantidade de plataformas que oferecem algum tipo de apoio tecnológico, um dos produtores rurais

falou que não tem familiaridade com computador e *smartphone* então evite seu uso, afirmando que é seu filho que realiza as operações de compra de insumos; e que geralmente gosta de acessar a internet para ler as notícias do setor e verificar a cotação das *commodities*. Outro entrevistado diz não conhecer as plataformas disponibilizadas pela EMBRAPA, que apenas usa o aplicativo da cooperativa.

O representante da cooperativa, cita que existem muitos aplicativos, sites e guias técnicos na internet com intenção de aportar a gestão da avicultura de corte, porém que cada uma dessas aplicações tem um foco e não há uma plataforma integrada que atenda todas as necessidades. O representante também cita que os sistemas de controle ambiental ofertados pelas empresas são reativos, ou seja, por exemplo, a temperatura do galpão sobe, o avicultor observa no painel e reage ligando o sistema de ventilação. O Quadro 22 apresenta a Matriz de *needs* para as demandas tecnológicas.

Quadro 22 - Matriz de needs para as demandas de tecnologia IoT

Elemento	Lista de elementos	Importância	Satisfação	Oportunidade
N9	Predição no consumo de ração	9	4	14
N10	Predição no monitoramento ambiental	9	3	15
N11	Monitoramento de consumo energético no galpão	8	3	13
N12	Guias do Agro 4.0 e conteúdo técnico	6	1	11
N13	Plataforma de registro de ocorrências	5	2	8
N14	Modelo de plataforma de indicadores para acompanhamento coletivo	8	1	15

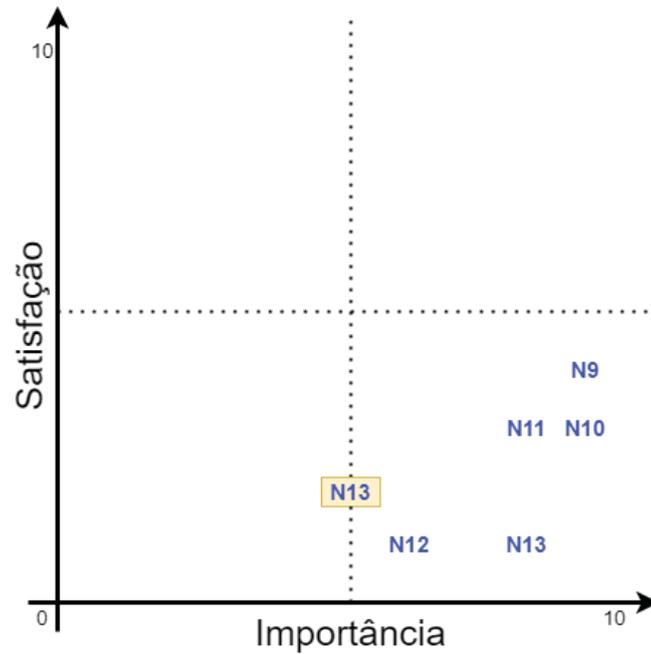
Fonte: dados da pesquisa (2023)

Em síntese, as lacunas verificadas estão relacionadas, de certo modo, com um modelo preditivo envolvendo insumos N9 e o monitoramento ambiental N10, envolvendo uma ação preditiva de controle de temperatura e umidade no ambiente. Também foi identificado o interesse pelo monitoramento do consumo energético dos dispositivos elétricos do galpão N11. Nas entrevistas também sobressaíram o interesse de uma integração de Guias com conteúdo técnico e de assuntos relacionados a notícias do Agro 4.0 (N12).

Por se tratar de uma plataforma integrativa com dispositivos eletrônicos, também foi citada a presença de uma plataforma de registro de chamados de problemas, de modo que o produtor consiga registrar as ocorrências (N13). Por fim, o N14 foi indicado como uma novidade entre os entrevistados, uma vez que desconhecem uma aplicação que forneça acompanhamento coletivo de desempenho.

A Figura 34 posiciona os elementos entre N9 e N14 nos quadrantes da Matriz de Needs.

Figura 34 – Posicionamento dos elementos de demanda tecnológica



Fonte: dados da pesquisa (2023)

Com o mapeamento das necessidades e prioridades, a Seção 5 apresenta a modelagem física e computacional da proposta IoT.

5 DESENVOLVIMENTO DA ARQUITETURA IOT

A seção apresenta a modelagem da arquitetura IoT em duas granjas de frangos de corte, e seguem os passos metodológicos da subseção 3.5.

Primeiro é mostrado como foi realizada a modelagem da camada física do sistema embarcado; seguida pela apresentação das lógicas dos algoritmos e esquemas da camada *back-end* com o ciclo de coleta, armazenamento e tratamento dos dados (**OBJETIVO ESPECÍFICO B**).

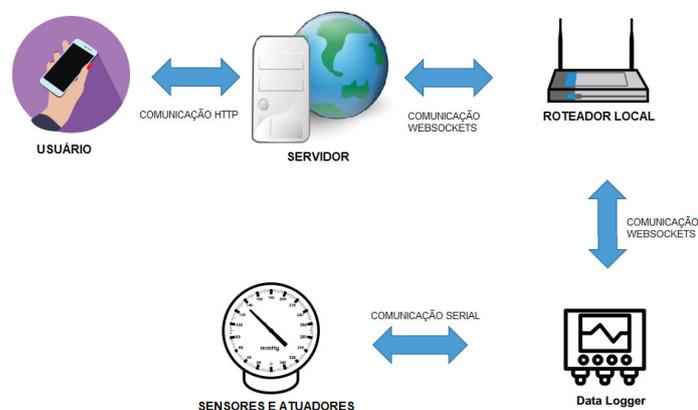
A seção é finalizada com a modelagem computacional da camada de *front-end* para *Web* e *Mobile* e suas respectivas integrações com o servidor (**OBJETIVO ESPECÍFICO C**).

5.1 MODELAGEM DA CAMADA FÍSICA

O desenvolvimento da camada física para a coleta e envio de dados, especificamente da comunicação entre o *Data Logger*, roteador local e o servidor externo é independente de cabos pois utiliza o *wi-fi* do NodeMCU, o que permite ao usuário controlar remotamente qualquer componente ligado aos microcontroladores, ou seja, acionar ou desativar qualquer porta digital (controle dos módulos relé).

Com o sistema é possível ter a ação proativa do usuário em mandar os comandos para os microcontroladores via servidor externo, também há a ação reativa onde o microcontrolador envia os dados sensorizados ao servidor. A Figura 35 ilustra o fluxo de comunicação do ambiente físico com o digital.

Figura 35 - Fluxo de comunicação da arquitetura IoT



Fonte: o autor (2023)

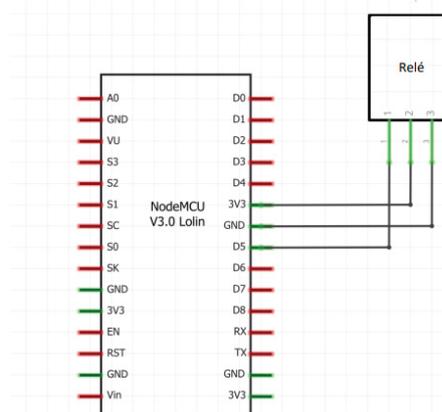
A camada física foi projetada de forma a atender um fluxo bidirecional de comunicação em tempo real entre o usuário e sua granja, não restringindo as formas reativas e proativas de comunicação e gestão dos dados.

A arquitetura para ser independente de serviços externos, podendo ser implementada em uma rede local para uso doméstico única e exclusiva do próprio produtor, contudo o protocolo HTTP possui a limitação do servidor só se comunicar com o cliente se houver uma requisição.

Para contornar o problema e dispensar a utilização de um *broker mqtt* o que demandaria maiores investimentos, configurações do usuário e aumento de requisitos para implementação da arquitetura, foi utilizado o protocolo *websocket*, o qual roda no mesmo servidor HTTP a única diferença é que toda vez que for mudado o *status* do atuador no banco de dados há o envio para o DataLogger que atua.

Para desenvolver a lógica de funcionamento das ações proativas, ou seja, verificar se o microcontrolador está recebendo os buffers de dados vindos do servidor e alocando na memória física do microcontrolador acionando ou desacionando alguma saída do microcontrolador foi utilizada um módulo relé de dois canais e acoplados a uma lâmpada, a qual estava conectada ao circuito da granja. O esquema elétrico utilizado é ilustrado na Figura 36.

Figura 36 – Esquema elétrico do atuador - modulo rele

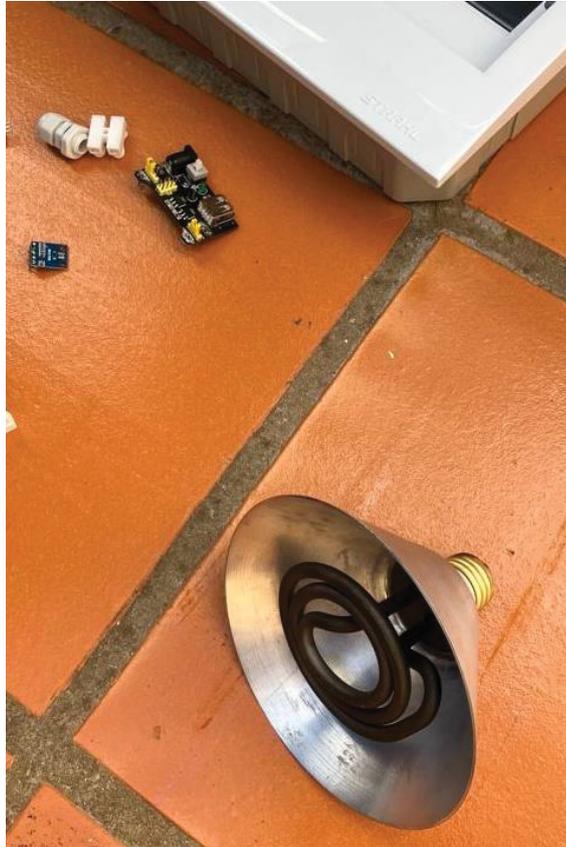


Fonte: o autor (2021)

O usuário enviava o comando via plataforma web para o servidor que por sua vez via *websocket* encaminhava o comando ao microcontrolador, alterando o estado do pino D5 que estava conectado ao módulo relé. No caso da pesquisa, o relé estava conectado a uma lâmpada de campânula e foi utilizada na fase de alojamento para aumentar a temperatura e luminosidade nos primeiros quinze dias.

A Figura 37 apresenta a campânula elétrica para aquecimento das aves que foi utilizada junto com o módulo relé da proposta (atuador) no alojamento.

Figura 37 - Integração do Data Logger com o atuador

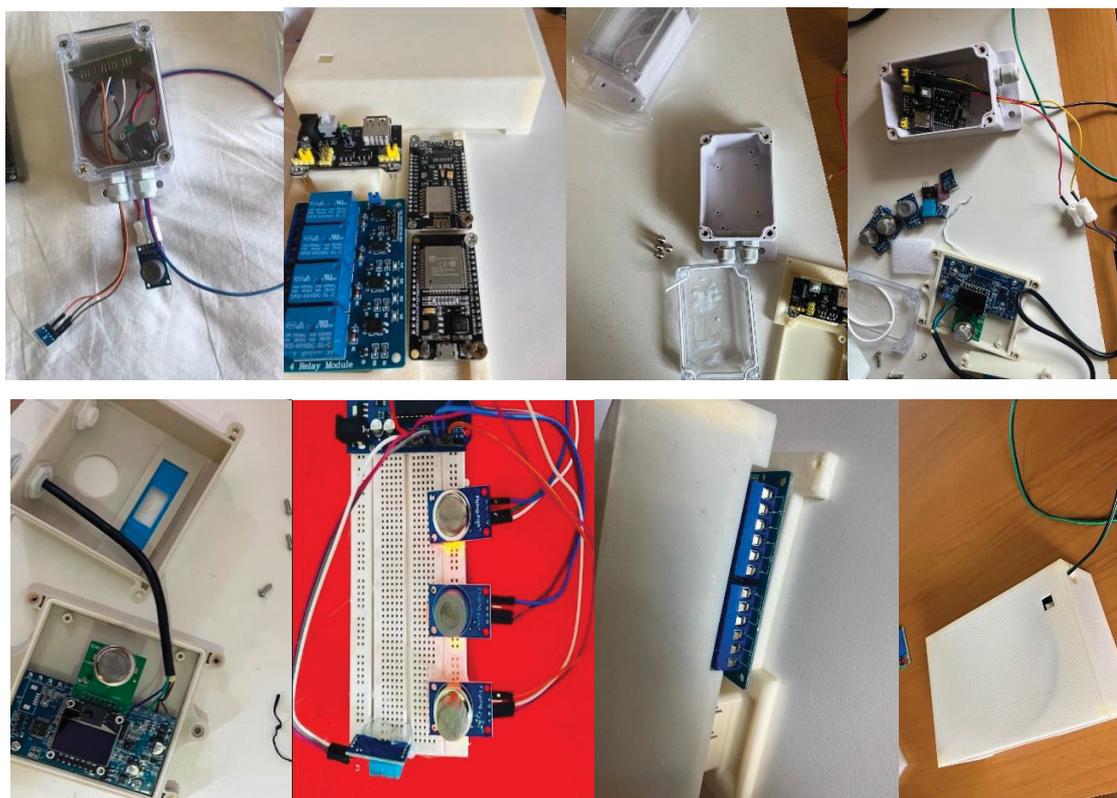


Fonte: o autor (2023)

O funcionamento reativo do *Data Logger* diz respeito aos sensores, e sua lógica baseia-se em repetição, a cada período determinado o algoritmo do *Data Logger* executa e envia uma requisição *get* via HTTP para o servidor, informando: o ip do microcontrolador, carimbo de data e horário e os dados de leitura dos sensores..

A Figura 38 mostra registros dos testes e montagem dos sensores e demais componentes do *Data Logger*.

Figura 38 – Registro das montagem do Data Logger



Fonte: o autor (2023)

Cabe ressaltar que antes da implementação, foram realizados testes nos *Data Logger*, os quais ficaram 12 horas conectados enviando dados ao servidor, de modo a verificar se apresentariam algum problema ou falhas. Após isso, foram encaminhados para acomodação e conexão na granja, conforme a Figura 23 apresentada anteriormente.

5.2 MODELAGEM DO BACK-END

O *back-end* é responsável por gerir todo o fluxo de dados da arquitetura desde a ponta, na coleta dos dados no ambiente físico até a sua persistência no banco de dados, em outras palavras, essa etapa engloba a programação do microcontrolador e toda a configuração do servidor para que interaja com o registrador de dados, além disso o *back-end* por meio da API vai também alimentar o sistema que o usuário consome (*front-end*).

5.2.1 Módulo de coleta, armazenamento e tratamento dos dados

O ciclo de coleta de dados diz respeito ao algoritmo compilado no microcontrolador utilizando a linguagem C++, a lógica consiste em estabelecer uma conexão com o servidor e enviar a requisição com os dados. A Figura 39 apresenta o fluxograma de funcionamento inicial do algoritmo para a extração dos dados, onde é aberta a conexão com a *internet*, nesse caso, via *WiFi*, seguido pela comunicação serial entre os sensores e o microcontrolador que captura os dados.

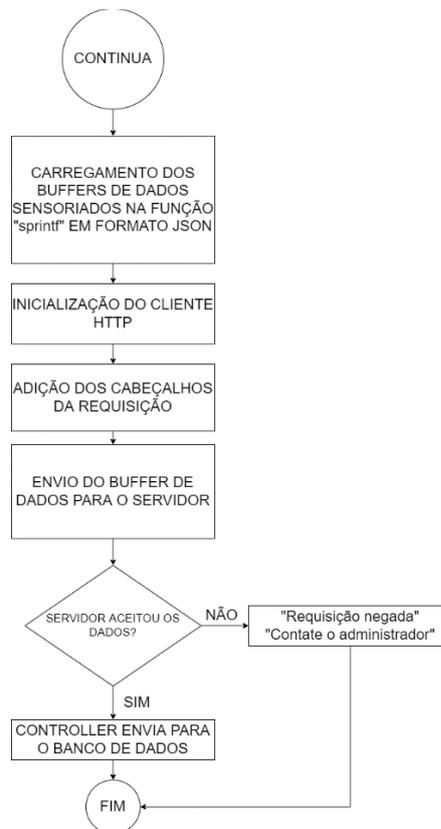
Figura 39 – Fluxograma 1 de extração dos dados



Fonte: o autor (2023)

O procedimento seguinte do algoritmo é a formatação em JSON dos dados e envio via método “*http-post*” para a API, a qual realiza a persistência dos dados no banco de dados (*MongoDB*), conforme ilustrado na Figura 40

Figura 40 - Fluxograma 2 de extração de dados



Fonte: o autor (2023)

Para serem aceitos pelo servidor, os dados devem conter em seu buffer JSON: i) identificação do usuário da granja no qual está instalado; ii) endereço de IP do microcontrolador na rede; iii) as grandezas de medida que estão sendo enviadas; iv) modelo do sensor; v) valor capturado pelo sensor. Caso contrário, o servidor nega a requisição e os dados enviados são perdidos, necessitando uma nova coleta e tentativa de envio.

A identificação do usuário serve como chave de segurança para evitar que sejam enviados dados ao servidor sem ser um microcontrolador cadastrado ou para que haja tentativas maliciosas em preencher o banco de dados sem permissão.

O endereço de IP identifica a localização física do microcontrolador, ou seja, caso exista mais de um microcontrolador dentro do mesmo ambiente é possível saber de quais sensores estão vindo os dados, garantindo a rastreabilidade dos dados e facilitando identificar eventuais danos físico aos sensores que estejam causando o envio incorreto ao servidor. O algoritmo de coleta de dados pode ser verificado na integra no repositório "<https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas>".

A camada de armazenamento de dados contempla as rotas do tipo “*post*” do servidor, os “*controllers*” responsáveis pela persistência no banco de dados e, por fim, o esquema do *MongoDB*. Existem seis rotas na API responsáveis pelos processos de armazenamento de dados no *MongoDB*, e o método “*http*” é do tipo “*post*” e o utilizado no *controller* é o “.store”; o Quadro 23 descreve as rotas.

Quadro 23 - Rotas da API do tipo POST

Nome da rota	Descrição	Observações adicionais
/users	Cadastra um usuário no banco de dados, coletando: nome, e-mail, senha.	As validações são realizadas com a biblioteca “yup” do JS e exigem o tipo de dado <i>string</i> , com o mínimo de 6 caracteres para a senha e o formato “@domínio” para o e-mail. A rota também evita o cadastro duplicado de usuários e criptografa a senha ao gravar no banco de dados.
/session	Abre uma seção para um usuário previamente cadastrado,	É concedido um <i>token</i> criptografado de autorização para acesso na aplicação com validade de 30 dias e se renova automaticamente a cada acesso ao sistema. Isso evita que o usuário seja deslogado caso faça uso da aplicação continuamente.
/farms	Cadastra informações do produtor rural, como o nome da propriedade, cidade, responsável, cultivos, descrição, latitude e longitude	Essa rota permite o usuário a cadastrar sua propriedade, os dados de geolocalização são utilizados para o uso do recurso de mapa na aplicação <i>front-end</i> .
/heps	Cadastro de chamado do usuário, com seu id, assunto do problema e descrição.	Compõem o módulo de “ajuda”, onde podem conectar o produtor rural com o suporte de TI e outros profissionais de interesse como Veterinários, Zootecnistas e/ou Eng. Agrônomos.
/sensors	Rota acionada pelo microcontrolador para o cadastro dos dados coletados.	Para o cadastro dos dados nessa rota são requeridos o <i>token</i> de autorização e o id do usuário.
/notification	Envia notificações para o <i>front-end</i> de cada usuário	Essa rota só pode ser acessada com o <i>deploy</i> da aplicação, uma vez que ela realiza o serviço de enviar notificações aos usuários, como por exemplo: “Alertas”.

Fonte: o autor (2023)

O esquema do banco de dados *MongoDB* se dá por documentos, por ser NoSQL não existem relacionamentos entre entidades, esquema “chave: valor” de arquivo JSON.

Como pode ser visto no exemplo da Figura 41, o documento armazenado refere-se a um array de objetos da rota “/sensors”, identificados por um “_id”, “user_id”, “tipoMedida1”, “tipoMedida2”, “unidadeMedida1” e assim por diante.

Figura 41 - Exemplo de documento JSON armazenado no MongoDB

```
1 [
2   {
3     "_id": "60d9ebbf14cc00001f06dd4f",
4     "user_id": "60d606c936ce5c001f176ef5",
5     "tipoMedida1": "Umidade",
6     "tipoMedida2": "Temperatura",
7     "unidadeMedida1": "%",
8     "unidadeMedida2": "°C",
9     "modeloSensor": "DHT-11",
10    "valor1": 63.099998,
11    "valor2": 21.5,
12    "ipArduino": "192.168.25.15",
13    "timestamp": "2021-06-28T15:33:19.429Z",
14    "__v": 0
15  },
16  {
17    "_id": "60df3e09c21239001fe04d38",
18    "tipoMedida1": "Umidade",
19    "tipoMedida2": "Temperatura",
20    "modeloSensor": "DHT11",
21    "valor1": 11,
22    "valor2": 10,
23    "ipArduino": "192.168.13.15",
24    "timestamp": "2021-07-02T16:25:45.966Z",
25    "__v": 0
26  },
27  {
28    "_id": "60f8cdf40d29ad001f6cddb2",
29    "user_id": "60d606c936ce5c001f176ef5",
30    "tipoMedida1": "Umidade",
31    "tipoMedida2": "Temperatura",
32    "unidadeMedida1": "%",
33    "unidadeMedida2": "°C",
34    "modeloSensor": "DHT-11",
35    "valor1": 64,
36    "valor2": 17.1,
37    "ipArduino": "192.168.25.15",
38    "timestamp": "2021-07-22T01:46:28.476Z",
39    "__v": 0
40  }
41 ]
```

Fonte: o autor (2023)

A documentação da API e o modelo do banco de dados encontram-se no Apêndice 9 e no repositório público da pesquisa (["https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas"](https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas)).

Sempre que persistido os dados cadastrais de um usuário, os mesmos passam por um processo de criptografia JWT, o qual, descaracteriza sua identificação, dando ao usuário um código alfanumérico do tipo *string* com 24 caracteres aleatórios.

O tratamento de dados ocorre principalmente para o atendimento das regras do negócio, ou seja, são as rotas de acesso aos dados. Pois no contexto IoT os usuários realizam muitas consultas (*queries*) nos dados coletados. No *back-end* as consultas são realizadas pelo método "http get", os *controllers* filtram as informações do banco de dados e retornam com os parâmetros pedidos.

No caso do avicultor, por exemplo, ao iniciar um novo ciclo de crescimento e engorda de frango buscará obter dados associados aquele período em que os frangos ficam alojados em sua granja, tais como: temperatura, umidade, níveis de gás amônio, índice de perda diária entre outros. Ou até mesmo consultar dados passados, para comparação. Dessa forma, o servidor recebe o pedido de consulta, filtra os dados e fornece ao requerente.

De modo a garantir a preservação de dados sensíveis de pessoas, o tratamento dos dados advindos do cadastro na plataforma passa obrigatoriamente por um processo de criptografia. O servidor não fornece uma rota para retorno dos elementos de cadastro de usuários, esses dados são utilizados exclusivamente para autenticação na plataforma.

5.2.2 Camada de serviços para os atuadores

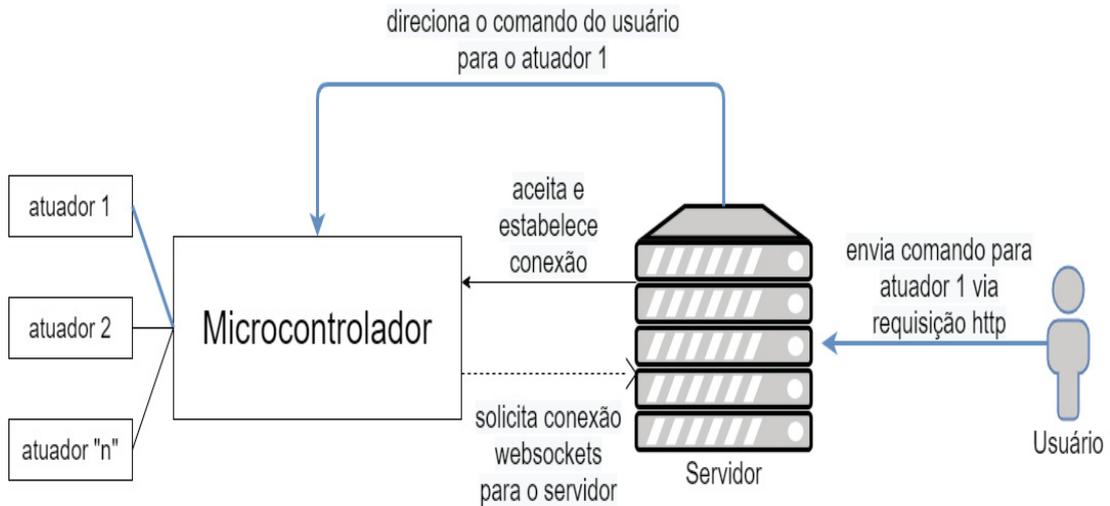
Por serem elementos finais de um controle, as ações dos atuadores podem ocorrer de duas formas: i) Ação automatizada na qual o atuador é acionado a cada período de tempo pré-determinado ou a uma condição ambiental detectada via sensores. Por exemplo, é possível programar o sistema de ventilação para ser acionado a cada 24 horas com duração de 50 minutos; e/ou acionar o sistema de ventilação sempre que a temperatura ultrapassar um determinado valor. Ou ii) tomada de decisão do usuário, permitindo que os atuadores sejam acionados e desacionados a qualquer momento, tanto via aplicação quanto localmente nas granjas.

Para que seja possível a gestão dos atuadores remotamente é necessário que o microcontrolador estabeleça uma conexão com o servidor via *WebSockets*, permitindo ao servidor enviar comandos ao microcontrolador sem precisar de uma requisição.

Essa solução resolve a limitação de controle dos atuadores somente em rede local, como nos trabalhos de Mulay, Ochiai e Esaki (2017) e Li e Liu (2020) onde o microcontrolador é configurado como servidor e recebe requisições "*http get*" para ativamente ou desativamento.

A proposta apresentada de conexão proativa do microcontrolador ao servidor externo soluciona problemas de segurança, rede local e escalonamento de implementação; se apresentando também como alternativa ao protocolo MQTT. A Figura 42 ilustra o fluxo de comunicação dos atuadores com o servidor.

Figura 42 - Fluxo de comunicação dos atuadores



Fonte: o autor (2023)

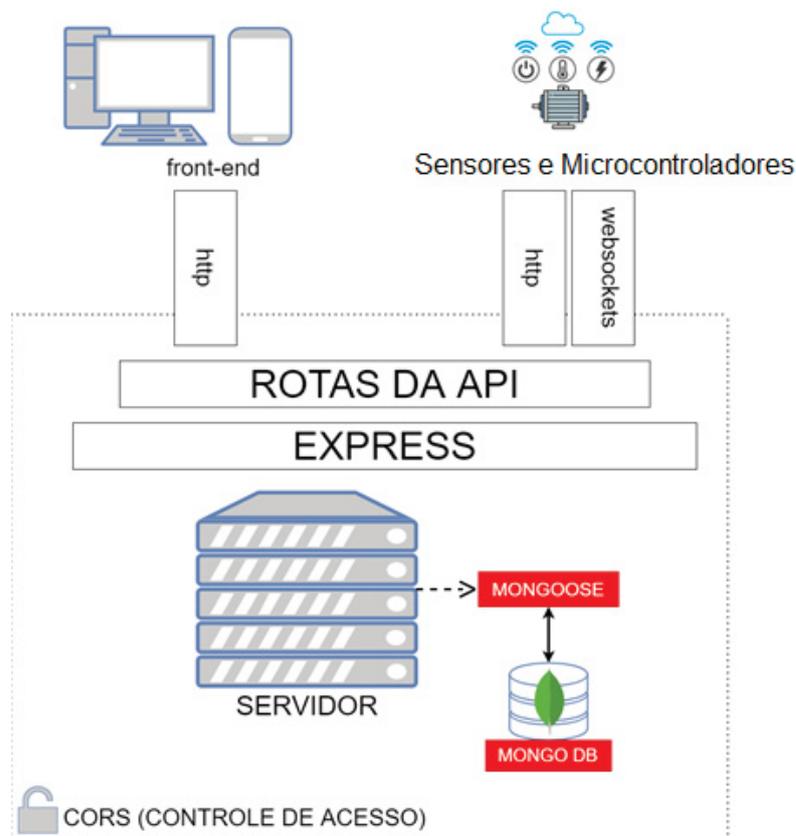
A subseção seguinte apresenta a estrutura do servidor e suas respectivas interações dentro da arquitetura proposta.

5.2.3 Configurações do servidor

A configuração do servidor foi definida para receber requisições de usuários (*front-end*) do tipo "*http*" em um esquema de rotas, onde cada requisição é direcionada a uma rota específica. Além dos usuários, o servidor também recebe requisições "*http*" dos microcontroladores com os dados dos sensores, o qual direciona para o armazenamento no *MongoDB*. Para permitir o controle dos atuadores, o servidor também abre uma porta de comunicação *websockets* para que o microcontrolador solicite conexão.

A comunicação entre o servidor e os clientes são realizadas pela biblioteca "*CORS - Access-Control-Allow-Origin*", que funciona como um moderador que verifica a origem (*IP*) das requisições e não permite o servidor realizar operações se não forem endereços *IP* autorizados, protegendo o sistema de ataques cibernéticos. A estrutura do servidor é ilustrada conforme Figura 43 .

Figura 43 - Estrutura do servidor



Fonte: o autor (2023)

São utilizadas seis bibliotecas para o servidor, sendo uma do framework express, do banco de dados (MongoDB), CORS como módulo de segurança, e http e socketio para configuração dos modos de comunicação. O arquivo de configuração do servidor está no repositório público da tese, especificamente no *link*: ["https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas"](https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas)

5.3 MODELAGEM DO FRONT-END

A modelagem computacional do *front-end* é responsável pela camada visual e de navegação da aplicação. Como a aplicação é multiplataforma, a modelagem será distinta para os navegadores *web* e para dispositivos *mobile* com sistemas operacionais iOS e Android. O desenvolvimento da etapa de *front-end* engloba os elementos de experiência de usuário (UX), como o objetivo da aplicação, as necessidades do usuário, interações, elementos visuais de interface e navegação.

5.3.1 Front-End Web

A página web e mobile foram construídas com um layout semelhante, e o *framework ReactJS* para o desenvolvimento web da aplicação permite criar componentes distintos e reutilizáveis que encapsulam o comportamento da página, renderizando apenas alguns dos elementos, permitindo a navegação ficar mais rápida e fluida (PAUL; NALWAYA, 2019).

A escolha da paleta de cores seguiu os princípios citados por Nerone et al. (2020), os quais afirmam que o princípio para a escolha da paleta de cores deve seguir uma lógica visual, utilize a roda de cores buscando um oposto de cores, tons de cores e o tom de cada cor definida, evitando exceder cinco cores distintas. Foram selecionadas quatro cores distintas, sendo que para uma delas foram selecionados três tons (verde). A Figura 44 exibe as cores escolhidas para integrar a camada do *front-end*.

Figura 44 - Paleta de cores front-end



Fonte: o autor (2023)

O usuário pode ter acesso de duas formas, se cadastrando diretamente na plataforma (tela de cadastro –Figura 45)ou sendo previamente cadastrada pelo administrador do sistema. Para o cadastro de usuário são requeridos apenas seu nome (para identificação dentro do aplicativo) e informações de *login*: endereço de *e-mail* (único para cada usuário) e a senha de acesso.

Figura 45 - Tela de Cadastro web

Bem vindo ao Farm UFPR 4.0

A APLICAÇÃO ESTÁ EM DESENVOLVIMENTO E PODE SOFRER ALGUMAS ALTERAÇÕES.

Obrigado pela disponibilidade em testar as funcionalidades e enviar um feedback, é muito importante para nós.

Cadastrar

Fonte: o autor (2023)

Com essas duas informações (*e-mail* e senha) o usuário cadastrado pode iniciar uma sessão pela tela de *login* (Figura 46), com as credenciais corretas é concedido um *token* de autenticação ao usuário para que possa navegar na aplicação.

Figura 46 - Tela de Login web

Bem vindo de volta

A APLICAÇÃO ESTÁ EM DESENVOLVIMENTO E PODE SOFRER ALGUMAS ALTERAÇÕES.

Obrigado pela disponibilidade em testar as funcionalidades e enviar um feedback, é muito importante para nós.

Entrar

[Esqueceu a senha?](#)

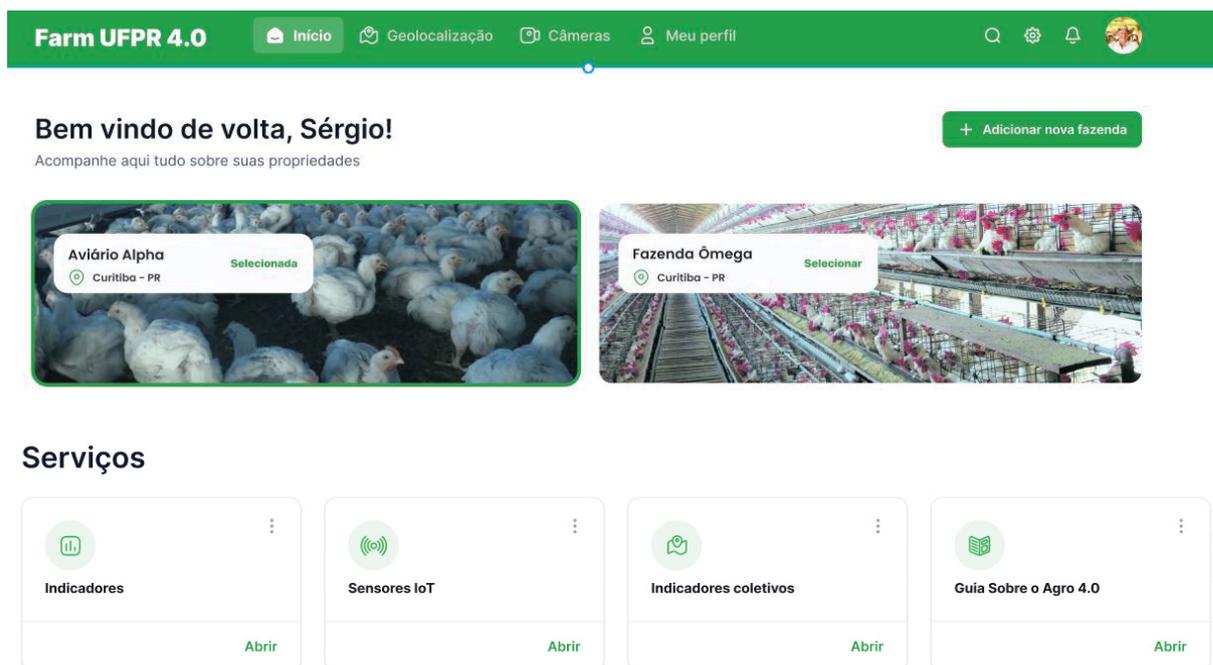
Ainda não tem acesso? [Cadastre-se aqui](#)

Fonte: o autor (2023)

Na tela de inicial de boas-vindas (tela *home*) - Figura 47 o usuário encontra o menu com as funcionalidades da aplicação – botões de início, geolocalização, perfil. Também está visível o avatar (item B), nome do produtor e sua (s) granja (s).

Além dos serviços que a plataforma oferece – Indicadores, Sensores IoT, Indicadores Coletivos e os guias sobre o agronegócio.

Figura 47 - Tela de inicial da aplicação web

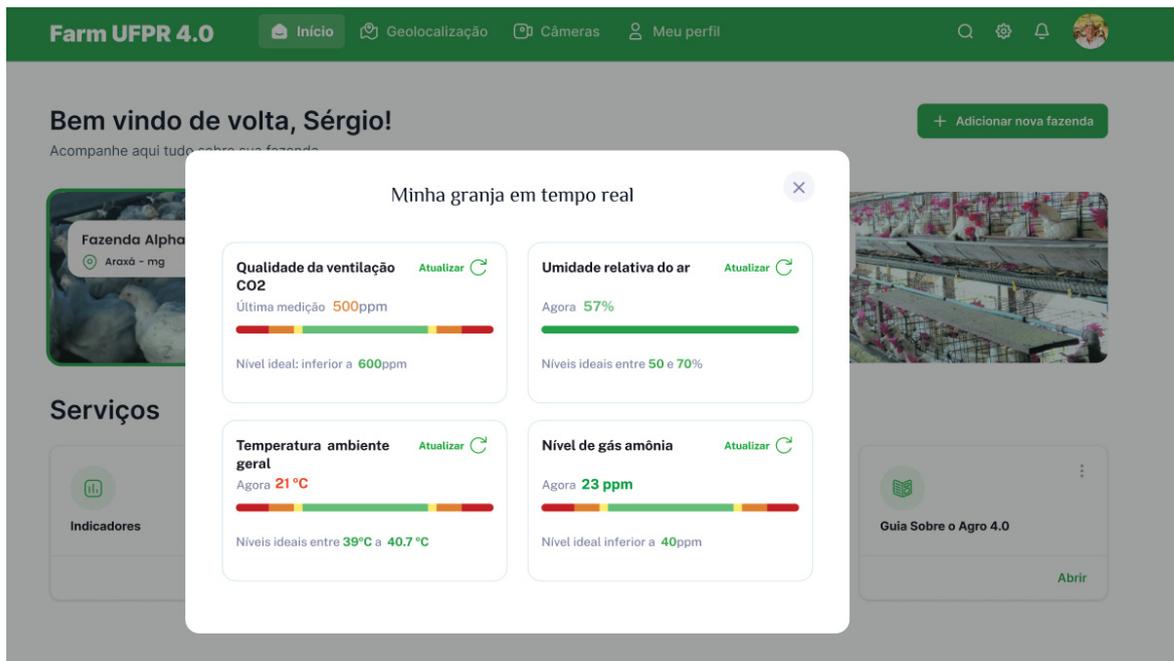


Fonte: o autor (2023)

No botão “Indicadores” o usuário terá acesso aos seus indicadores e conseguirá acompanhar em tempo real a situação de sua granja. Também é possível aplicar filtros nos dados de modo que possa realizar análises mais personalizadas e específicas.

Complementando o painel de dados, outras funcionalidades da aplicação são o controle dos atuadores e o acompanhamento dos sensores, nessas rotas o usuário consegue acompanhar dados dos sensores em tempo real (Figura 48)

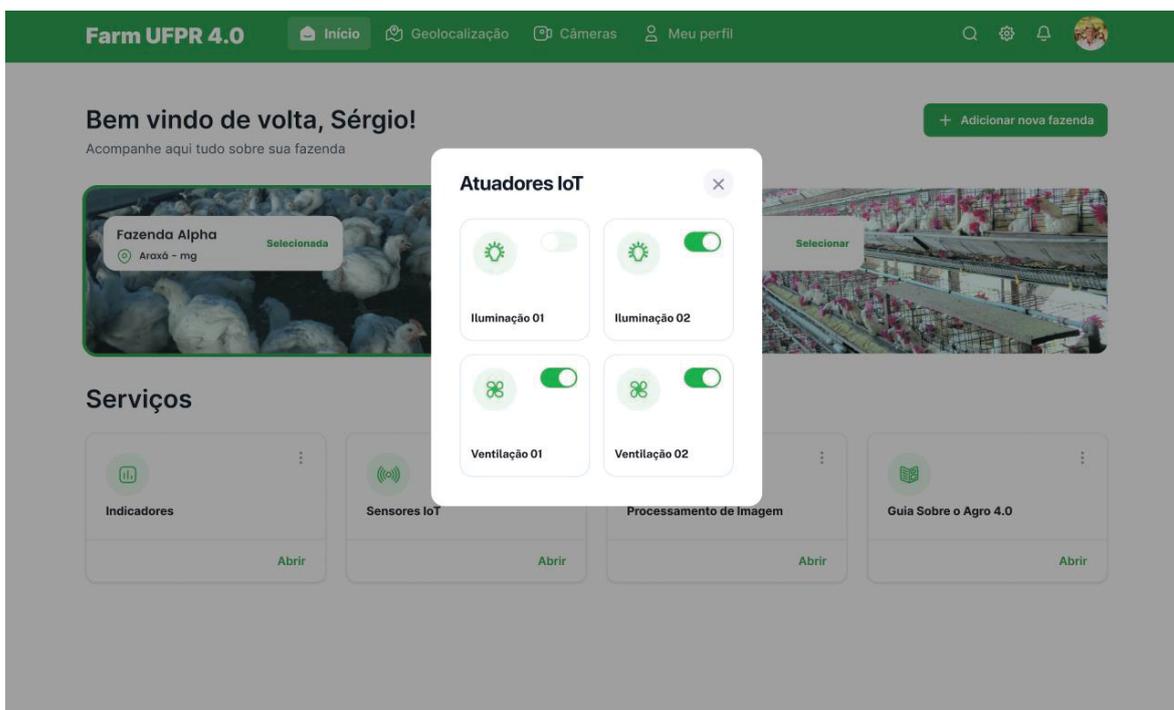
Figura 48 - Tela de acompanhamento de dados sensoriados web



Fonte: o autor (2023)

E uma tela para verificar os atuadores, consegue manipular via botão *switch* capaz de ligar e desligar. Essa manipulação não anula o uso manual dos equipamentos, que podem interagir juntos (Figura 49).

Figura 49 - Tela de manipulação dos atuadores web

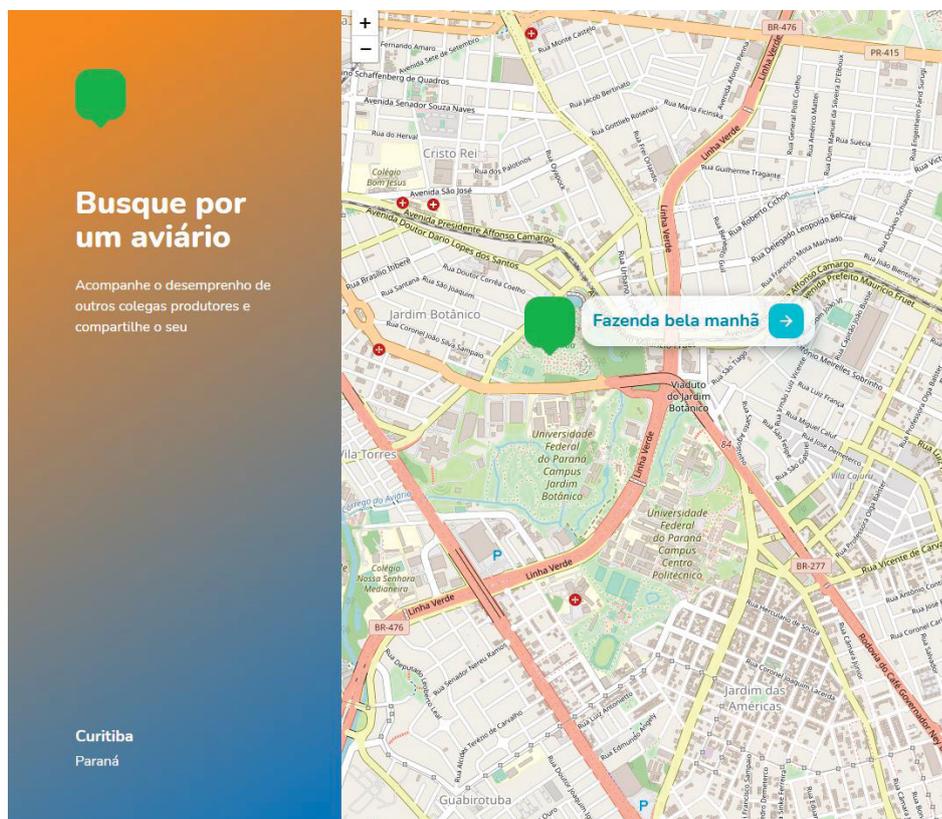


Fonte: o autor (2023)

Com a funcionalidade de geolocalização os usuários conseguem identificar o próprio aviário e outros cadastrados na plataforma, a intenção dessa funcionalidade é gerar informação coletiva sobre a produção, uma vez que o avicultor consegue comparar seus indicadores pessoais com uma média geral de produtores de uma mesma região.

Essa funcionalidade foi apontada nas entrevistas, e foi classificada como N12 na montagem da matriz de *needs* da Figura 33. A página para visualização do mapa está ilustrada na Figura 50.

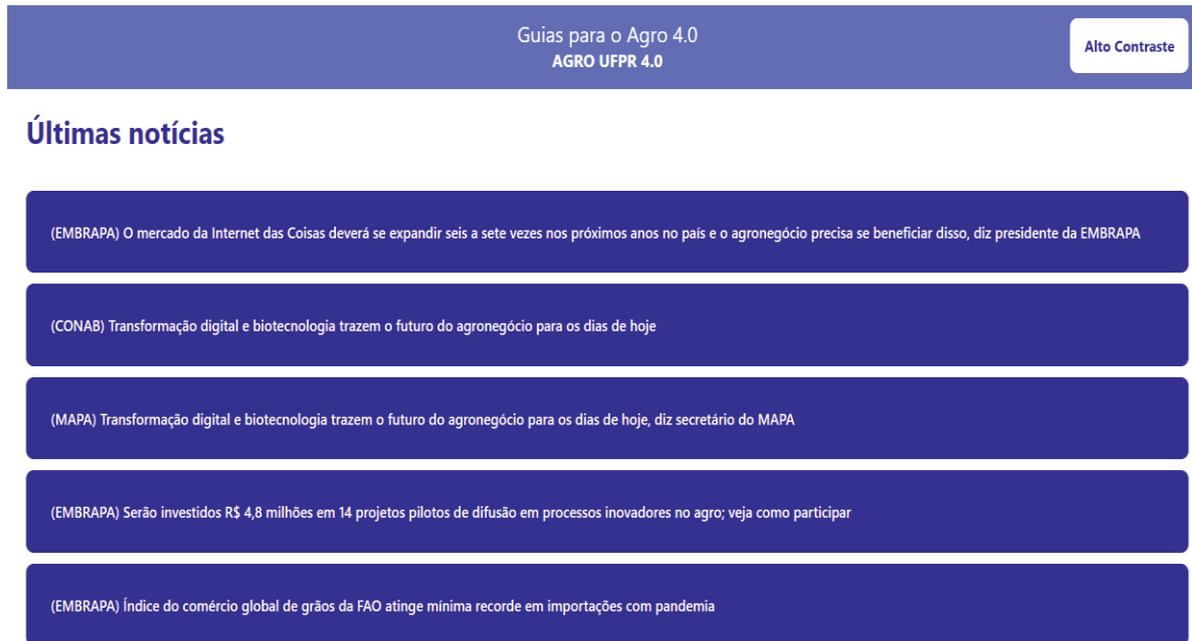
Figura 50 - Tela de Geolocalização Web



Fonte: o autor (2023)

Além das informações internas da própria produção, a aplicação disponibiliza mais duas funcionalidades, “Guias sobre o Agronegócio 4.0” e “Indicadores Nacionais do Agronegócio”; a primeira funcionalidade (Figura 51) tem o objetivo disponibilizar o acesso a informação de fontes oficiais do mercado Agro 4.0 com o foco nas tendências e novidades do setor, sob a perspectiva de democratização da informação e das tecnologias, essa funcionalidade encontra respaldo em Oliveira, Melo e Xavier (2020) e Moraes (2020) que afirmam ser possível disseminar conhecimentos junto à sociedade em contextos específicos.

Figura 51 - Exemplo da página Guias para o Agronegócio 4.0



Fonte: o autor (2023)

A página com as guias do Agro 4.0 são atualizadas automaticamente por meio de uma API externa desenvolvida pelo *Google* chamada “*News API*” acessada em “<https://newsapi.org/s/google-news-br-api>”, que está parametrizada para retornar a fonte da notícia (em parênteses), e o título da notícia; ao clicar o usuário é automaticamente redirecionado para a página noticiada na íntegra.

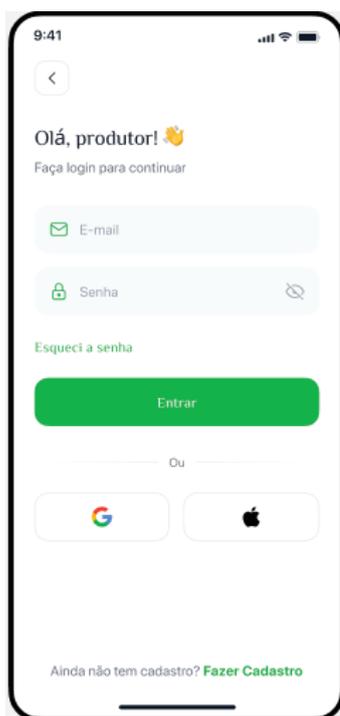
Essa funcionalidade de guias sobre o Agro 4.0 apareceram nas entrevistas como um fator de potencial importância, N14 na matriz de *needs* da Figura 33

5.3.2 Front-End Mobile

Para o desenvolvimento *Mobile* foi utilizado o *framework JavaScript do React Native*, unindo em um código único a renderização em tela (de modo nativo) independente se o *Smartphone* utiliza o sistema *Android* ou *iOS* (PAUL; NALWAYA, 2019).

Assim como na *web*, o usuário pode se cadastrar manualmente ou receber as credenciais do administrador do sistema. Para o cadastro de usuário são requeridos apenas seu nome (para identificação dentro do aplicativo) e informações de *login*: endereço de *e-mail* (único para cada usuário) e a senha de acesso (Figura 52).

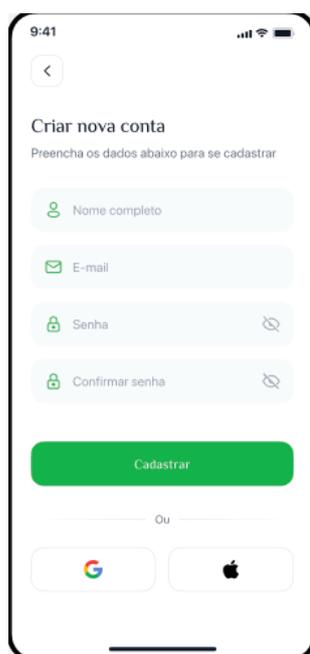
Figura 52 - Tela de cadastro *mobile*



Fonte: o autor (2023)

Com as credenciais de cadastro, o usuário pode iniciar uma sessão no aplicativo a partir da tela inicial vista na Figura 53. O diagrama de sequência dessas operações (cadastro e *login*) são iguais ao acesso *Web* e encontram-se no Apêndice 10.

Figura 53 - Tela de login *mobile*



Fonte: o autor (2023)

A tela principal do *mobile* encontram-se as quatro funcionalidades, o usuário encontra seus dados no cabeçalho (item A), um botão de acesso rápido ao acompanhamento em tempo real dos sensores (item B) e os botões para acessar os indicadores, tela de ação de sensores e atuadores, mapa de geolocalização e a página de guias do Agronegócio, conforme ilustrado em vermelho na Figura 54.

Figura 54 - Tela *Home mobile*



Fonte: o autor (2023)

A funcionalidade “Indicadores” substitui a funcionalidade “*Dashboard*” presente na versão *Web*, uma vez que o tamanho da tela é menor, e a velocidade de renderização dos dados em tela deve ser considerado, é levado para essa funcionalidade apenas as informações essenciais ao usuário, com gráficos montados utilizando os dados coletados (Figura 55).

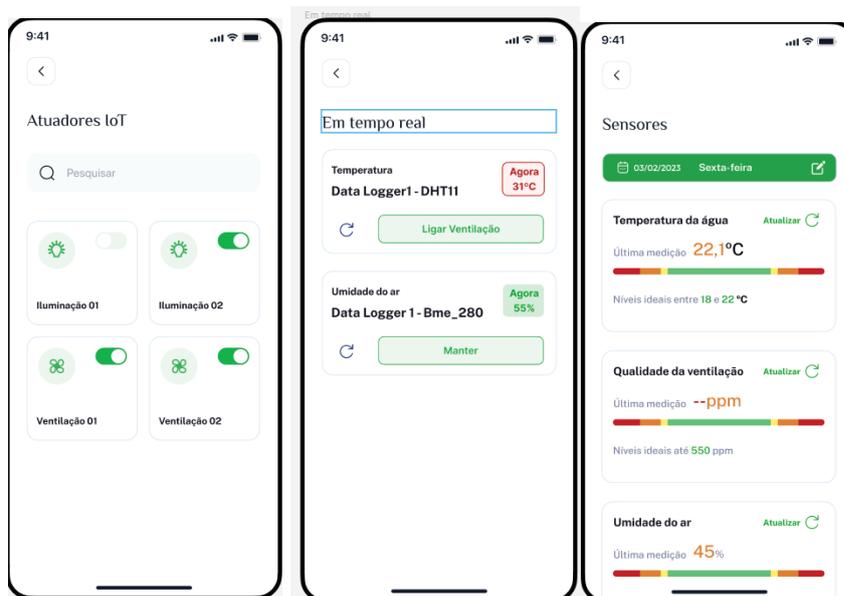
Figura 55 - Tela de indicadores *mobile*



Fonte: o autor (2023)

As funcionalidades “Sensores e atuadores IoT” fornecem acesso a todos os dispositivos conectados na granja do usuário que acessa a plataforma, no caso do exemplo da Figura 56, é possível verificar que estão implementados dois atuadores de acionamento de iluminação e um atuador de acionamento de ventilação.

Figura 56 - Tela de Sensores e Atuadores

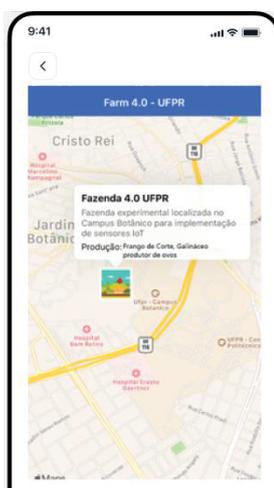


Fonte: o autor (2023)

Na tela de sensores, estão sendo monitorados a umidade relativa do ar do incubatório com os frangos de engorda, o sistema indica o dado capturado e a ação a ser tomada, seja acionar um atuador ou apenas manter o monitoramento.

A funcionalidade do mapa (geolocalização) no mobile é semelhante a funcionalidade na *web*, o mapa abre com a sua localização atual e mapeia outros usuários cadastrados que estão próximos; ao clicar na fazenda que aparece no mapa é aberto uma caixa com informações da fazenda (Figura 57).

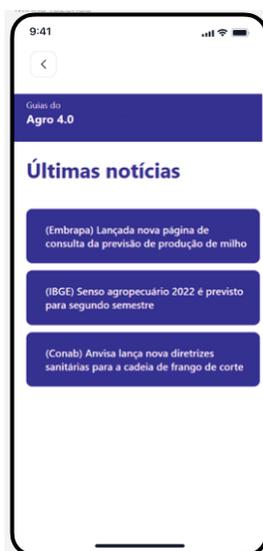
Figura 57 - Geolocalização mobile



Fonte: o autor (2023)

A última funcionalidade, são as Guias sobre Agronegócio 4.0 presentes também na versão *Web* (Figura 58).

Figura 58 - Guias do Agro 4.0 mobile



Fonte: o autor (2023)

A partir da seção seguinte, são tratados os dados coletados in-loco pelo pesquisador. A plataforma mobile não foi usada efetivamente pelos produtores de frango, uma vez que seria necessário publicar em uma loja de aplicativos. Os avicultores experimentaram as funcionalidades no dia a dia de coleta de dados por meio da plataforma web.

6 AVALIAÇÃO DA PROPOSTA

A avaliação da experiência do usuário, em muitos casos, se torna algo subjetivo, uma vez que envolve a percepção do usuário no uso, satisfação (ou insatisfação) com a experiência, ainda somam seus desejos, identificação e necessidades (MARQUES ET AL; 2019). Sendo assim, a avaliação da experiência do usuário é realizada a partir da triangulação de múltiplas experiências (técnica, de usuários como cliente e de objetivos do problema a ser resolvido).

Conforme as etapas do Quadro 14 que norteiam a coleta de dados, o formulário de avaliação da arquitetura IoT (Apêndice 4) foi enviado a granjeiros que puderam ter contato com a plataforma web, além de terem recebido também fotos e explicações de como funcionou a parte de *hardware* da arquitetura. Houveram 23 respostas, sendo 18 donos de granjas (pequenos produtores), 1 gestor de cooperativa e 4 funcionários que trabalham em granjas.

Dos 18 proprietários, 72% possuem ensino médio completo, 11% até o ensino fundamental completo e 17% com formação universitária (graduação). O único gestor de cooperativa que participou da pesquisa possui graduação e MBA na área da agronomia. Dos quatro funcionários de granja que responderam a pesquisa, três possuem ensino médio completo e um respondente está cursando administração.

Com relação ao questionário, todas as respostas variaram de “1 a 10”, o Quadro 24 apresenta as estatísticas descritivas das nove perguntas derivadas das dimensões dos elementos de UX visto no Quadro 15.

O elemento “interface” da dimensão Estético Visual obteve uma resposta de nota 2, que pelo questionário significava “interface confusa/feia”, e o próprio respondente colocou uma observação do qual afirma não ter muita afinidade com tecnologia (aplicativos e sites).

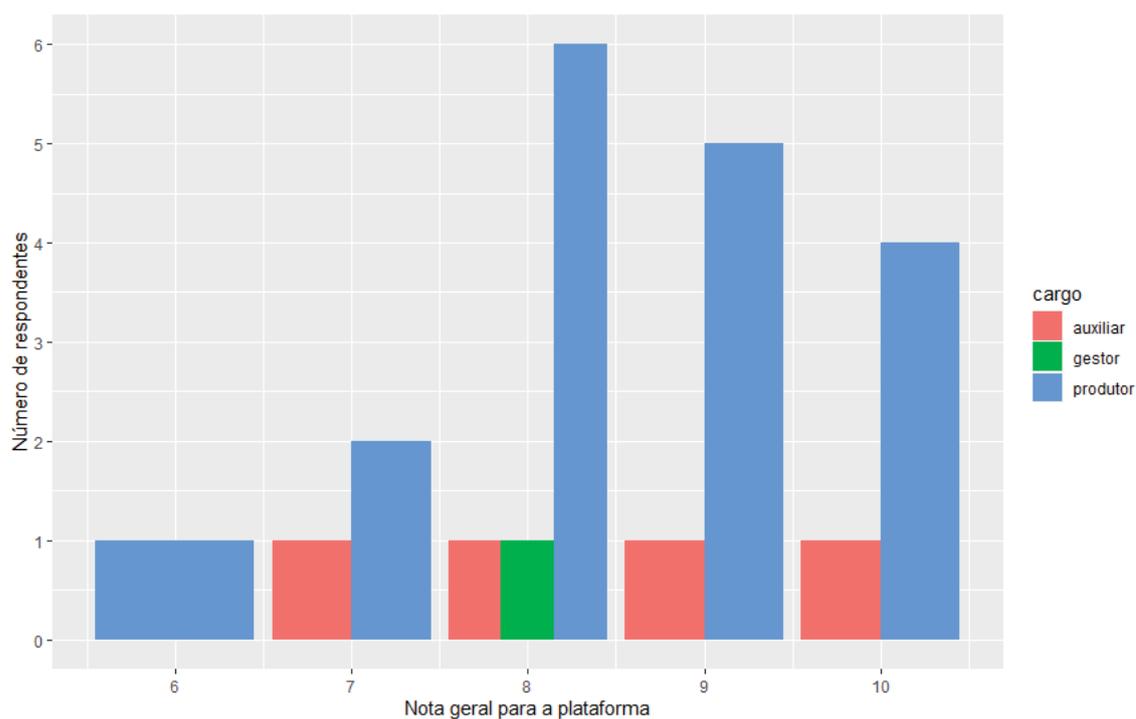
Quadro 24 - Estatísticas descritivas avaliação da proposta

Dimensão avaliada	Pergunta	Estatísticas descritivas				
		Média	Desvio Padrão	Mediana	Máximo	Minímo
Estético visual	3	7,31	1,69	7	10	2
	4	7,92	1,16	8	10	5
Estímulo/Engajamento	5	7,1	1,22	7	10	5
Usabilidade	6	8,22	0,99	8	10	5
	7	7,81	1,09	8	10	6
Aprendizado	8	8,25	1,46	8	10	4
	9	7,28	1,59	8	10	3
Utilidade	10	8,73	0,95	9	10	7
Experiência geral	13	8,52	1,12	8	10	6

Fonte: dados da pesquisa (2023)

A dimensão com a maior média e menor desvio padrão foi a de utilidade, principalmente na questão de que a interface é amigável, útil e que é necessário pouco esforço mental para operá-la. O Gráfico 1 ilustra a nota geral que os participantes deram para a plataforma.

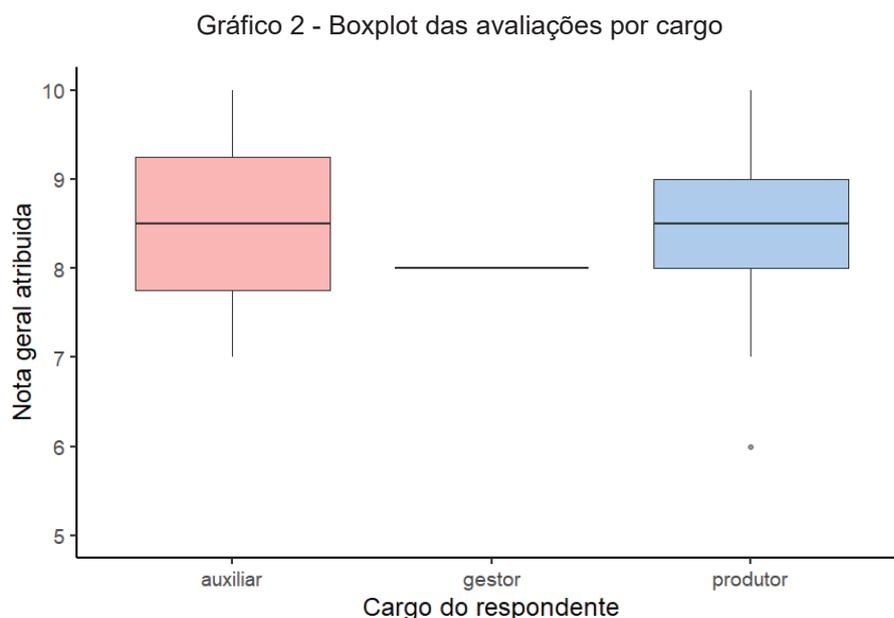
Gráfico 1 - Nota geral das avaliações da plataforma por cargo



Fonte: dados da pesquisa (2023)

Mais de 80% (19 respondentes) avaliaram a proposta com nota 8 ou superior, enquanto apenas quatro respondentes avaliaram com notas seis e sete.

No Gráfico 2 é possível ver o gráfico de *boxplot* comparando as avaliações por cargo.



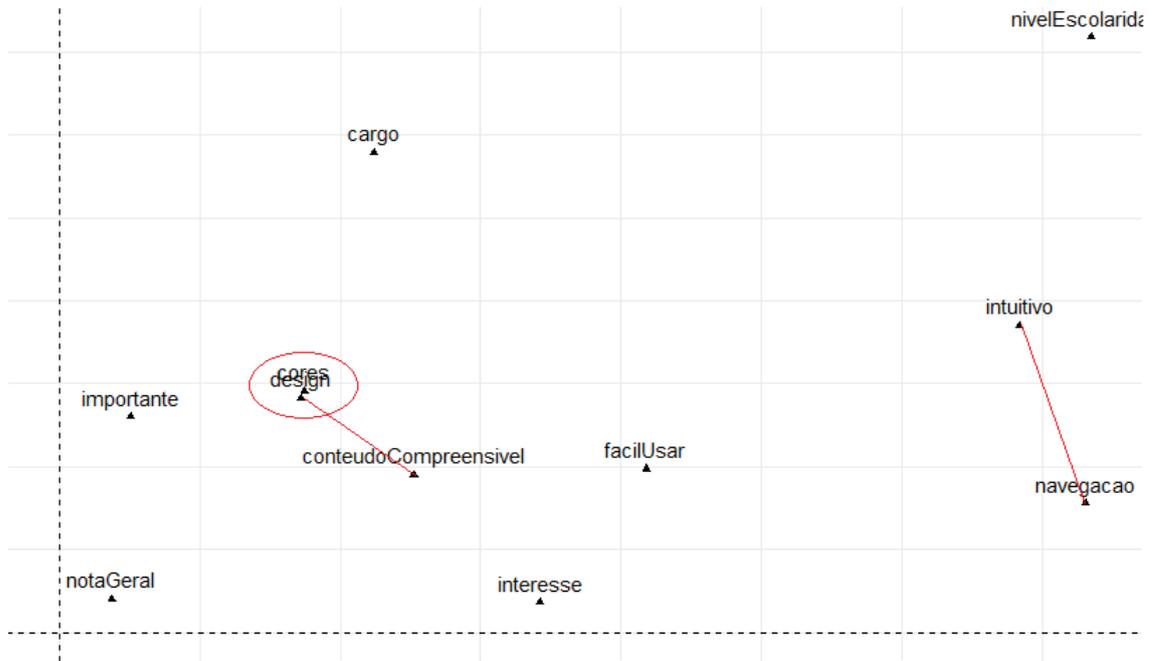
Fonte: dados da pesquisa (2023)

Dos auxiliares a menor avaliação foi 7 e a maior 10, sendo 8,5 a nota da mediana. O único gestor participante avaliou com nota 8. E dos produtores/proprietários foi notado um *outlier*, ou seja, uma nota de avaliação considerada estatisticamente afastada da média e da variância das respostas, a nota foi 6, enquanto as demais estiveram entre 7 e 10 com mediana de 8,5.

A última etapa da avaliação é a análise de correspondência múltipla, que tem por objetivo verificar a relação entre o cargo, nível de escolaridade e experiência geral no uso da plataforma proposta, de modo a nortear as melhorias na experiência do usuário. É importante que no desenvolvimento de um produto o público alvo seja reconhecido e suas expectativas sejam atendidas e consequentemente gerar engajamento no uso (DHARMAYANTI; BACHTIAR; WIBAWA, 2018).

O primeiro resultado da análise de correspondência múltipla é a relação gráfica entre os grupos, conforme visto no Gráfico 3

Gráfico 3 - Fatores da ACM



Fonte: dados da pesquisa (2023)

Analisando somente os fatores (colunas da base de dados) o primeiro gráfico indica que as variáveis “cores” e “design” estão intimamente relacionadas na percepção dos respondentes, e as variáveis “intuitivo” e “navegação” também apontam para uma relação entre elas. Esse resultado converge com os conceitos de UX descritos por Hussain et al (2018) onde afirmaram que a paleta de cores e como é desenhada a aplicação é relevante para uma sensação de conforto por parte do usuário, principalmente nos contrastes das cores entre os componentes tornando os conteúdos compreensíveis. Ademais, ter a percepção de que uma tela é intuitiva acontece à medida que o usuário realiza a navegação dentro da aplicação, observando como há um sentido entre a navegação e a disposição dos itens (GARRETT, 2010).

Os demais fatores, se comportaram de forma independente, e para conseguir explorar individualmente a categoria das respostas foi gerado outro gráfico, agora verificando a distribuição dessas categorias e a interação entre elas Gráfico 4.

Gráfico 4 - Distribuição gráfica das categorias



Fonte: dados da pesquisa (2023)

Na distribuição das categorias, os resultados são relacionados com as respostas, no caso do (A) como houve somente um gestor de cooperativa houve a relação do mesmo com seu nível de escolaridade, contudo o gráfico não trouxe nenhum elemento da experiência do usuário. No caso do (B) a análise mostra que quem possui escolaridade de nível do ensino fundamental achou razoavelmente fácil a navegação e ficou próximo do (C) que considerou pouco intuitiva e razoável navegação, o que pode indicar que o grau de instrução e familiaridade do usuário com a plataforma tem relação com sua formação.

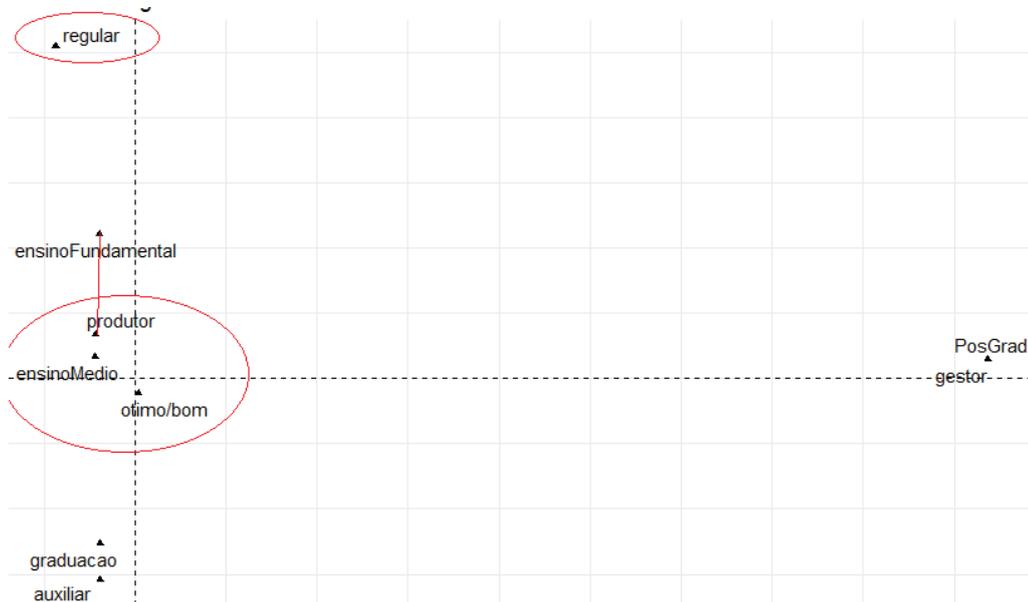
Por outro lado, na região (D) houve uma aproximação de quem considerou a plataforma razoavelmente importante com um design ruim e uma nota geral regular.

E por fim, no (E), onde houveram as maiores concentrações de respostas, os respondentes que citaram um excelente design, cores, que a aplicação é muito compreensível, muito intuitiva e fácil de usar avaliaram como nota geral a proposta da plataforma como “ótimo/bom”. O que reforça os resultados encontrados nas estatísticas descritivas da etapa 1 e 2 do protocolo de análise, a qual apontou um resultado quantitativo de experiência geral de nota 8,52.

Para conseguir verificar a experiência geral de uso da plataforma, foram removidas das análises as variáveis das dimensões de UX, considerando somente

três variáveis: “cargo”, “nivelEscolaridade” e “notaGeral”. E o resultado é mostrado no Gráfico 5.

Gráfico 5 - Distribuição gráfica das categorias com menos fatores



Fonte: dados da pesquisa (2023)

Com um número de fatores reduzidos, é possível observar que a avaliação regular não ficou próxima de nenhum cargo ou nível de escolaridade, ou seja, não foi identificada uma relação em quem avaliou a proposta como regular com seu cargo e/ou nível de escolaridade. Por outro lado, quem avaliou a proposta como ótimo/bom está relacionado com o cargo de “produtor” que é de porte pequeno e proprietário da granja (foco da pesquisa), o que pode apontar para um feedback positivo na experiência de uso da plataforma.

Na pesquisa de Mendoza et al (2021) é relatado que quem tem uma maior percepção de utilidade e valor em um produto digital normalmente é quem conhece a regra do negócio e todos os seus processos, conseqüentemente entende de maneira mais fácil os benefícios.

7 VALIDAÇÃO DA PROPOSTA

A fim de avaliar a comprovação da tese proposta, foi aplicado o protocolo de análise (visto na seção 3.7) nos dados coletados.

A seção foi dividida em três subtópicos, onde cada um vai apresentar os resultados de cada um dos cruzamentos de dados do Quadro 19.

A primeira base de dados a ser utilizada é dos relatórios gerenciais e de produção, tanto os dados fornecidos pela cooperativa, quanto os coletados com o PRODUTOR 1 e PRODUTOR 2. O primeiro passo é o tratamento dos dados coletados, eliminando qualquer valor nulo, distorcido e/ou não utilizado. O script com cada um dos procedimentos de tratamento dos dados pode ser visto no Apêndice 8.

Serão utilizadas duas bases de dados, a primeira com as variáveis do Quadro 17 com 14 variáveis (colunas) e 175 observações (linhas). Essa base está segmentada em cinco produtores, dois produtores controlem (os quais tiveram um Data Logger implementado) e três produtores que não tiveram qualquer tipo de intervenção, apenas os dados históricos do ano de 2022.

A Tabela 2 apresenta as estatísticas descritivas (média, desvio padrão, mediana, máximo e mínimo) dos cinco produtores avaliados.

Tabela 2 - Estatísticas descritivas por granja

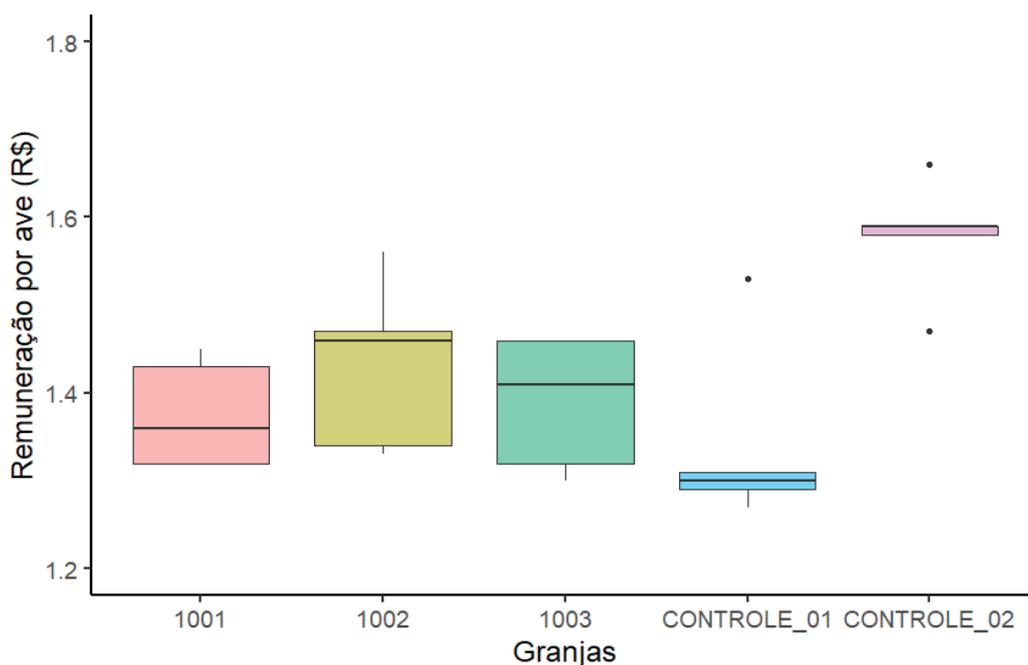
GRANJA	ESTATISTICA	por lote		por semana			por lote		
		R\$/ave	Produção	mortalida	refugo	IEP	Conv Alim	Temp_med	Umid_med
Controle_01 = PRODUTOR 2	media	R\$ 1,34	9600	59	14	373	1,3	27,03	66,54
	desv pad	R\$ 0,09	743	47	4,5	57	0,34	3,75	12,99
	mediana	R\$ 1,30	9485	47	16	373	1,33	27	63,8
	max	R\$ 1,53	11200	211	22	490	1,82	35	91
	min	R\$ 1,27	8452	21	7	247	1,1	21	44
Controle_02 = PRODUTOR 1	media	R\$ 1,58	26457	67	14	401	1,25	27,02	67,25
	desv pad	R\$ 0,06	654	85	4,4	49	0,32	4,1	14,36
	mediana	R\$ 1,59	26193	34	15	406	1,31	26	68
	max	R\$ 1,66	27550	301	22	482	1,68	36	94
	min	R\$ 1,47	25466	23	7	307	1,09	21	40
1001	media	R\$ 1,37	8631	43	14	359	1,36	27	66,86
	desv pad	R\$ 0,05	386	27	4,7	43	0,35	4,22	13,74
	mediana	R\$ 1,36	8510	36	15	364	1,33	27	67
	max	R\$ 1,45	9500	133	22	457	1,91	35	93
	min	R\$ 1,32	8124	23	7	236	1,25	21	41
1002	media	R\$ 1,43	8077	40	16	370	1,3	27,28	67,14
	desv pad	R\$ 0,08	384	26	4,3	48	0,34	4,1	14,92
	mediana	R\$ 1,46	8176	30	18	381	1,33	28	66
	max	R\$ 1,56	8700	121	22	443	1,92	35	94
	min	R\$ 1,33	7351	23	7	260	1,19	21,77	40
1003	media	R\$ 1,39	8256	40	14	379	1,28	27,27	66,74
	desv pad	R\$ 0,06	778	29	4,8	48	0,32	4,39	13,38
	mediana	R\$ 1,41	8607	29	14	390	1,29	26	65
	max	R\$ 1,46	9000	126	22	456	1,82	36	95
	min	R\$ 1,30	6641	24	7	262	1,17	21,2	42

Fonte: dados da pesquisa (2023)

Em linhas gerais da estatística descritiva, o PRODUTOR 1 (granja Controle_02) obteve a maior remuneração média por ave, produção e índice de eficiência produtiva, além de ter o menor índice de conversão alimentar. Isso pode indicar que seu sistema de automação e controle de ambiência pode estar corroborando para melhores resultados, além de uma maior infraestrutura e número de funcionários para se dedicarem ao manejo dos animais.

O Gráfico 6, mostra um *boxplot* por granja de acordo com as respectivas remunerações.

Gráfico 6 - Boxplot de remuneração por ave



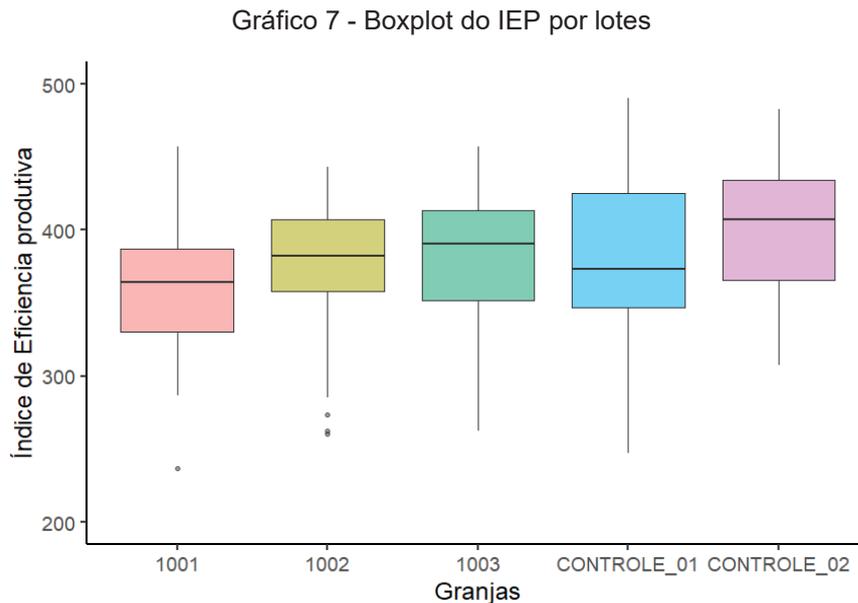
Fonte: dados da pesquisa (2023)

O aviário “Controle_02” possui maior aporte tecnológico na ambiência e aparece destacado com uma remuneração acima dos demais, inclusive com remunerações bem consistentes exceto com dois *outliers* sendo uma remuneração menor que R\$1,55 por ave e uma outra acima da mediana próximo de R\$1,70.

Por outro lado, o aviário “Controle_01”, pequeno produtor, teve no ano de 2022 remuneração menor que os demais do mesmo porte, ficando próximo a R\$1,30 por ave e com um *outlier* superior, com uma remuneração próximo de R\$1,60 por ave justamente no lote em que houve a intervenção tecnológica.

Esse resultado vai ao encontro com as pesquisas de Rafique et al. (2020) que em um ambiente controlado de frangos, com o auxílio de tecnologia também obtiveram desempenho superior.

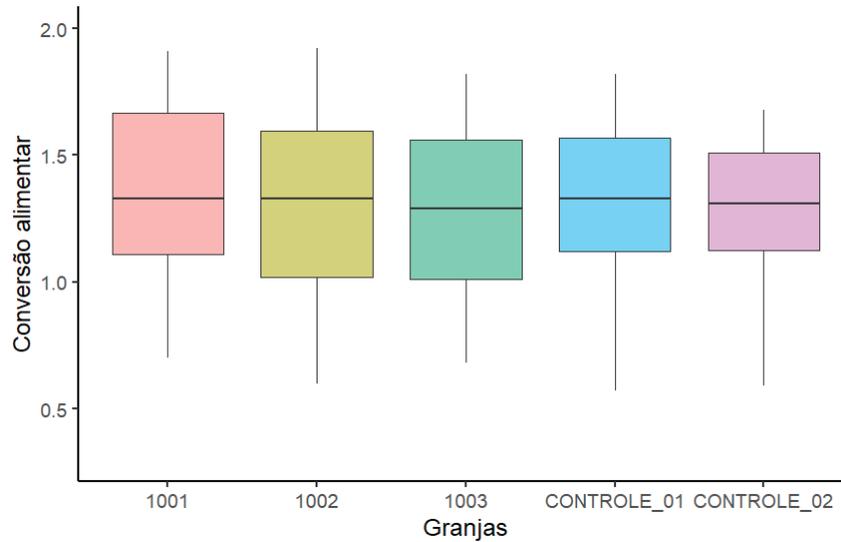
Apesar da remuneração da granja “Controle_01” ter ficado abaixo dos três aviários de mesmo porte, a eficiência produtiva entre elas é semelhante tomando com base a mediana do Gráfico 7.



Fonte: dados da pesquisa (2023)

Para investigar a menor remuneração foi gerado outro gráfico *boxplot* da conversão alimentar das aves por lote (Gráfico 8), o qual apontou que pela mediana o PRODUTOR 2 precisa de um maior volume de ração para conseguir aumentar em 1 Kg os frangos do lote.

Gráfico 8 - Boxplot da conversão alimentar por lotes

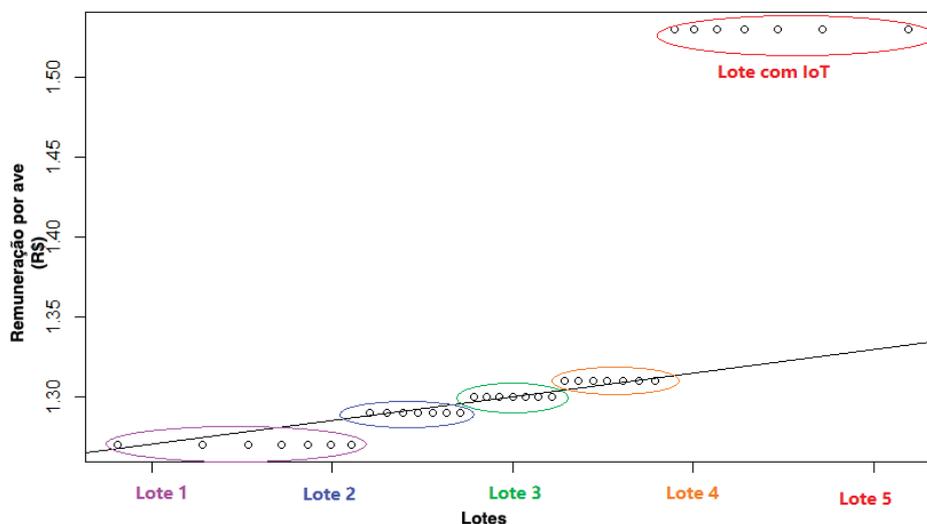


Fonte: dados da pesquisa (2023)

Para realizar o cruzamento dos dados do lote que teve o suporte da arquitetura IoT x lotes antecedentes à pesquisa, somente do PRODUTOR 2, é necessário verificar a distribuição dos dados (normal ou não normal) e sua homogeneidade.

No tocante à distribuição dos dados do PRODUTOR 2, o teste de normalidade apontou para uma distribuição não normal ($W= 0,612$), p-valor (0,000). Assim sendo, o teste a ser utilizado é o U de Mann-Whitney para comparar os grupos. O Gráfico 9 do tipo Q-Q ilustra como se deu o afastamento da distribuição normal e destaca o maior pagamento por ave no lote 5.

Gráfico 9 - Gráfico Q-Q remuneração por ave PRODUTOR 2



Fonte: dados da pesquisa (2023)

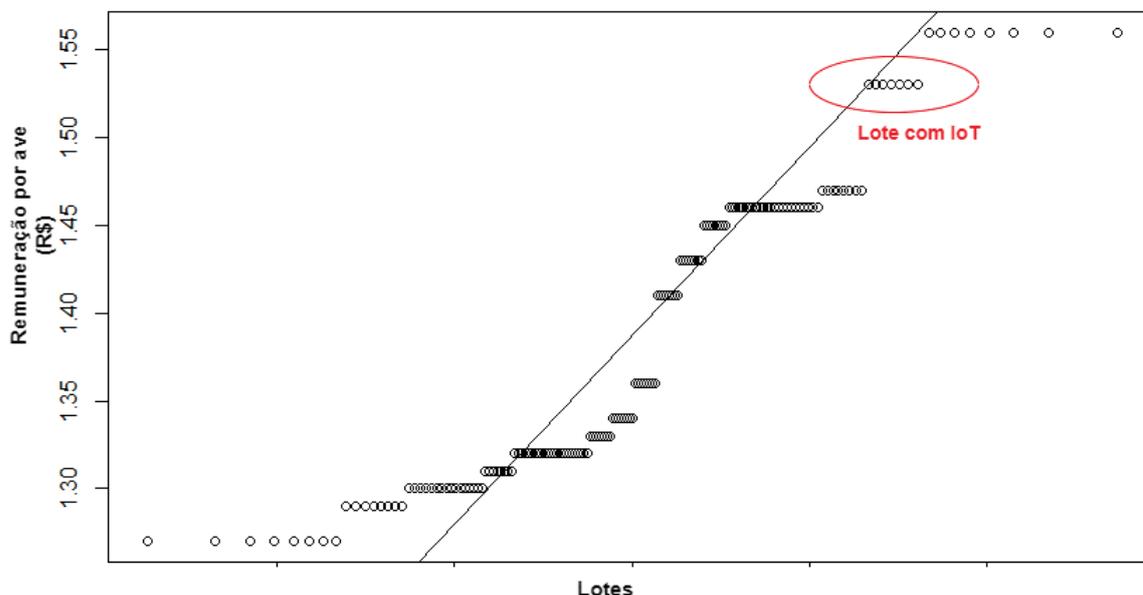
Sendo assim, o teste utilizado para fazer o cruzamento de dados entre o lote com a intervenção da arquitetura IoT do PRODUTOR 2 x os quatro Lotes do PRODUTOR 2 de antes da pesquisa é o teste U de Mann-Whitney.

O teste U de Mann-Whitney confirmou a diferença estatisticamente significativa entre o lote produzido com o suporte da arquitetura IoT x os lotes sem esse suporte ($W=0$, p -valor = 0,000).

O segundo cruzamento de dados é entre: Resultados sensorizados do PRODUTOR 2 x Resultados não sensorizados de todos os produtores de pequeno porte.

A distribuição dos dados entre o lote com IoT e os lotes sem IoT apontaram para uma distribuição não normal ($W= 0,891$), p -valor (0,000). O Gráfico 10 Q-Q ilustra que o lote com IoT conseguiu o segundo maior pagamento do ano de 2022 – R\$1,53, ficando com menor remuneração apenas para um lote de outro pequeno produtor – R\$ 1,56.

Gráfico 10 - Gráfico Q-Q remuneração por ave pequenos produtores



Fonte: dados da pesquisa (2023)

Como a distribuição dos dados se comportou como não normal, foi novamente utilizado o teste U de Mann-Whitney, que confirmou a diferença estatisticamente significativa entre o lote do PRODUTOR 2 produzido com o suporte da arquitetura IoT x os lotes dos pequenos produtores ($W=56$, p -valor = 0,000).

Diante disso, com os resultados, é possível afirmar categoricamente que a arquitetura IoT apresentou bom resultado comparado aos demais lotes produzidos.

Sendo que na primeira comparação, os resultados somente entre os lotes do PRODUTOR 2 mostraram um aumento médio de ganho por ave de R\$1,33 para R\$1,53, enquanto a média de todos os pequenos produtores permaneceu próximo a R\$1,40.

A segunda base de dados conta com dados do *Data Logger* implementado no Produtor 1 e seus dados próprios do sistema comercial já utilizado. A base de dados conta com 5 variáveis (colunas) e 1080 observações (linhas), com dados dos sensores de todos os dias de produção de um lote.

A primeira etapa de análise dessa base de dados são as estatísticas descritivas, a Tabela 3 sintetiza os resultados dos cálculos descritivos entre o sistema comercial de automação e o Data Logger da arquitetura IoT proposta

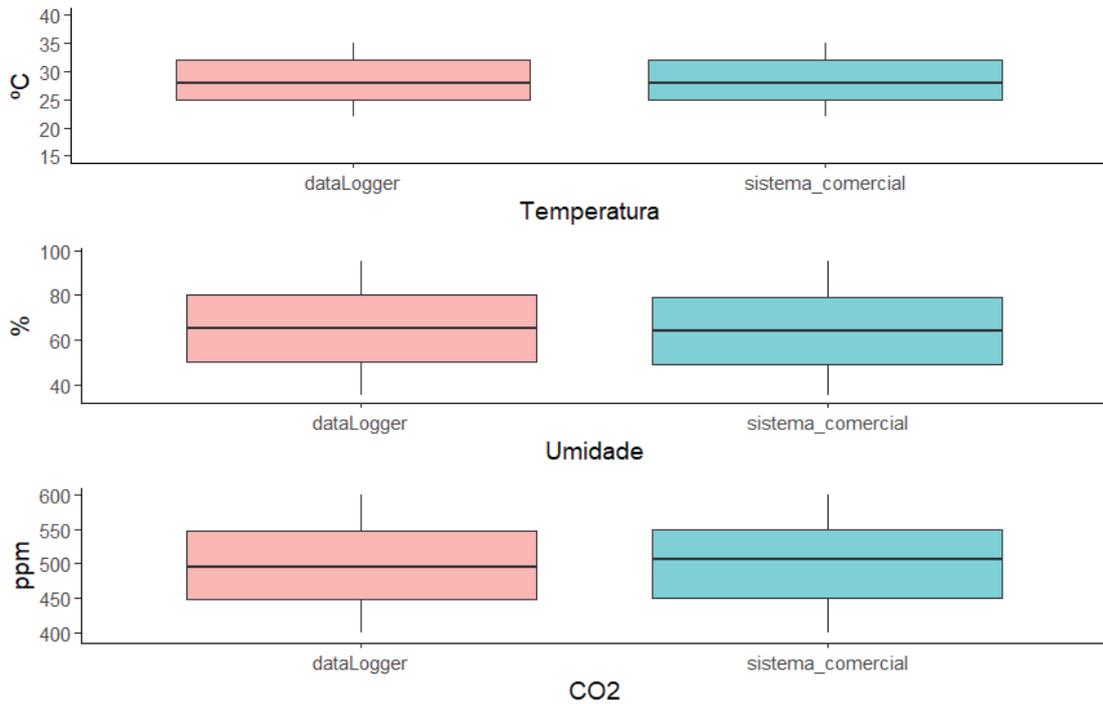
Tabela 3 – Estatísticas descritivas da mensuração

Grandeza mensurada	Estatística	Data Logger	Sistema de automação
Temperatura (°C)	média	28,52	28,31
	desv pad	4,14	4,01
	mediana	28	28
	máx	35	35
	mín	22	22
Umidade (%)	média	65,16	64,55
	desv pad	17,21	17,24
	mediana	65	64
	máx	95	94
	mín	35	33
CO2 (ppm)	média	497	502,31
	desv pad	58,46	58,17
	mediana	495,5	507
	máx	680	697
	mín	400	400

Fonte: dados da pesquisa (2023)

Os resultados apontam uma pequena variação entre os valores medidos, a diferença de temperatura entre a média do *Data Logger* e o sistema de automação foi de 0,21°C, enquanto que o de umidade foi de 0,61% e de CO2 um valor de 5,31ppm. Os *boxplots* da Figura 59 indicam uma igualdade entre os sensores da arquitetura IoT com o sistema de automação já utilizado pela granja nos resultados dos valores mínimos, máximos, mediana e não apresentaram nenhum dado fora do padrão.

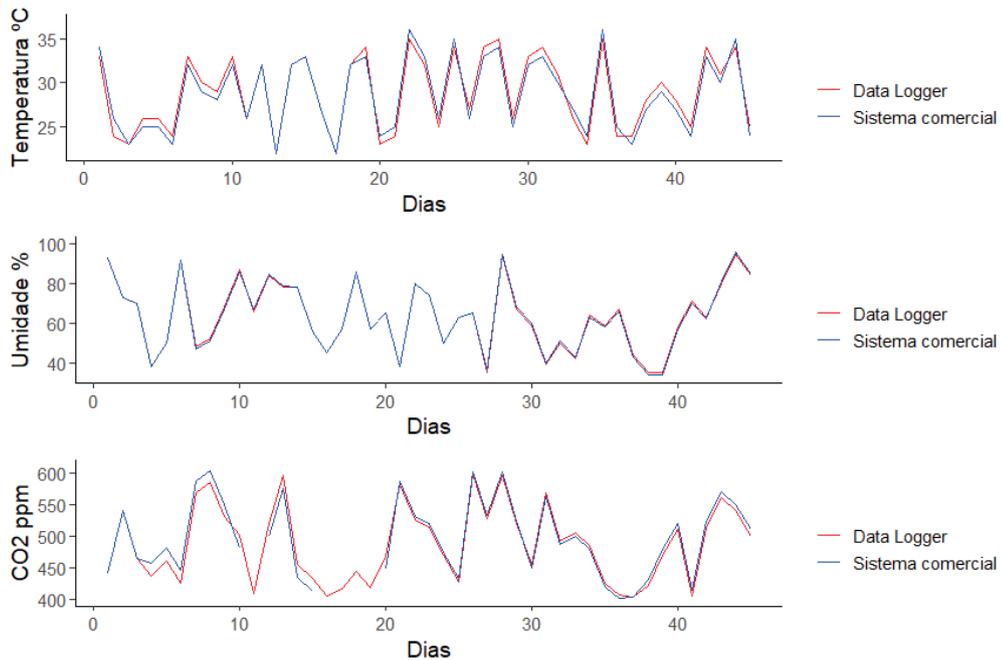
Figura 59 - Boxplots da comparação entre sistemas



Fonte: dados da pesquisa (2023)

A Figura 60 com os os gráficos em linha ratificam os resultados encontrados nos *boxplots* da Figura 59, onde a mediana, máximos e mínimos ficaram nos mesmos valores.

Figura 60 – Gráfico em linha da comparação entre os sistemas



Fonte: dados da pesquisa (2023)

As linhas sobrepostas mostram a similaridade das aferições de temperatura, umidade e dióxido de carbono no ambiente, tanto do sistema comercial quanto do Data Logger da arquitetura IoT.

Seguindo o protocolo de análise, o teste de normalidade apontou que a distribuição dos dados não se afasta da distribuição normal $W=0,879$ e $p\text{-valor} = 0,767$. A homogeneidade da variância ficou em $K = 0,588$ e $p\text{-valor} = 0,442$ confirmando também que os dados são homogêneos. Assim sendo, é possível utilizar o teste *t* de *student* para confirmar que não há diferença estatisticamente significativa entre os dois sistemas de medição, esse teste atendendo ao cruzamento de dados número 3 do Quadro 19.

Os resultados do teste *t* foram: $t= 0,819$ e $p\text{-valor} = 0,412$ o que indica que não há diferença estatisticamente significativa entre os dados dos dois registradores, ou seja, nesse experimento o *Data Logger* da proposta IoT obteve desempenho semelhante ao do sistema comercial já utilizado pelo PRODUTOR 1. Incidindo principalmente na confiabilidade dos sensores, respeitando os prazos de calibração descritos pelos fabricantes dos sensores.

8 VISÃO CONCEITUAL DA ARQUITETURA IOT

A Internet das Coisas é um dos motores da quarta revolução industrial, impactando como são gerenciados os meios de produção, tomadas de decisão, impactando inclusive nas formas de comunicação, nas relações sociais e até nos impactos ambientais.

Com a premissa de permitir conectividade, interação e integração de objetos inteligentes endereçáveis que colaboram uns com os outros, essa tecnologia tem impactado até nos métodos de desenvolvimento de software, levando as empresas do ramo a se atentar não somente ao *software*, mas também a conectividade e capacidade de processamento e do funcionamento dos *hardwares* das “coisas”.

Dessa forma, considerando as particularidades exigidas pelos sistemas de IoT, os desafios relacionados ao desenvolvimento de sistemas de IoT devem ser considerados desde o nível arquitetural.

O modelo proposto está dividido em três níveis, sendo o primeiro responsável pela parte física:

1 - O *Datalogger* (registrador de dados) é instalado localmente no aviário, necessitando apenas de conexão com a rede elétrica e um ponto de internet. O objetivo é a presença física na granja, são os hardwares provedores das fontes de dados (sensores). Por meio da *internet* é possível estabelecer a comunicação entre a granja e o servidor (garantindo a interoperabilidade do sistema), para a pesquisa foi utilizado a rede *wi-fi* da propriedade, contudo, caso não haja um ponto de internet no local é possível utilizar um microcontrolador com entrada para chip GSM (tecnologia para conexão móvel) suportando 2G, 3G e 4G. Tendo como única diferença a conexão com a rede, o restante da lógica do *firmware* não é afetada.

A importância da precisão dos sensores está ligada com a qualidade dos dados, ou seja, que os dados sejam fidedignos e válidos para serem persistidos e utilizados nas outras instâncias da arquitetura. Os microcontroladores devem ser identificados unicamente; diante do grande número de “coisas” que poderão estar interconectadas, um identificador único permitirá rastrear/monitorar cada item no ambiente físico, garantindo precisão no processo de gestão, não sendo possível o *hardware* de um usuário ser identificado como de outra pessoa. Essa etapa foi mostrada na pesquisa, com a inclusão do endereço de IP de cada hardware, identificando cada dado enviado.

2 – O nível *Back-end* possui a função de garantir a interação entre o registrador de dados e o usuário (por meio do *front-end*), para que isso aconteça é necessário que haja o gerenciamento da conexão (comunicação) entre usuário e máquina, captura, envio e persistência dos dados, para posterior consumo dos dados pelo *front-end*.

A principal função da comunicação é viabilizar a integração e interoperabilidade do sistema de *Internet* das Coisas. Para os atuadores, que utilizam o protocolo de *websockets*, a autenticação do dispositivo tem a finalidade de aceitar novas conexões, na prática, o servidor e os *Data Loggers* estarão interconectados por meio da Internet e ocorrerão solicitação e respostas de ambas as partes.

A autenticação certificará que a requisição partiu de uma fonte conhecida (usuário com permissão e portando um *token* válido); assim é concedida permissão de acesso ao usuário (segurança) previamente cadastrado no servidor.

A captura dos dados é realizada fisicamente pelos sensores e o algoritmo é um *script* de repetição, o qual coleta o dado do sensor via comunicação serial e envia ao microcontrolador, que por sua vez realiza uma requisição *http* do tipo “*post*” para armazenar no banco de dados, esse procedimento é mediado pela API responsável pelo método “*.store*” da rota dos sensores. Em síntese, a persistência é o componente responsável pelo armazenamento dos dados, dessa forma, entende-se que para atender às funcionalidades da aplicação será usado um sistema de gerenciamento de banco de dados (SGBD) com alta disponibilidade, escalabilidade e desempenho do tipo NoSQL (*Not Only SQL*). O *MongoDB* é capaz de gerir grandes volumes de dados, realizar leitura e gravação de dados rapidamente, escalável e gratuito.

No tocante ao modelo de dados, o componente utiliza uma API para representação dos dados em JSON. A escolha do padrão de serialização de objeto se dá pelo fato do JSON ser amplamente utilizado e compatível com várias linguagens de programação, gerando arquivos exportáveis e seguros.

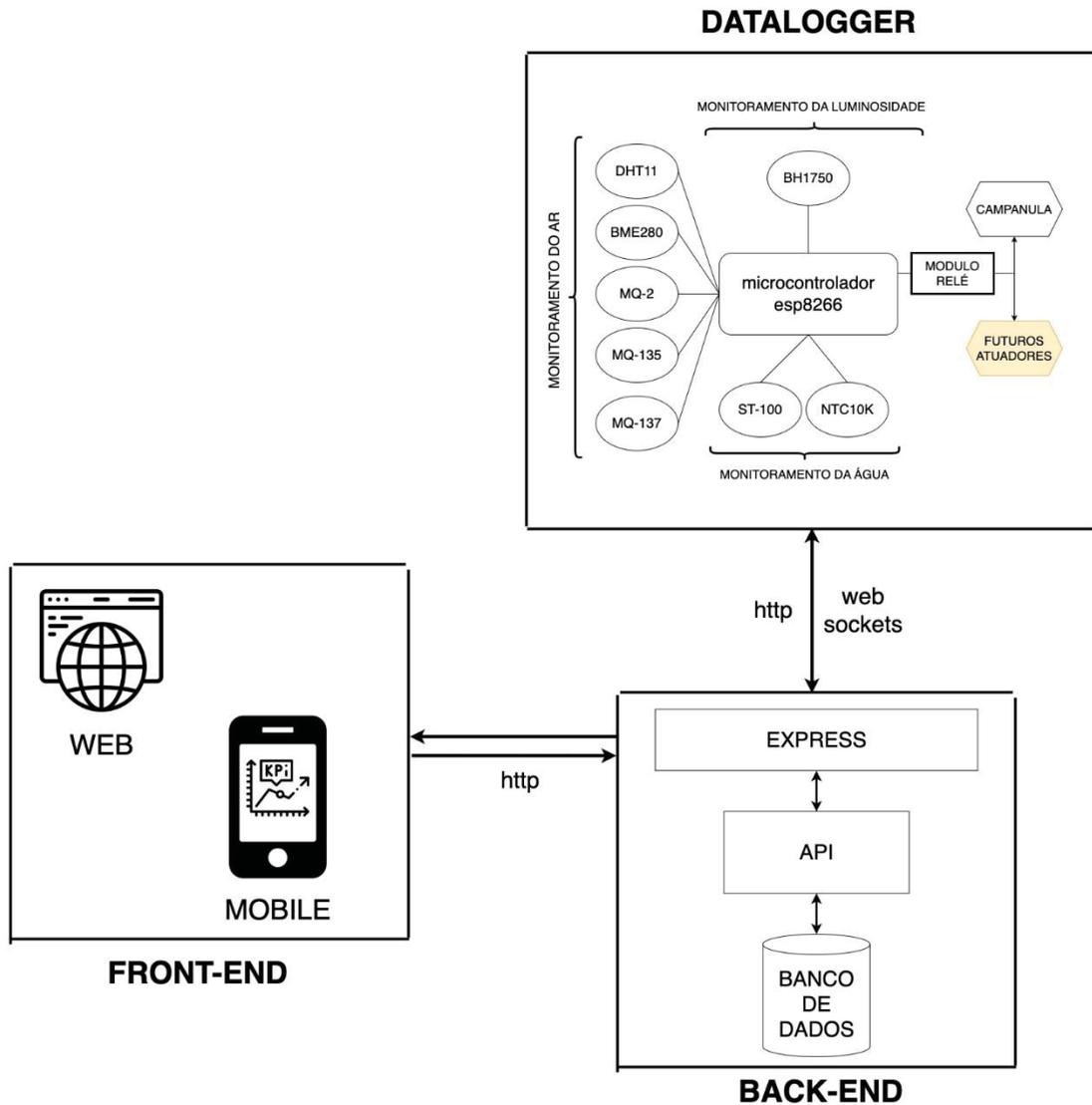
3 – Nível de *Front-end* pode ser considerado o mais próximo dos usuários (avicultores) que fazem uso da arquitetura IoT. Ele é responsável por disponibilizar dados e informações aos stakeholders do sistema por meio das plataformas *web* e *mobile*.

É na camada visual também que a aplicação solicita dados previamente armazenados e tratados nas rotas da aplicação, de modo a fornecer gráficos e indicadores do que está acontecendo em tempo real na granja. Componentes

responsáveis por facilitar a visualização dos dados e tomada de decisão por parte dos usuários.

A Figura 61 apresenta a visão conceitual da arquitetura, considerando os três níveis existentes:

Figura 61 – Modelo conceitual da Arquitetura IoT



Fonte: o autor (2023)

Com o isolamento dos níveis da arquitetura é possível atualizar tecnologias ou componentes eletrônicos da camada física, mantendo as interfaces (*back-end* e *front-end*), pois os módulos de comunicação e de serviços não precisam ser alterados. Assim como é possível atualizar, trocar o modelo de dados, com os devidos ajustes na API sem comprometer o registrador de dados, uma vez que cada nível é

responsável exclusivamente por sua função, não havendo uma dependência entre elas.

Os sistemas IoT de “alto custo” – sistemas comerciais – em geral, possuem mais recursos. Por exemplo, um sistema de alto custo pode incluir sensores mais precisos e confiáveis (sensores proprietários), capacidade maior de processamento de dados, conectividade mais rápida e confiável.

Além de uma interface de usuário mais amigável e recursos de segurança mais avançados. Outro fator, que pode diferenciar e encarecer um sistema IoT é a oferta pelo suporte técnico e manutenção mais abrangentes, bem como opções de personalização.

Em geral, a escolha entre um sistema IoT de alto custo e um de baixo custo dependerá das necessidades específicas de um determinado projeto ou aplicação.

Contudo, em um cenário nacional, onde grande parte da produção de alimentos estão ligadas diretamente com pequenos produtores (pequenas famílias), tendo a sucessão familiar como um dos obstáculos para a continuidade dos negócios, a Internet das Coisas pode ser um suporte nessa transição.

Outra barreira para a digitalização de granjas, está relacionado ao financiamento para os investimentos, uma vez que não basta apenas a inclusão de equipamentos que sensoriam, armazenam e gerenciam dados, são oferecidas soluções nas quais requerem uma reforma completa na estrutura dos aviários.

Conseguir consolidar um protótipo comercial, escalável e de baixo custo pode ajudar pequenos avicultores a terem seu sistema IoT para consumo local e compartilhar dados com outros pequenos produtores, uma vez que compartilham quase que das mesmas dificuldades, dúvidas e desafios do setor.

Uma vez que a modernização dos sistemas produtivos atrai as atenções das novas gerações. Isso pode se tornar um trunfo para a continuidade dos negócios, engajando os atores envolvidos.

No entanto, o custo de implementação desses sistemas tem sido um dos principais obstáculos para muitas empresas, especialmente as de pequeno e médio porte.

A proposta dessa tese em disponibilizar um sistema IoT desenvolvido de ponta a ponta com tecnologias livres, sem a dependência de sistemas operacionais, ou mensalidades pode ajudar a baratear os custos de implementação e manutenção. Os sensores e atuadores podem ser substituídos mantendo os módulos, assim como

o microcontrolador. Tendo o *firmware* em um repositório público e permanente permitindo o acesso para *download* e compilação. Inclusive para futuras contribuições de outros pesquisadores da área, os quais podem aperfeiçoar ou modificá-los para utilizar em contextos específicos.

Essas práticas também contribuem para que haja maior acesso a *hardwares* e *softwares* completos disponíveis para a comunidade científica, descentralizando a propriedade intelectual de empresas desenvolvedoras de sistemas. Uma vez que em muitas publicações científicas boa parte dos pesquisadores acabam desenvolvendo apenas uma parte do sistema para conseguir coletar dados, deixando de analisar as possibilidades do que é possível fazer com os mesmos. Com as regulamentações que travam o compartilhamento de dados pessoais, fica cada vez mais difícil conseguir dados de terceiros com o objetivo de pesquisa.

A modelagem dos dados para compartilhamento coletivo dos dados também é uma contribuição, inclusive para órgãos de senso como o IBGE, EMBRAPA, IDR-PR (Instituto de Desenvolvimento Rural do Paraná), os quais podem fazer parcerias com a rede de usuários do sistema para que consigam ter uma noção do desempenho regional uma vez que a plataforma integra dados de geolocalização dos produtores.

Em síntese, a democratização da indústria 4.0 por meio de sistemas IoT de baixo custo também tem um grande impacto na economia e no desenvolvimento social. Através da implementação desses sistemas, as empresas podem criar novos empregos e oportunidades para a comunidade local, além de melhorar a qualidade dos produtos e serviços que oferecem aos clientes.

A disponibilidade de sistemas IoT de baixo custo pouco a pouco tem ajudado a viabilizar a indústria 4.0, permitindo que mais empresas tenham acesso à tecnologia avançada e possam competir de maneira mais eficiente no mercado. Isso não só impulsiona a inovação e a produtividade, mas também tem um impacto positivo na economia e no desenvolvimento social.

9 CONSIDERAÇÕES FINAIS

Essa pesquisa tinha como objetivo a modelagem, construção e validação de uma arquitetura baseada em IoT de baixo custo capaz de orientar e aportar a gestão de pequenos produtores de frango de corte, inserindo-os no contexto do Agronegócio 4.0 de modo a permitir melhorar seus processos decisórios; identificando parâmetros críticos da ambiência animal, como a ventilação, temperatura, umidade, presença de gases nocivos à saúde animal, temperatura da água e sua turbidez.

Ao final da modelagem física e computacional (*hardware* e *software*) da arquitetura IoT foi possível consolidar a proposta e validá-la estatisticamente na perspectiva de ganhos monetários, que é derivado de resultados de eficiência produtiva, gasto energético e de insumos dentre outras métricas do ramo. Além disso, houve também a dimensão de experiência de usuário, na qual a interface da plataforma foi enviada para produtores que pudessem explorá-las e dar um *feedback* em forma de avaliação.

Então, além das contribuições de âmbito científico-teórico, a pesquisa também contribuiu na perspectiva científico -prático gerando uma aproximação com a sociedade e suas demandas, fomentando um diálogo e gerando resultados práticos para um setor importante da economia nacional.

Com a arquitetura IoT proposta, pequenos produtores de frango de corte podem iniciar o processo de transformação digital melhorando seus resultados de várias maneiras, compartilhar seus dados com outros produtores de mesmo porte, gerando uma base de conhecimento para os próprios interessados no negócio. Esse compartilhamento de indicadores, pode ser utilizado como parâmetro para a criação de políticas públicas para o segmento, fortalecer o setor e democratizar o acesso a tecnologias da Indústria 4.0.

Alguns dos ganhos com a difusão da arquitetura IoT proposta pela pesquisa são listados:

- Monitoramento da granja em tempo real: com a coleta de dados em tempo real sobre as condições dos animais, o ambiente de criação e a gestão da produção, a plataforma IoT ajuda os produtores a tomarem decisões, melhorando assim a eficiência e a qualidade da produção e deixando cada vez mais de depender da intuição ou da tradição.

- Otimização de recursos: O monitoramento remoto dos sistemas de temperatura, umidade e ventilação pode ajudar a reduzir o uso de recursos elétricos da granja, por exemplo.

- Automatização: Com o módulo de atuadores é possível que haja a integração para que o produtor consiga tomar ações mesmo distante da granja, como por exemplo o acionamento de ventilação, cortinas, aspersão de água ou até outros módulos não descritos nessa pesquisa, uma vez que a camada de comunicação está pronta no *back-end* da aplicação.

- Rastreabilidade: A proposta contempla no seu modelo de dados, a possibilidade de rastreio das condições ambientais que o lote teve ao longo das semanas. E isso pode inclusive ser trabalhado posteriormente para uma integração com as cooperativas, agroindústrias para que venha o histórico digitalizado e integrado aos relatórios gerenciais, prática que hoje em dia é realizada manualmente pelos granjeiros nas anotações das condições do ambiente.

No tocante aos resultados das entrevistas e avaliação, os criadores de frangos de corte atribuíram importância ao bem-estar animal e consideram os princípios de “boa alimentação”, “bom alojamento” e “boa saúde” como os mais importantes para conseguir um nível adequado de remuneração, conseqüentemente poucas perdas e desperdícios em água, ração e insumos medicamentosais quando necessários.

Por fim, a Internet das Coisas (IoT) tem um potencial enorme para transformar a forma como os pequenos produtores de corte gerenciam suas operações. A integração de sensores com sistemas multiplataforma permite a coleta de dados em tempo real sobre as condições dos animais, o ambiente de criação e a gestão da produção.

9.1 LIMITAÇÕES

A primeira limitação da pesquisa foi o fato de não ter conseguido implementar a arquitetura em mais propriedades e tendo somente 1 lote, a ausência de uma pesquisa qualitativa para verificar os elementos que fizeram um pequeno produtor conseguir um valor maior por cabeça de frango, provavelmente foram questões de manejo e alojamento.

9.2 TRABALHOS FUTUROS

Apesar das limitações, surgem também algumas oportunidades de expansão do projeto e pesquisas futuras, que são listadas a seguir:

Modelar um sistema para controle do vazão sanitário entre um lote de frangos e outro, verificando se é possível reduzir o tempo de sanitização ou de mensurar a qualidade e sanidade ambiental.

Implementar a arquitetura em uma granja com ambiente controlado e comparar mais lotes com frangos em condições iguais.

Realizar um estudo qualitativo para explorar quais elementos específicos de manejo podem contribuir com o melhor desempenho produtivo.

Implementar um algoritmo preditivo que consiga encontrar o melhor momento para sugerir a intervenção na granja, considerando dados climáticos internos e externos.

REFERÊNCIAS

- ABDI, H; VALENTIN, D. Multiple correspondence analysis. **Encyclopedia of measurement and statistics**, v. 2, n. 4, p. 651-657, 2007.
- ABIOYE, E. A. et al. IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation experiment. **Information Processing in Agriculture**, 2020.
- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9241-171:2018**. Ergonomia da interação humano-sistema: Orientações sobre acessibilidade de software, elaborada pela Comissão de Estudo Especial de Ergonomia da Interação Humano-sistema - Requisitos e diretrizes. Rio de Janeiro. 2018.
- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27701:2019**. Técnicas de segurança — Extensão da ABNT NBR ISO/IEC 27001 e ABNT NBR ISO/IEC 27002 para gestão da privacidade da informação — Requisitos e diretrizes. Rio de Janeiro. 2019.
- ABPA – Associação Brasileira de Proteína Animal. **Relatório Anual de produção Brasil**. Disponível em: https://abpa-br.org/wp-content/uploads/2020/05/abpa_relatorio_anual_2020_portugues_web.pdf . Acesso em: 05 de março de 2021
- ABREU, V. M. N; ABREU, P. G. **Os desafios da ambiência sobre os sistemas de aves no Brasil**. 2011.
- ADAFRUIT. **DHT11 basic temperature-humidity sensor + extras**. Disponível em: <https://www.adafruit.com/product/386> . Acessado em 19 de setembro de 2022
- ALBERTIN, A. L; M. ALBERTIN, R. M. A internet das coisas irá muito além as coisas. **GV EXECUTIVO**, v. 16, n. 2, p. 12-17, 2017.
- ALFARIDZI, M. D; YULIANTI, L. P. UI-UX design and analysis of local medicine and medication mobile-based apps using task-centered design process. In: **2020 International Conference on Information Technology Systems and Innovation (ICITSI)**. IEEE, 2020. p. 443-450.
- ALIWARGA, H. K; SATRIATAMA, A. H; PRATAMA, B. F. A. Performance comparison of fleet management system using IoT node device based on MQTT and HTTP protocol. In: **AIP Conference Proceedings**. AIP Publishing LLC, 2020. p. 020009.
- ALMEIDA, R. M. A; MORAES, C. H. V; SERAPHIM, T. F. P. **Programação de Sistemas Embarcados: Desenvolvendo Software para Microcontroladores em Linguagem C**. Elsevier Brasil, 2017.

AL-QASEEMI, S. A. et al. IoT architecture challenges and issues: Lack of standardization. In: **2016 Future technologies conference (FTC)**. IEEE, 2016. p. 731-738.

AL-SARAWI, S. et al. Internet of Things (IoT) communication protocols. In: **2017 8th International conference on information technology (ICIT)**. IEEE, 2017. p. 685-690.

AMALI, C.; GURU, K.; SRIDEVI, D. IoT BASED SMART LOGISTICS MANAGEMENT SYSTEM USING GPS AND GSM **International Journal of Advanced Research in Engineering and Technology (IJARET)**, v. 11, p.3035-3041, 2020.

AMARAL, A.G. et al . Efeito do ambiente de produção sobre frangos de corte sexados criados em galpão comercial. **Arquivo Brasileiro de Medicina Veterinária e Zootecnia**, Belo Horizonte, v. 63, n. 3, p. 649-658. 2011.

AQUINO, C; GANDEE, T. **Front-End Web Development: The Big Nerd Ranch Guide**. Pearson Technology Group, 2016.

ARDUINO PRO. **Documentação Arduino: Guia Geral**. Disponível em: < <https://www.arduino.cc/pro/tutorials/portenta-h7> > . Acesso em 03 de Mai. 2021.

ARSHAD, J. et al. Deployment of Wireless Sensor Network and IoT Platform to Implement an Intelligent Animal Monitoring System. **Sustainability**, v. 14, n. 10, p. 6249, 2022.

ARTUZO, F. D; SOARES, C; WEISS, C. R. Inovação de processo: O impacto ambiental e econômico da adoção da agricultura de precisão. **Espacios**, v. 38, n. 2, p. 1-6, 2017.

ASTILL, J. et al. Smart poultry management: Smart sensors, big data, and the internet of things. **Computers and Electronics in Agriculture**, v. 170, p. 105291, 2020.

BANDYOPADHYAY, D; SEN, J. Internet of things: Applications and challenges in technology and standardization. **Wireless personal communications**, v. 58, n. 1, p. 49-69, 2011.

BAÑUELOS, E. L. et al. A Systematic Review for Cognitive State-Based QoE/UX Evaluation. **Sensors**, v. 21, n. 10, p. 3439, 2021.

BARUA, T; DOSHI, R; HIRAN, K. K. Mobile Applications Development. In: **Mobile Applications Development**. De Gruyter, 2020.

BASSI, A; LANGE, S. The need for a common ground for the IoT: The history and reasoning behind the iot-a project. In: **Enabling Things to Talk**. Springer, Berlin, Heidelberg, 2013. p. 13-16.

BATALHA, M. O. **Gestão agroindustrial**. 4ªed. São Paulo: Editora Atlas, 2021.

BERTOLINI, C. et al. **Linguagem de programação I**. 2019.

BICA, M. R. R. et al. Sistema de monitoramento de temperatura em silo de armazenamento de grãos com comunicação sem fio. **Brazilian Journal of Development**, v. 7, n. 5, 2021.

BIØRN-HANSEN, A et al. An empirical investigation of performance overhead in cross-platform mobile development frameworks. **Empirical Software Engineering**, v. 25, p. 2997-3040, 2020.

BOLFE, É. L. et al. Precision and Digital Agriculture: Adoption of Technologies and Perception of Brazilian Farmers. **Agriculture**, v. 10, n. 12, p. 653, 2020.

BONI, V; QUARESMA, S. J. Aprendendo a entrevistar: como fazer entrevistas em Ciências Sociais. **Em tese**, v. 2, n. 1, p. 68-80, 2005.

BOSCH. **BME280 Data Sheet**. 2020. Disponível em: < <https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf>>. Acesso em: 01/10/2018.

BRASIL. Decreto nº 9.854, de 24 de junho de 2019. Institui o Plano Nacional de Internet das Coisas e dispõe sobre a Câmara de Gestão e Acompanhamento do Desenvolvimento de Sistemas de Comunicação Máquina a Máquina e Internet das Coisas., Brasília, 2019. Disponível em: http://www.planalto.gov.br/ccivil_03/ Ato2019-2022/2019/Decreto/D9854.htm . Acesso em: 07 de mar. 2020.

BRASIL. Secretária de Agricultura e Pecuária do estado do Paraná. **Produção de frango de corte do estado do Paraná 2021**. Disponível: <https://www.aen.pr.gov.br/modules/noticias/article.php?storyid=111514&tit=Parana-produz-um-terco-da-carne-de-frango-do-Brasil> . acessado em 03 de maio de 2021.

BREJE, A. R. et al. Comparative study of data sending methods for XML and JSON models. **International Journal of Advanced Computer Science and Applications**, v. 9, n. 12, 2018.

BURTON, R. J. F; FISCHER, H. The Succession Crisis in E uropean Agriculture. **Sociologia Ruralis**, v. 55, n. 2, p. 155-166, 2015.

CANUTO, A. Agronegócio: a modernização conservadora que gera exclusão pela produtividade. **Revista Nera**, n. 5, p. 1-12, 2012.

CARRANCO, J. S. et al. Comparative analysis of meteorological monitoring using an integrated low-cost environmental unit based on the Internet of Things (IoT) with an Automatic Meteorological Station (AWS). In: **2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)**. IEEE, 2017. p. 1-6.

CARVALHO JÚNIOR, L. C; GIAROLA, P. C. M. UM RETRATO DA CADEIA PRODUTIVA DE CARNE AVÍCOLA EM SANTA CATARINA E NO BRASIL NO INÍCIO DO SÉCULO XXI. **Revista Americana de Empreendedorismo e Inovação**, v. 2, n. 2, p. 141-150, 2020.

CARVALHO, P. N. A política agrícola comum da Europa: controvérsias e continuidade. **Texto para Discussão**, 2016.

CASTAÑEDA M. A; CASTAÑO M. V. M. Smart frost measurement for anti-disaster intelligent control in greenhouses via embedding IoT and hybrid AI methods. **Measurement**, v. 164, p. 108043, 2020.

CASTRO PAULA, H. V; CHELOTTI, M. C. O Processo de Modernização Tecnológica na Agricultura e a disputa territorial no Campo Brasileiro. **Espaço em revista**, v. 20, n. 1, 2018.

CHEN, S. et al. A vision of IoT: Applications, challenges, and opportunities with china perspective. **IEEE Internet of Things journal**, v. 1, n. 4, p. 349-359, 2014.

COMISSÃO EUROPEIA (2020). **A política agrícola comum**: A política agrícola comum apoia os agricultores e garante a segurança alimentar da Europa. Acessado em 16 de Dezembro, 2022 em "https://ec.europa.eu/info/food-farming-fisheries/key-policies/commonagricultural-policy/cap-glance_pt#legalfoundations"

CNA – CONFEDERAÇÃO NACIONAL DO AGRONEGÓCIO. PIB do Agronegócio brasileiro em 2020. Acessado: <https://www.cnabrasil.org.br/boletins/pib-do-agronegocio-alcanca-participacao-de-26-6-no-pib-brasileiro-em-2020> em 10 de abril de 2021.

CONTE, G. Y. C; ARAUJO, G. J. F. A internet da indústria das coisas: oportunidades e ganhos potenciais. **Ágora: revista de divulgação científica**, v. 25, p. 1-17, 2020.

CONWAY, G; GRIFFON, M; SABOURIN, E. **Uma agricultura sustentável para a segurança alimentar mundial**. EMBRAPA-CPATSA, 1998.

ÇORAK, B. H. et al. Comparative analysis of IoT communication protocols. In: **2018 International symposium on networks, computers and communications (ISNCC)**. IEEE, 2018. p. 1-6.

COSTANZO, S; FLORES, A. M. IoT Non-contact Body Temperature Measurement System Implementing Access Control for COVID-19. In: **Trends and Applications in Information Systems and Technologies: Volume 4 9**. Springer International Publishing, 2021. p. 255-264.

CRUZ, M A. A. et al. A proposal for bridging application layer protocols to HTTP on IoT solutions. **Future Generation Computer Systems**, v. 97, p. 145-152, 2019.

CRUZ, N. B. da et al. Family farming access to credit and to technical assistance in Brazil. **Revista de Economia e Sociologia Rural**, v. 59, n. 3, 2021.

DENARDIN, G. W; BARRIQUELLO, C. H. **Sistemas operacionais de tempo real e sua aplicação em sistemas embarcados**. Editora Blucher, 2019.

DHARMAYANTI, D.; BACHTIAR, A. M.; WIBAWA, A. P. Analysis of user interface and user experience on comrades application. In: **IOP Conference Series: Materials Science and Engineering**. IOP Publishing, 2018.

DIAS, F. M. MODERNIZAÇÃO DA AGRÍCULTURA E O AGRONEGÓCIO NO BRASIL: ALGUNS ELEMENTOS TEÓRICOS. **Revista de Geografia, Meio Ambiente e Ensino**, v. 11, n. 2, p. 13-24, 2020.

DINIZ, D. B; SILVA, R. F. **INTEROPERABILIDADE DE APLICAÇÕES COM SOAP (SIMPLE OBJECT ACCESS PROTOCOL)**. 2021.

DOBREAN, D; DIOSAN, L. Model View Controller in iOS mobile applications development. In: **SEKE**. 2019. p. 547-716.

DONKOH, A. Ambient temperature: a factor affecting performance and physiological response of broiler chickens. **International Journal of Biometeorology**, v. 33, n. 4, p. 259-265, 1989.

DUCKETT, J. **Javascript e Jquery, Desenvolvimento de Interfaces Web Interativas**. 1º Edição. Altabooks, 2016.

ELMASRI, R. ET AL. **Sistemas de Banco de Dados**. 2019.

EMBRAPA - Empresa Brasileira de Pesquisa Agropecuária. **Estatísticas de Frangos de corte no Brasil**. Disponível em: <https://www.embrapa.br/suinos-e-aves/cias/estatisticas/frangos/brasil> . Acessado em 02 de fevereiro de 2021.

FAO. **Cost and affordability of healthy diets across and within countries: Background paper for The State of Food Security and Nutrition in the World 2020**. FAO Agricultural Development Economics Technical Study No. 9. Food & Agriculture Org., 2020.

FAROOQ, M. O; WHEELOCK, I; PESCH, D. IoT-connect: An interoperability framework for smart home communication protocols. **IEEE Consumer Electronics Magazine**, v. 9, n. 1, p. 22-29, 2019.

FERRARI, A. B; GONÇALVES, B. S. Integração do framework de Garrett com as abordagens Lean UX e Ágil: exemplo aplicado ao desenvolvimento de um aplicativo. **e-Revista LOGO**, v. 6, p. 78-91, 2017.

FERRAZ, C. O; PINTO, W. F. Tecnologia da Informação para a agropecuária: utilização de ferramentas da tecnologia da informação no apoio a tomada de decisões em pequenas propriedades. **Revista Eletrônica Competências Digitais para Agricultura Familiar**, v. 3, n. 1, p. 38-49, 2017.

FERREIRA, C. P. HACKING E INFILTRAÇÃO POLICIAIS EM RESPOSTA AO USO DE CRIPTOGRAFIA POR ORGANIZAÇÕES CRIMINOSAS. **Revista Brasileira de Ciências Policiais**, v. 12, n. 5, p. 19-48, 2021.

FERREIRA, J. V.; SANTOS, V. A.; PORTELA, C. S. Avaliação da Experiência do Usuário e da Usabilidade de Aplicativos para Prática de Exercícios Físicos: Um Mapeamento Sistemático da Literatura. In: **Anais do XXI Simpósio Brasileiro de Computação Aplicada à Saúde**. SBC, 2021. p. 13-24.

FIELD, A. **Descobrimo a estatística usando o SPSS**. 2. ed. Porto Alegre: Bookman, 2009. p. 689

GALLO, B. Avicultura de alta performance no século 21. **ANAIS DO XVI SIMPÓSIO BRASIL SUL DE AVICULTURA E VII BRASIL SUL POULTRY FAIR**, p. 75, 2015.

GAMA, H. S et al. Especificação de Requisitos para Sistemas Embarcados: Uma Revisão Sistemática da Literatura. **Brazilian Journal of Development**, v. 7, n. 3, p. 25216-25226, 2021.

GARG, H; DAVE, M. Securing iot devices and securelyconnecting the dots using rest api and middleware. In: **2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)**. IEEE, 2019. p. 1-6.

GARRETT, J. J. **The elements of user experience: user-centered design for the web and beyond**. Pearson Education, 2010.

GEBBERS, Robin; ADAMCHUK, Viacheslav I. Precision agriculture and food security. **Science**, v. 327, n. 5967, p. 828-831, 2010.

GHADGE, A. et al. The impact of Industry 4.0 implementation on supply chains. **Journal of Manufacturing Technology Management**, 2020.

GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. 6. ed. São Paulo: Atlas, 2008.

GOKHALE, P; BHAT, O; BHAT, S. Introduction to IOT. **International Advanced Research Journal in Science, Engineering and Technology**, v. 5, n. 1, p. 41-44, 2018.

GRIMBLATT, V. IoT for Agribusiness: An overview. In: **2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS)**. IEEE, 2020. p. 1-4.

GUNDIM, Lígia et al. Causas de condenações de frangos de corte relacionadas a manejo e ambiência. **Enciclopédia Biosfera**, v. 11, n. 21, 2015.

HABERLI JR, C; OLIVEIRA, T; YANAZE, M. The adoption stages (Evaluation, Adoption, and Routinisation) of ERP systems with business analytics functionality in the context of farms. **Computers and electronics in agriculture**, v. 156, p. 334-348, 2019.

HAFEZ, H. M.; ATTIA, Y. A. Challenges to the poultry industry: current perspectives and strategic future after the COVID-19 outbreak. **Frontiers in veterinary science**, v. 7, p. 516, 2020.

HANSEN, C. W; JENSEN, P. S; SKOVSGAARD, C. V. Modern gender roles and agricultural history: the Neolithic inheritance. **Journal of Economic Growth**, v. 20, n. 4, p. 365-404, 2015.

HAIR, J. F. et al. **Análise multivariada de dados**. 5. ed. Porto Alegre: Bookman, 2009. 593 p.

HENN, J. D. et al. Emissão de dióxido de carbono na cama de primeiro lote de duas linhagens de frangos de corte. In: **Embrapa Suínos e Aves-Artigo em anais de congresso (ALICE)**. In: SEMINÁRIO INTERNACIONAL DE AVES E SUÍNOS, 12., 2013, Florianópolis. Anais... Florianópolis: Gessulli, 2013. AVESUI 2013., 2013.

HINDERKS, A et al. Applying Importance-Performance Analysis (IPA) to Interpret the Results of the User Experience Questionnaire (UEQ). **Journal of Web Engineering**, p. 243–266-243–266, 2020.

HODDIE, P; PRADER, L. IoT Development for ESP32 and ESP8266 with JavaScript: A Practical Guide to XS and the Moddable SDK. **Apress**, 2020.

HUSSAIN, J et al. Model-based adaptive user interface based on context and user experience evaluation. **Journal on Multimodal User Interfaces**, v. 12, n. 1, p. 1-16, 2018.

HUNING, T. R.; WAHL, F. The origins of agricultural inheritance traditions. **Journal of Comparative Economics**, v. 49, n. 3, p. 660-674, 2021.

INAL, Y. ET AL. Usability evaluations of mobile mental health technologies: systematic review. **Journal of medical Internet research**, v. 22, n. 1, p. e15337, 2020.

IPARDES - IparDES - Instituto Paranaense de Desenvolvimento Econômico e Social. Agronegócio e o PIB paranaense, 12p. 2019. Acesso: <http://www.ipardes.pr.gov.br/Noticia/Agronegocio-e-responsavel-por-339-do-PIB-paranaense> em 15 de março de 2021.

JINKUANG W., ET AL. Circuit and Electronic Foundation [M]. Guangzhou: Sun Yat-sen University Press, 2015.6:206-253.

JOHNSEN, F. T. et al. Application of IoT in military operations in a smart city. In: **2018 International Conference on Military Communications and Information Systems (ICMCIS)**. IEEE, 2018. p. 1-8.

JOHNSON, J. Designing with the mind in mind: simple guide to understanding user interface design guidelines. **Morgan Kaufmann**, 2020.

KAFLE, V. P.; FUKUSHIMA, Y; HARAI, H. Internet of things standardization in ITU and prospective networking technologies. **IEEE Communications Magazine**, v. 54, n. 9, p. 43-49, 2016.

KANAN, R; ELHASSAN, O; BENSALAM, R. An IoT-based autonomous system for workers' safety in construction sites with real-time alarming, monitoring, and positioning strategies. **Automation in Construction**, v. 88, p. 73-86, 2018.

KHALID, S. et al. A Novel Model for IoT-Based Meter Using ATmega328P Microcontroller and Google Cloud Store. In: **Advances in Electromechanical Technologies**. Springer, Singapore, 2021. p. 639-648.

KRČO, S; POKRIĆ, B; CARREZ, F. Designing IoT architecture (s): A European perspective. In: **2014 IEEE world forum on internet of things (WF-IoT)**. IEEE, 2014. p. 79-84.

KRISHNA, A. et al. IoT composer: composition and deployment of IoT applications. In: **2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)**. IEEE, 2019. p. 19-22.

KUHN, J. et al. Proposição de um Modelo Matemático para a Previsão Volumétrica de Silos Armazenadores de Ração. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 9, n. 1, 2022.

KUIPER, M; CUI, H. D. Using food loss reduction to reach food security and environmental objectives—A search for promising leverage points. **Food Policy**, p. 101915, 2020.

KUMAR, K; KUMAR, N; SHAH, R. Role of IoT to avoid spreading of COVID-19. **International Journal of Intelligent Networks**, v. 1, p. 32-35, 2020.

KUMARI, M; KUMAR, A; SINGHAL, R. Design and Analysis of IoT-Based Intelligent Robot for Real-Time Monitoring and Control. In: **2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)**. IEEE, 2020. p. 549-552.

LANGBECKER, T. B; BEURON, T. A. SUSTENTABILIDADE E AGRONEGÓCIO: ALINHAMENTOS AO PROCESSO DECISÓRIO EM UMA EMPRESA RURAL. **Revista GESTO: Revista de Gestão Estratégica de Organizações**, v. 8, n. 2, p. 71-89, 2020.

LI, H; LIU, X. Simulation of ship maneuvering training environment based on WebGL and WebSocket. In: **2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT)**. IEEE, 2020. p. 70-73.

LI, X. et al. Blockchain-empowered Data-driven Networks: A Survey and Outlook. **ACM Computing Surveys (CSUR)**, v. 54, n. 3, p. 1-38, 2021.

LIAO, D. et al. Design and evaluation of affective virtual reality system based on multimodal physiological signals and self-assessment manikin. **IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology**, v. 4, n. 3, p. 216-224, 2019.

LOPES, T; VALENTIM, N. UUDT-MA: Técnica Para Projeto da Usabilidade e Experiência do Usuário em Aplicações Móveis. In: **Anais Estendidos do XVIII Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais**. SBC, 2019. p. 146-149.

LUCKENBAUGH, A. **LibTracker--A React Native & Quarkus App to Track Your Personal Book Library**. 2020.

MABROUKI, J. et al. IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. **Big Data Mining and Analytics**, v. 4, n. 1, p. 25-32, 2021.

MACIEL, J. I. L. INDICADORES PARA O AGRONEGÓCIO UTILIZANDO FERRAMENTA WEB DE VISUALIZAÇÃO DA INFORMAÇÃO. **Dissertação (Mestrado em Ciência, Gestão e Tecnologia da Informação)** – Universidade Federal do Paraná. Curitiba, p. 239. 2019.

MAGRANI, E. **A internet das coisas**. Editora FGV, 2018.

MAIA, C. L. B; FURTADO, E. S. Uma revisão sistemática sobre medição da experiência do usuário. In: **Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems**. 2014. p. 369-372.

MAIA, C. L. B; FURTADO, E. S. Uma revisão sistemática sobre medição da experiência do usuário. In: **Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems**. 2014. p. 369-372.

MAISTRO, M. C. M.; MONTEBELLO, A. E. S; SANTOS, J. A. Desafios do agro empreendedorismo: as startups do campo. **Brazilian Journal of Development**, v. 5, n. 9, p. 14949-14964, 2019.

MAMAT, N. H. et al. Internet of Things in Flood Warning System: An Overview on the Hardware Implementation. In: **Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics**. Springer, Singapore, 2021. p. 269-279.

MARJANI, M. et al. Big IoT data analytics: architecture, opportunities, and open research challenges. **IEEE Access**, v. 5, p. 5247-5261, 2017.

MARQUES, G. Internet of Things Sensor Data Analysis for Enhanced Living Environments: A Literature Review and a Case Study Results on Air Quality Sensing. **Enabling AI Applications in Data Science**, p. 397-414, 2021.

MARQUES, L. C. et al. **UX-Tips: uma técnica de avaliação de user experience para aplicações de software**. 2019.

MEDEIROS, S. R. Pequeno glossário para a agropecuária 4.0. **Embrapa Pecuária Sudeste-Artigo de divulgação na mídia (INFOTECA-E)**, 2019.

MEHMOOD, N. Q; CULMONE, R.; MOSTARDA, L. Modeling temporal aspects of sensor data for MongoDB NoSQL database. **Journal of Big Data**, v. 4, n. 1, p. 1-35, 2017.

MEIER, A; KAUFMANN, M. **SQL & NoSQL databases**. Springer Fachmedien Wiesbaden, 2019.

MENDES, J. A. J; GERÓLAMO, M. C.; ZUIN, V. G. Modelo de maturidade para o agronegócio 4.0: uma revisão bibliográfica sistemática das dimensões de análise com base nos modelos aplicados a indústria 4.0. **Anais**, 2020.

MENDOZA, G. F. et al. Design of Learning Digital Tools Through a User Experience Design Methodology. In: **Advances in Computer, Communication and Computational Sciences**. Springer, Singapore, 2021. p. 755-764.

MISHRA, A. K.; EL-OSTA, H. S. Effect of agricultural policy on succession decisions of farm households. **Review of Economics of the Household**, v. 6, n. 3, p. 285-307, 2008.

MIT APP INVENTOR. Documentation. **Helpful explanations for all of the components and blocks in App Inventor**. Disponível em: < <https://appinventor.mit.edu/explore/library> >. Acessado em 05 de Junho de 2021.

MODUGU, S. R; FARHAT, H. Implementation of the Internet of Things Application Based on Spring Boot Microservices and REST Architecture. In: **Proceedings of the Computational Methods in Systems and Software**. Springer, Cham, 2020. p. 20-31.

MORAES, W. B. **Construindo aplicações com NodeJS-3a edição**. Novatec Editora, 2021.

MULAY, A; OCHIAI, H; ESAKI, H. IoT websocket connection management algorithm for early warning earthquake alert applications. In: **Companion Proceedings of the 10th International Conference on Utility and Cloud Computing**. 2017. p. 189-194.

NAIK, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In: **2017 IEEE international systems engineering symposium (ISSE)**. IEEE, 2017. p. 1-7.

NAIME, J. de M.; CAMARGO NETO, J.; VAZ, Carlos Manoel Pedro. Avaliação geral, resultados, perspectivas e uso de ferramentas de agricultura de precisão. **Embrapa Instrumentação-Capítulo em livro científico (ALICE)**, 2011.

NALENDRA, A. K. et al. Monitoring System IoT-Broiler Chicken Cage Effectiveness of Seeing Reactions from Chickens. In: **Journal of Physics: Conference Series**. IOP Publishing, 2021.

NAZARENO, A. C. ET AL. Bem-estar na produção de frango de corte em diferentes sistemas de criação. **Revista Engenharia Agrícola**, Jaboticabal , v. 31, n. 1, p. 13-22, fev. 2011.

NGU, A. H. et al. IoT middleware: A survey on issues and enabling technologies. **IEEE Internet of Things Journal**, v. 4, n. 1, p. 1-20, 2016.

Node-Red. Documentação Oficial: **Guia do usuário**. Disponível em: < <https://nodered.org/docs/user-guide/> > Acesso em 10 de Mai. 2021.

NORMAN, D. A.; NIELSEN, J. Gestural interfaces: a step backward in usability. **Interactions**, v. 17, n. 5, p. 46-49, 2010.

NOVICK, V. **React Native-Building Mobile Apps with JavaScript**. Packt Publishing Ltd, 2017.

OJO, R. O. et al. Internet of Things and Machine Learning techniques in poultry health and welfare management: A systematic literature review. **Computers and Electronics in Agriculture**, v. 200, p. 107266, 2022.

OLIVEIRA, F. L; NÓBREGA, L. Aprendizagem em Dispositivos móveis: o uso do Whatsapp como Plataforma no M-learning. **REVISTA DE TECNOLOGIA APLICADA**, v. 10, n. 1, p. 50-59, 2021.

OLIVEIRA, G.; MAGALHÃES, A.; FIALHO, S. Indicadores em educação a distância: o uso da matriz de necessidades para ambientes virtuais de aprendizagem. **In: XVII Congresso Internacional de Informática Educativa, TISE**. Santiago, Chile. 2012. p. 409-412.

OLIVEIRA, L; TONIN, P. C; VICENZI, S. L. Comportamento dos custos totais de produção no segmento da avicultura de postura no estado do Paraná: estudo baseado na análise de regressão linear múltipla. **Revista Produção Online**, v. 20, n. 1, p. 28-46, 2020.

OLIVEIRA, L. F. P; MOREIRA, A. P.; SILVA, M. F. Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. **Robotics**, v. 10, n. 2, p. 52, 2021.

OLIVEIRA, S. **Internet das coisas com ESP8266, Arduino e Raspberry Pi**. 2 ED. Novatec Editora, 2021.

OLIVEIRA, V. L. C. et al. Estudo dos Agronegócios 4.0–Tecnologias, desafios e benefícios nos Agronegócios. **Research, Society and Development**, v. 11, n. 13, p. e363111335379-e363111335379, 2022.

PATEL, A; DEVAKI, P. Survey on NodeMCU and Raspberry pi: IoT. **International Research Journal of Engineering and Technology (IRJET)**, v. 6, n. 4, p. 5101-5105, 2019.

PAUFERRO, G. B. A; PAIVA, S. V. F; LESSA, N. M. IoT: conceitos de segurança de dados e criptografia. **Cogitare**, v. 2, n. 3, p. 40-52, 2020.

PAUL, A; NALWAYA, A. **React Native for Mobile Development**. California: Apress, Berkeley, CA. <https://doi.org/10.1007/978-1-4842-4454-8>, 2019.

PAWAR, P. et al. Hitching Medical IoT Devices to Blockchain for Personal Health Information Management. **Blockchain Technology for IoT Applications**, p. 191, 2021.

PERKINS, L; REDMOND, E; WILSON, J. **Seven databases in seven weeks: a guide to modern databases and the NoSQL movement**. Pragmatic Bookshelf, 2018.

PHASINAM, K. et al. Application of IoT and cloud computing in automation of agriculture irrigation. **Journal of Food Quality**, 2022.

PINHEIRO, J. S. et al. **Análise do desenvolvimento de aplicativos mobile nativos e multiplataforma**. 2020.

PROCÓPIO, D. P. et al. Avaliação conjuntural da avicultura no Brasil. **Research, Society and Development**, v. 9, n. 3, p. e47932312-e47932312, 2020.

QUEIRÓS, R; PORTELA, F. **Desenvolvimento Avançado para a Web: Do Front-end Ao Back-end**. São Paulo: FCA, 2020.

RAEMY, M. **RESTful API mit Java Spring**. 2020.

RAFIQUE, W. et al. An application development framework for Internet-of-Things service orchestration. **IEEE Internet of Things Journal**, v. 7, n. 5, p. 4543-4556, 2020.

RAHMAN, R. A; SHAH, B. Security analysis of IoT protocols: A focus in CoAP. **In: 2016 3rd MEC international conference on big data and smart city (ICBDSC)**. IEEE, 2016. p. 1-7.

RAJASEKAR, P. et al. CLOUD BASED FARM MONITORING USING IOT. **International Journal of Modern Agriculture**, v. 10, n. 1, p. 469-476, 2021.

RAJESWARI, S.; SUTHENDRAN, K.; RAJAKUMAR, K. A smart agricultural model by integrating IoT, mobile and cloud-based big data analytics. **In: 2017 international conference on intelligent computing and control (I2C2)**. IEEE, 2017. p. 1-5.

RAMOS, G. The continuity of family agriculture and the succession system: The Basque case. **Journal of Comparative Family Studies**, v. 36, n. 3, p. 367-375, 2005.

RANI, D; KUMAR, N. A Review of IoT Techniques and Devices: Smart Agriculture Perspective. **Proceedings of ICRIC 2019**, p. 113-123, 2020.

RAPELA, M. A. **Fostering Innovation for Agriculture 4.0**. Springer International Publishing, 2019.

REGADO, A; PINTO, A. S. Avaliação de lojas online segundo as 7 variáveis do UX. **Research Bulletin (Cadernos de Investigação) of the Master in E-Business**, v. 1, n. 1, 2021.

REHMAN, A. et al. A revisit of internet of things technologies for monitoring and control strategies in smart agriculture. **Agronomy**, v. 12, n. 1, p. 127, 2022.

RICHARDSON, B; CAMPBELL-YEO, M; SMIT, M. Mobile application user experience checklist: A tool to assess attention to core UX principles. **International Journal of Human-Computer Interaction**, p. 1-8, 2021.

RIVERO, L; CONTE, T. A systematic mapping study on research contributions on UX evaluation technologies. In: **Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems**. 2017. p. 1-10.

ROCHA, G. S. R; OLIVEIRA, L; TALAMINI, E. Blockchain Applications in Agribusiness: A Systematic Review. **Future Internet**, v. 13, n. 4, p. 95, 2021.

RODRIGUEZ, R. Y; JULCAPOMA, M. R. Implementation of a sensor node for monitoring physiological signals with Websocket communication and data visualization in a node.js server over the Internet. In: **2020 IEEE Engineering International Research Conference (EIRCON)**. IEEE, 2020. p. 1-4.

ROWE, E; DAWKINS, M. S; GEBHARDT, S. G. H. A systematic review of precision livestock farming in the poultry sector: Is technology focussed on improving bird welfare?. **Animals**, v. 9, n. 9, p. 614, 2019.

RUBÍ, J. N. S; GONDIM, P. R. L. IoT-based platform for environment data sharing in smart cities. **International Journal of Communication Systems**, v. 34, n. 2, p. e4515, 2021.

SALAM, A. Internet of things in agricultural innovation and security. In: **Internet of Things for Sustainable Community Development**. Springer, Cham, 2020. p. 71-112.

SAMPIERI, R. H. et al. **Metodologia de pesquisa**. 5. ed. Porto Alegre: Penso, 2006. 624 p.

SANTOS FILHO, J. I.; TALAMINI, D. JD; MARTINS, F. M. Conjuntura econômica da avicultura brasileira. **Embrapa Suínos e Aves-Artigo em periódico indexado (ALICE)**, 2019.

SANTOS, J. O. et al. A evolução da agricultura orgânica. **Revista brasileira de gestão ambiental**, v. 6, n. 1, p. 35-41, 2012.

SANTOS, T. C.; ESPERIDIÃO, T. L; AMARANTE, M. S. AGRICULTURA 4.0. **Revista Pesquisa E Ação**, v. 5, n. 4, p. 122-131, 2019.

SCHWEAN-LARDNER, K; CLASSEN, H. **Iluminación para pollo de engorde**, Estados Unidos, 2010.

SEBESTA, R. W. **Conceitos de Linguagens de Programação-11**. Bookman Editora, 2018.

SIDNA, J. et al. Analysis and evaluation of communication Protocols for IoT Applications. In: **Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications**. 2020. p. 1-6.

SILBERSCHATZ, A; SUNDARSHAN, S.; KORTH, H. F. **Sistema de banco de dados**. Ed. 7, Elsevier Brasil, 2020.

SILVA, D. R. ET. AL. Latency evaluation for MQTT and WebSocket Protocols: an Industry 4.0 perspective. In: **2018 IEEE Symposium on Computers and Communications (ISCC)**. IEEE, 2018. p. 01233-01238.

SILVA, M. S. **Web Design Responsivo: aprenda a criar sites que se adaptam automaticamente a qualquer dispositivo, desde desktops até telefones celulares**. Novatec Editora, 2018.

SILVA, V. R. et al. Análise dos impactos da logística 4.0 em uma empresa do ramo agrícola da cidade de Cambé, Estado do Paraná, Brasil. **Research, Society and Development**, v. 9, n. 8, p.68-83, 2020.

SMITH, B. **JSON Básico: Conheça o formato de dados preferido da web**. 2.ED. São Paulo: Novatec, 2020.

SONI, A; RANGA, V. API features individualizing of web services: REST and SOAP. **International Journal of Innovative Technology and Exploring Engineering (IJITEE)** ISSN, p. 2278-3075, 2019.

SRINIVASAN, C. R. et al. A review on the different types of Internet of Things (IoT). **Journal of Advanced Research in Dynamical and Control Systems**, v. 11, n. 1, p. 154-158, 2019.

SRUTHI, M.; KAVITHA, B. R. A survey on iot platform. **International Journal of Scientific Research and Modern Education**, ISSN (online), p. 2455-5630, 2016.

SUESS-REYES, J; FUETSCH, E. The future of family farming: A literature review on innovative, sustainable and succession-oriented strategies. **Journal of rural studies**, v. 47, p. 117-140, 2016.

SUNG, W. T; HSIAO, S. J.; SHIH, J. A.. Construction of indoor thermal comfort environmental monitoring system based on the IoT architecture. **Journal of Sensors**, v. 2019, 2019.

TAN, P. et al. Teaching management system with applications of RFID and IoT technology. **Education Sciences**, v. 8, n. 1, p. 26, 2018.

THAKUR, R. N; PANDEY, U. S. The role of model-view controller in object oriented software development. **Nepal Journal of Multidisciplinary Research**, v. 2, n. 2, p. 1-6, 2019.

THOMAZINI, D; ALBUQUERQUE, P. U. B. **Sensores industriais: fundamentos e aplicações**. Saraiva Educação SA, 2020.

TIGHTIZ, L.; YANG, H. A comprehensive review on IoT protocols' features in smart grid communication. **Energies**, v. 13, n. 11, p. 2762, 2020.

TIWARY, G. P.; STROULIA, E; SRIVASTAVA, A. Compression of XML and JSON API Responses. **IEEE Access**, v. 9, p. 57426-57439, 2021.

TJB, D. D; SUBRAMANI, A.; SOLANKI, V. K. Smart City: IOT Based Prototype for Parking Monitoring and Management System Commanded by Mobile App. *Annals of Computer Science and Information Systems*, v. 10, p. 341-343, 2017.

TORKAMAN, A; SEYYEDI, M. A. Analyzing IoT reference architecture models. **International Journal of Computer Science and Software Engineering**, v. 5, n. 8, p. 154, 2016.

TOSCHI, G. M.; CAMPOS, L. B.; CUGNASCA, C. E. Home automation networks: A survey. **Computer Standards & Interfaces**, v. 50, p. 42-54, 2017. ISSN 0920-5489.

TOUQEER, H et al. Smart home security: challenges, issues and solutions at different IoT layers. **The Journal of Supercomputing**, p. 1-37, 2021.

TRIANAFYLLOU, A; SARIGIANNIDIS, P; BIBI, S. Precision agriculture: A remote sensing monitoring system architecture. **Information**, v. 10, n. 11, p. 348, 2019.

TU, M; LIM, M. K.; YANG, M. F. IoT-based production logistics and supply chain system–Part 1. **Industrial Management & Data Systems**, 2018.

TUN, Z. et al. Mongodb on Cloud for Weather Data (Temperature and Humidity) in Sittway. In: **2020 IEEE Conference on Computer Applications (ICCA)**. IEEE, 2020. p. 1-6.

ULFADA, E; NURFIANA, N; HANDAYANI, R. D. Perancangan Desain Ui/Ux Pada Implementasi Sistem Kontrol Smart Farming Berbasis Internet Of Things (IoT). In: **Prosiding Seminar Nasional Darmajaya**. 2022. p. 145-155.

ULLAH, R. et al. EEWMP: An IoT-Based Energy-Efficient Water Management Platform for Smart Irrigation. **Scientific Programming**, v. 2021, 2021.

USDA – United States Department Agriculture. **Livestock and Poultry: World Markets and Trade in 2021**. Disponível em: https://apps.fas.usda.gov/psdonline/circulars/livestock_poultry.pdf . Acessado em: 03 de junho de 2021.

VALLURU, D; KOTIKAM, G; HARIBABU, K. Smart Agriculture Management of Intelligent Things Using NB-IoT. **International Journal of Modern Agriculture**, v. 10, n. 1, p. 78-86, 2021.

VANNIER, L. R et al. Carne de Frango-Um estudo sobre o consumo, atitudes e o comportamento do consumidor brasileiro. **The Journal of Engineering and Exact Sciences**, v. 8, n. 9, p. 14854-01e, 2022.

VERDOUW, C. et al. **Digital twins in smart farming**. *Agricultural Systems*, v. 189, p. 046-103, 2021.

VERMULM, R. **Políticas para o Desenvolvimento da Indústria 4.0 no Brasil**. Instituto de Estudos para o Desenvolvimento Industrial (IED). São Paulo, 2018.

VIEIRA, F.M.C. et al. Influência das condições térmicas do galpão de espera climatizado na mortalidade pré-abate de frangos de corte. **Arquivo Brasileiro de Medicina Veterinária e Zootecnia**, Belo Horizonte, v. 68, n. 2, p. 475-482, abr. 2016.

VOGADO, G. M. S. et al. Evolução da avicultura brasileira. **Nucleus Animalium**, v. 8, n. 1, p. 49-58, 2016.

VOUTOS, Y. et al. An IoT-based insular monitoring architecture for smart viticulture. In: **2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)**. IEEE, 2019. p. 1-4.

WU, C.J; YOSHIOKA, K; MATSUMOTO, T. ThingGate: A Gateway for Managing Traffic of Bare-metal IoT Honeypot. **Journal of Information Processing**, v. 28, p. 481-492, 2020.

YADAV, V. S. et al. Exploring the application of Industry 4.0 technologies in the agricultural food supply chain: a systematic literature review. **Computers & Industrial Engineering**, p. 304, 2022.

ZANIN, A. et al. Robotização na ordenha leiteira e o conceito do Agronegócio 4.0. In: **Anais do Congresso Brasileiro de Custos-ABC**. 2022.

ZEYMER, J. S; SILVA, C. H. O; SCHNEIDER, M. G. Comparação entre a exatidão de sensores analógicos e digitais utilizados em sistemas de termometria para armazenagem de grãos. **Brazilian Journal of Animal and Environmental Research**, v. 4, n. 1, p. 813-819, 2021.

ZHAI, Z. et al. Decision support systems for agriculture 4.0: Survey and challenges. **Computers and Electronics in Agriculture**, v. 170, p. 105256, 2020.

Zhengzhou Winsen Electronics Technology Co. Ltda. **SENSOR, Flammable Gas. User Manual**, version, v. 1, 2020.

ZOCCA, R. et al. Decision-making computationally aided in the management of energy sources used in agrifood industries. **Energy Procedia**, v. 161, p. 100-107, 2019.

APÊNDICE 1 – GUIA DE ENTREVISTA:

Seção 1: Identificação do Respondente:

Cargo: _____

Tempo de Empresa: _____

Formação: _____

Tipo da Empresa: () Origem Familiar () Sociedade Anônima () Cooperativa

Outra: _____

Localização: _____

Fornecer para: () Cooperativas () Agroindústria

Outros: _____

Número de Funcionários: _____

Classificação da granja: () Pequena () Média () Grande

(Classificação conforme resolução número 4.174 de 2012 do Ministério da Agricultura, Pecuária e Abastecimento)

Seção 2 – Relação do Produtor com a agroindústria

- Qual é a relação comercial da granja com os frigoríficos (agroindústrias)?
- São fornecidos insumos produtivos e orientações técnicas especializadas?
- A remuneração está diretamente relacionada com a produtividade? São estabelecidas metas de produtividade por parte das agroindústrias?
- Demais considerações

Seção 3 – Questões práticas de produção e manejo

- Como é o processo de entrada e saída dos frangos?
- Quantas etapas os frangos passam dentro da granja? Qual o período de tempo de cada etapa?
- Qual a etapa mais crítica (importante) para o processo?
- Demais considerações

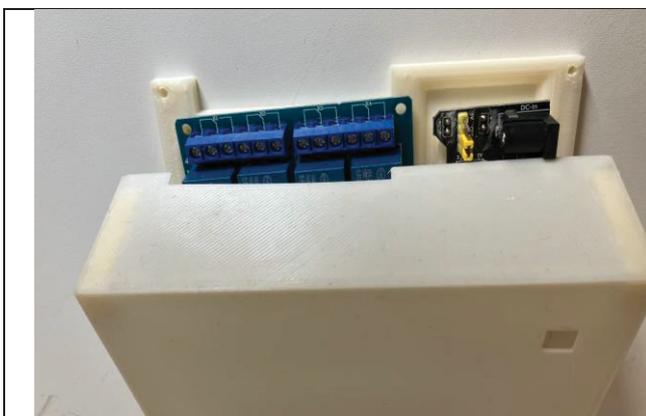
Seção 4 – Questões tecnológicas: problemas, demandas e prioridades

- A sua granja possui algum sistema de gestão? Ou conhece alguma granja que possua alguma automação?
- Conhece os sistemas de internet das coisas para a avicultura? Qual motivo de não ter interesse em investir em um sistema automatizado na granja?
- Qual é a maior dificuldade de controlar no processo produtivo?
- O que ajudaria na gestão? O que é prioridade?
- As soluções existentes hoje atendem/atenderiam suas necessidades?
- Demais considerações

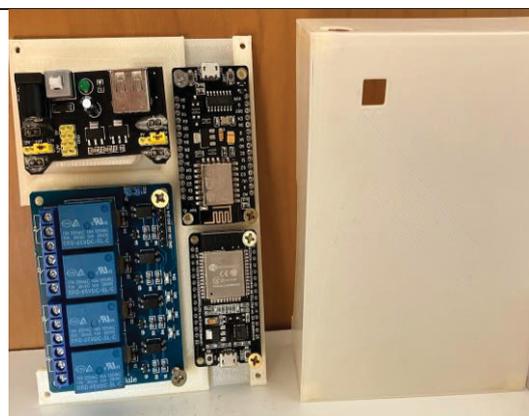
APÊNDICE 2 – CUSTO DOS COMPONENTES UTILIZADOS

Nome	Ilustração	Função/Objetivo	Quantidade utilizada para 1 data logger	Preço unitário estimado em dólar
NodeMCU V3 Lolin		Conectar e controlar os sensores e atuadores. Também tem a função de conectar-se à rede e se comunicar (via requisições) com o <i>client-side</i> .	1un	US\$ 4
Sensor BMP280		Sensor utilizado para aferir temperatura (°C), umidade(%) e pressão barométrica.	1un	US\$ 9
Sensor MQ-2		Capturar a presença de gases inflamáveis e fumaça.	1un	US\$ 3
Sensor MQ-135		Mensurar a quantidade (ppm) de gases tóxicos - amônia, dióxido de carbono, benzeno, óxido nítrico e também fumaça e álcool	1 un	US\$3,50
Sensor MQ-137		Medir a quantidade de NH3 (gás amônia) no ambiente em PPM	1 un	US\$ 45
Sensor Bh1750		Sensor que mede o quantitativo de luminosidade (lux) no ambiente.	1un	US\$ 3
Cabos jumper		Utilizado como condutor para conectar o microcontrolador com os sensores e demais componentes presentes no circuito internamente no data logger	6 cabos	US\$ 0,10
Cabo duas vias		Utilizado para conectar o data logger com a parte externa, como a conexão elétrica e os sensores que ficam no ambiente.	20cm	US\$ 1
Conector de fio duplo tipo presilha		Emenda de fios que funcionam como encaixe sem a necessidade de solda. Utilizada para aumentar o tamanho da conexão entre os componentes e permitir o plug-and-play do registrador de dados.	10 unidades	US\$ 2
modulo relé 4 canais		Utilizado para facilitar o acionamento de cargas através de um microcontrolador utilizando o software desenvolvido na pesquisa.	1un	US\$ 4
TOTAL			20 itens	US\$ 75,10

APÊNDICE 3 – FOTOS DOS DATA LOGGER



(A)



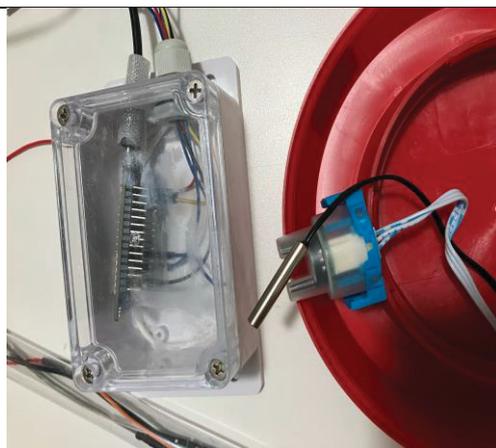
(B)



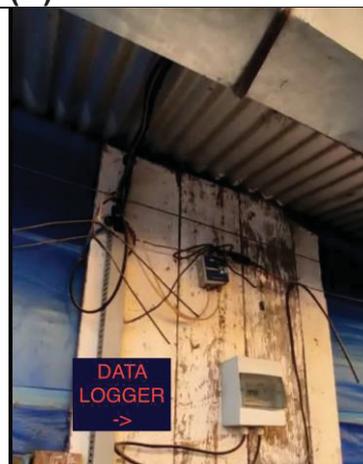
(C)



(D)



(E)



(F)

(A)(B)(C) – Projeto da case contendo o microcontrolador, modulo relé e fonte de alimentação

(D) - sensores diversos utilizados, case com fonte e microcontrolador

(E) - Data logger com conexões dos sensores para monitoramento da água dos frangos (sensor de turbidez e temperatura).

(F) Instalação do data logger dentro de uma caixa de disjuntor (para proteger dos fenômenos ambientais) e conexões

APÊNDICE 4 – FORMULÁRIO DE AVALIAÇÃO DA ARQUITETURA IOT

Avaliação da aplicação IoT

Uma das etapas de minha tese é avaliar a proposta de uma aplicação IoT, cujo link para acesso está no corpo do e-mail. Dentre os objetivos está "Avaliar a proposta com produtores e outros atores da cadeia produtiva de frango de corte". Desde já agradeço sua participação.
O tempo de preenchimento é menor que 5 minutos.

 souza.lucasj@gmail.com (não compartilhado) [Alternar conta](#) 

*Obrigatório

1 - Eu sou: *

Produtor / Granjeiro

Agroindústria

Cooperativa

Outro: _____

2 - Meu nível de escolaridade: *

Ensino Fundamental

Ensino Médio

Graduação

Pós Graduação

Outro: _____

Assinale a opção de acordo com seu grau de concordância. Considerando uma escala de 0 a 100%

3 - A aplicação apresenta um design de interface agradável e bonito *

1 2 3 4 5 6 7 8 9 10

Interface
confusa e feia
(0%)

Interface agradável
e bonita (100%)

4 - O esquema de cores e contrastes apresentado é adequado *

1 2 3 4 5 6 7 8 9 10

Paleta de cores
não agradável (0%)

Paleta de cores
agradável (100%)

5 - A aplicação desperta o interesse de obtê-la *

1 2 3 4 5 6 7 8 9 10

Não desperta
interesse (0%)

Desperta
Interesse (100%)

6 - É agradável navegar pela aplicação. *

1 2 3 4 5 6 7 8 9 10

Desagradável (0%)

Agradável (100%)

7 - O conteúdo da aplicação (textos, imagens, informações, ícones) são apresentados de forma visível e compreensível. *

1 2 3 4 5 6 7 8 9 10

Não são visíveis e/ou compreensíveis (0%) Conteúdos visíveis e compreensíveis (100%)

8 - A aplicação é fácil suficiente para realizar as atividades sem dificuldades. *

1 2 3 4 5 6 7 8 9 10

É difícil realizar as operações (0%) Realizo as operações de forma fácil (100%)

9 - A aplicação não exige muito esforço mental para lembrar como usá-la. *

1 2 3 4 5 6 7 8 9 10

Preciso ficar lembrando como usar a aplicação (0%) A aplicação é amigável e intuitiva (100%)

10 - A aplicação auxilia em alguma(s) atividade(s) importante(s) *

1 2 3 4 5 6 7 8 9 10

Não resolve nenhum problema (0%) A aplicação é útil para minha gestão (100%)

11 - Quais as 3 principais dificuldades em uma granja? *

Sua resposta

12 - Quais funcionalidades não são uteis? E quais funcionalidades que não existem no aplicativo poderiam ser uteis?

Sua resposta

EXPERIÊNCIA GERAL

13 - Nota geral da experiência de uso da aplicação

1 2 3 4 5 6 7 8 9 10

Críticas, dúvidas, sugestões e demais considerações sobre a experiência de uso da aplicação

Sua resposta

OBRIGADO POR PARTICIPAR DA PESQUISA.

Caso queira (opcional), preencha seu e-mail para receber um feedback da pesquisa e receber atualizações sobre o desenvolvimento da aplicação.

APÊNDICE 5 – SCRIPT ANÁLISES AVALIAÇÃO

```
#script avaliação da proposta IoT
#bibliotecas
library(readxl)
library(FSA)
library(ggplot2)
library(tidyverse)

#carregando os dados
dados <- read_excel("E:/TESE/dados/avaliacao_respostas.xlsx",
                    sheet = "dados_quantit")
View(dados)

#excluir var não utilizadas
dados$obs = NULL
dados$data = NULL

#####
#protocolo de avaliação
#####

#etapa 1 - descritivas
tabela = table(dados$nivelEscolaridade, dados$cargo)
prop.table(tabela, 1)*100
round(prop.table(tabela, 2)*100, 1)

summary(dados)

#graficos
#grafico de barras - avaliação por cargo
lado = ggplot(dados, aes(x= notaGeral, fill = cargo))+
  scale_y_continuous(limits = c(0,6),
                    breaks = seq(0,6))+
  xlab("Nota geral para a plataforma") +
  ylab("Número de respondentes")

lado = lado+geom_bar(position = "dodge")
lado

#boxplot cargo x nota
boxplot_geral = ggplot(dados, aes(y = notaGeral, x = cargo, fill = cargo)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(5,10), breaks = seq(5,10)) +
  theme_classic(base_size = 18) +
  xlab("Cargo do respondente") +
  ylab("Nota geral atribuida")

boxplot_geral

#ACM
#tratamento das variáveis para a ACM

#1 - excluindo colunas n utilizadas

dados$obs = NULL
dados$data = NULL
dados$design...5 = NULL
dados$cores...6 = NULL
dados$geralInteresse = NULL
dados$navegacao...8 = NULL
dados$conteudoCompreensivel = NULL
dados$facilUsar = NULL
dados$intuitivo...11 = NULL
dados$importante = NULL
dados$notaGeral = NULL

#transformar os dados em matriz

fatores = c("cargo", "nivelEscolaridade", "notaGeral", "design",
           "cores", "interesse", "navegacao", "conteudoCompreensivel",
           "facilUsar", "intuitivo", "importante")

dados[, fatores] = lapply(dados[, fatores], as.factor)
```

```
#rodar a acm
resultados = MCA (dados, ncp = 99)
autovalores = get_eigenvalue(resultados)

#mostrar graficamente as rela es entre os grupos
fviz_mca_biplot(resultados, repel = TRUE)
fviz_mca_var(resultados, choice = "mca.cor", col.var = "black")

#ETAPA 5: DISTRIBUI O GR  FICA DAS CATEGORIAS
fviz_mca_var(resultados, repel = TRUE, col.var = "black")

#graficos com menos variaveis
#removendo vari veis
dados$design = NULL
dados$cores = NULL
dados$interesse = NULL
dados$navegacao = NULL
dados$conteudoCompreensivel = NULL
dados$facilUsar = NULL
dados$intuitivo = NULL
dados$importante = NULL

#transformar a base de dados em matrizes (fatores)
fatores = c("cargo", "nivelEscolaridade", "notaGeral")
dados[, fatores] = lapply(dados[, fatores], as.factor)

#3  PASSO: rodar a analise de correspondencia multipla
resultados = MCA (dados, ncp = 99)
autovalores = get_eigenvalue(resultados)
#identificar o numero m ximo das dimens es]
autovalores

#mostrar graficamente as rela es entre os grupos
fviz_mca_biplot(resultados, repel = TRUE)
fviz_mca_var(resultados, choice = "mca.cor", col.var = "black")

#ETAPA 5: DISTRIBUI O GR  FICA DAS CATEGORIAS
fviz_mca_var(resultados, repel = TRUE, col.var = "black")
```


APÊNDICE 7 – BASES DE DADOS DA VALIDAÇÃO

Por conta do tamanho das bases de dados, o link para acesso na íntegra é: <https://github.com/lucas231090/Agro-4.0-UFPR-Doutorado-Lucas>

Neste apêndice estão somente as 40 primeiras linhas de cada base de dados para exemplificar.

data	cod_area	id_loca	semana	total_ares	incultivado	refugio	viabilidade	peso_med	grud_medio	conv_alimentar	lep	pag_aue	temp_med	umid_med	hume_t	hume_u	hume_p	dht_t	dht_u	mg-2	mg-137
jan/22	CONTROLE 01	GR 01	1	3600	118	10	98,69	170	2,6	0,76	315,98	R\$ 1,29	35,5	46,7							
jan/22	CONTROLE 01	GR 01	2	9672	65	8	97,95	449	32	1,03	304,99	R\$ 1,29	29,8	57,8							
jan/22	CONTROLE 01	GR 01	3	9599	48	22	97,23	885	42	1,22	335,88	R\$ 1,29	28,5	61,1							
fev/22	CONTROLE 01	GR 01	4	9529	32	12	96,79	1478	51	1,33	384,13	R\$ 1,29	26,9	65,5							
fev/22	CONTROLE 01	GR 01	5	9485	29	18	96,31	2155	63	1,50	395,31	R\$ 1,29	25,8	63,8							
fev/22	CONTROLE 01	GR 01	6	9438	37	12	95,81	2789	65	1,61	395,15	R\$ 1,29	23,8	73,2							
fev/22	CONTROLE 01	GR 01	7	9389	49	13	95,17	3001	68	1,76	360,63	R\$ 1,29	22,8	85,9							
mar/22	CONTROLE 01	GR 01	1	9327	378	95	95,17	3001	68	1,76	360,63	R\$ 1,29	27	65							
mar/22	CONTROLE 01	GR 02	1	11200	211	16	97,97	169	23	0,66	358,39	R\$ 1,31	35	44							
mar/22	CONTROLE 01	GR 02	2	10973	54	13	97,38	402	35	1,13	247,44	R\$ 1,31	32	59							
mar/22	CONTROLE 01	GR 02	3	10995	43	16	96,85	799	39	1,25	294,79	R\$ 1,31	29	68							
abr/22	CONTROLE 01	GR 02	4	10847	39	18	96,34	1521	50	1,23	425,47	R\$ 1,31	28	63							
abr/22	CONTROLE 01	GR 02	5	10790	45	16	95,79	2097	63	1,48	387,80	R\$ 1,31	27	68							
abr/22	CONTROLE 01	GR 02	6	10729	39	10	95,36	2901	76	1,55	424,93	R\$ 1,31	22	82							
mai/22	CONTROLE 01	GR 02	7	10680	61	8	94,74	3030	72	1,73	368,74	R\$ 1,31	21	83							
mai/22	CONTROLE 01	GR 02	7	10611	492	97	94,74	3030	72	1,73	368,74	R\$ 1,31	28	67							
mai/22	CONTROLE 01	GR 03	1	9900	177	19	98,02	195	23	0,63	453,42	R\$ 1,30	31	48							
jun/22	CONTROLE 01	GR 03	2	9704	33	8	97,61	590	32	1,16	354,60	R\$ 1,30	30	56							
jun/22	CONTROLE 01	GR 03	3	9663	47	11	97,02	763	39	1,33	285,04	R\$ 1,30	29	62							
jun/22	CONTROLE 01	GR 03	4	9605	41	20	96,40	1377	51	1,38	343,55	R\$ 1,30	27	65							
jun/22	CONTROLE 01	GR 03	5	9544	48	20	95,72	2149	60	1,50	391,80	R\$ 1,30	24	74							
ago/22	CONTROLE 01	GR 03	6	9476	47	9	95,15	2801	63	1,59	389,10	R\$ 1,30	22	72							
ago/22	CONTROLE 01	GR 03	7	9420	52	17	94,44	3090	77	1,77	366,59	R\$ 1,30	26	88							
fin de lote	CONTROLE 01	GR 03	1	9350	446	104	94,44	3090	77	1,77	366,59	R\$ 1,30	26	88							
set/22	CONTROLE 01	GR 04	1	9000	207	18	97,50	185	23	0,59	456,74	R\$ 1,27	31	45							
set/22	CONTROLE 01	GR 04	2	8775	69	7	96,66	565	36	1,09	357,87	R\$ 1,27	31	57							
set/22	CONTROLE 01	GR 04	3	8699	54	7	95,98	712	42	1,29	252,26	R\$ 1,27	29	62							
set/22	CONTROLE 01	GR 04	4	8638	39	12	95,41	1434	46	1,49	327,95	R\$ 1,27	26	63							
set/22	CONTROLE 01	GR 04	5	8587	47	14	94,73	2123	60	1,59	361,40	R\$ 1,27	25	70							
out/22	CONTROLE 01	GR 04	6	8526	58	16	93,91	2852	69	1,63	391,23	R\$ 1,27	23	82							
out/22	CONTROLE 01	GR 04	7	8452	40	9	93,37	3070	71	1,82	349,98	R\$ 1,27	22	91							
fin de lote	CONTROLE 01	GR 04	7	8403	514	83	93,37	3070	71	1,82	349,98	R\$ 1,27	27	67							
nov/22	CONTROLE 01	LUCAS	1	9500	98	13	98,83	198	33	0,57	480,44	R\$ 1,53	31	47							
nov/22	CONTROLE 01	LUCAS	2	9389	31	18	98,32	601	42	0,98	430,67	R\$ 1,53	31	57							
nov/22	CONTROLE 01	LUCAS	3	9340	38	22	97,76	890	48	1,11	373,57	R\$ 1,53	29	62							
dez/22	CONTROLE 01	LUCAS	4	9290	23	21	97,33	1602	56	1,28	457,73	R\$ 1,53	26	63							
dez/22	CONTROLE 01	LUCAS	5	9246	21	18	96,92	2246	67	1,36	453,26	R\$ 1,53	26	74							
dez/22	CONTROLE 01	LUCAS	6	9207	29	18	96,42	2850	72	1,51	453,30	R\$ 1,53	24	82							
dez/22	CONTROLE 01	LUCAS	7	9160	21	17	96,02	3290	79	1,63	430,69	R\$ 1,53	22	88							
fin de lote	CONTROLE 01	LUCAS	7	9122	251	127	96,02	3290	79	1,63	430,69	R\$ 1,53	27	68							

coleta	cod_sensor	temperatura	umidade	CO2
1	sensor_LCS	33	93	441
2	sensor_LCS	24	73	541
3	sensor_LCS	23	70	465
4	sensor_LCS	26	38	438
5	sensor_LCS	26	50	462
6	sensor_LCS	24	92	427
7	sensor_LCS	33	48	568
8	sensor_LCS	30	52	585
9	sensor_LCS	29	68	531
10	sensor_LCS	33	87	502
11	sensor_LCS	26	66	410
12	sensor_LCS	32	84	521
13	sensor_LCS	22	78	596
14	sensor_LCS	32	78	455
15	sensor_LCS	33	56	433
16	sensor_LCS	27	45	406
17	sensor_LCS	22	57	417
18	sensor_LCS	32	86	445
19	sensor_LCS	34	57	418
20	sensor_LCS	23	65	468
21	sensor_LCS	24	38	582
22	sensor_LCS	35	80	527
23	sensor_LCS	32	74	515
24	sensor_LCS	25	50	470
25	sensor_LCS	34	63	428
26	sensor_LCS	27	65	598
27	sensor_LCS	34	35	529
28	sensor_LCS	35	94	597
29	sensor_LCS	26	67	520
30	sensor_LCS	33	59	455
31	sensor_LCS	34	39	568
32	sensor_LCS	31	50	493
33	sensor_LCS	26	42	505
34	sensor_LCS	23	64	485
35	sensor_LCS	35	59	426
36	sensor_LCS	24	67	407
37	sensor_LCS	24	44	404
38	sensor_LCS	28	35	422
39	sensor_LCS	30	35	470
40	sensor_LCS	28	58	511

APÊNDICE 8 – SCRIPT VALIDAÇÃO

```
#protocolo de validação de dados
library(readxl)
library(sqldf)
library(ggplot2)
library(FSA)
library(cowplot)
options(scipen = 999)
dados <- read_excel("E:/TESE/dados/dados_granjas.xlsx",
  sheet = "dados_3")
View(dados)
#resumo dos dados
summary(dados)
#contar quantos valores NA tem no dataframe
sapply(dados, function(x) sum(is.na(x)))
#para excluir LINHAS com os valores com NA:
dados <- na.omit(dados)
#reverificar quantos valores NA ficaram no dataframe
sapply(dados, function(x) sum(is.na(x)))
#contar quantas linhas e colunas tem no dataframe
dim(dados)
#eliminar colunas não usadas
dados$data = NULL
#exploratório sobre o total produzido
boxplot(dados$total_aves~dados$cod_granja)
#separando os grupos
controle01 <- sqldf("SELECT * FROM dados WHERE cod_granja = 'CONTROLE_01'")
controle02 <- sqldf("SELECT * FROM dados WHERE cod_granja = 'CONTROLE_02'")
g1001 <- sqldf("SELECT * FROM dados WHERE cod_granja = '1001'")
g1002 <- sqldf("SELECT * FROM dados WHERE cod_granja = '1002'")
g1003 <- sqldf("SELECT * FROM dados WHERE cod_granja = '1003'")
#descritivos remuneração por ave
Summarize(controle01$pag_ave)
Summarize(controle02$pag_ave)
Summarize(g1001$pag_ave)
Summarize(g1002$pag_ave)
Summarize(g1003$pag_ave)
#descritivo produção
Summarize(controle01$total_aves)
Summarize(controle02$total_aves)
Summarize(g1001$total_aves)
Summarize(g1002$total_aves)
Summarize(g1003$total_aves)
#mortalidade
Summarize(controle01$mortalidade)
Summarize(controle02$mortalidade)
Summarize(g1001$mortalidade)
Summarize(g1002$mortalidade)
Summarize(g1003$mortalidade)
#refugos
Summarize(controle01$refugo)
Summarize(controle02$refugo)
Summarize(g1001$refugo)
Summarize(g1002$refugo)
Summarize(g1003$refugo)
#IEP
Summarize(controle01$iep)
Summarize(controle02$iep)
Summarize(g1001$iep)
Summarize(g1002$iep)
Summarize(g1003$iep)
#CONVERSAO ALIMENTAR
Summarize(controle01$conv_alimentar)
Summarize(controle02$conv_alimentar)
Summarize(g1001$conv_alimentar)
Summarize(g1002$conv_alimentar)
Summarize(g1003$conv_alimentar)
#temperatura
Summarize(controle01$temp_med)
Summarize(controle02$temp_med)
Summarize(g1001$temp_med)
Summarize(g1002$temp_med)
Summarize(g1003$temp_med)
```

```

#umidade
Summarize(controle01$umid_med)
Summarize(controle02$umid_med)
Summarize(g1001$umid_med)
Summarize(g1002$umid_med)
Summarize(g1003$umid_med)
#boxplot - remuneração por ave
ggplot(dados, aes(y = pag_ave, x = cod_granja, fill = cod_granja)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(1.20,1.8)) +
  theme_classic(base_size = 18) +
  xlab("Granjas") +
  ylab("Remuneração por ave (R$)")
#boxplot - iep
ggplot(dados, aes(y = iep, x = cod_granja, fill = cod_granja)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(200,500)) +
  theme_classic(base_size = 18) +
  xlab("Granjas") +
  ylab("Índice de Eficiencia produtiva")
#boxplot - conversão alimentar
ggplot(dados, aes(y = conv_alimentar, x = cod_granja, fill = cod_granja)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(0.3,2)) +
  theme_classic(base_size = 18) +
  xlab("Granjas") +
  ylab("Conversão alimentar")
#dados2 - estatísticas descritivas
#separando os grupos
datalogger <- sqldf("SELECT * FROM dados2 WHERE cod_sensor = 'sensor_LCS'")
inobram <- sqldf("SELECT * FROM dados2 WHERE cod_sensor = 'sensor_inobram'")
#temperatura cruzando dados data logger x inobram
Summarize(datalogger$temp)
Summarize(inobram$temp)
#umidade cruzando dados data logger x inobram
Summarize(datalogger$umid)
Summarize(inobram$umid)
#CO2 cruzando dados data logger x inobram
Summarize(datalogger$CO2)
Summarize(inobram$CO2)
#boxplot - temperatura
b1 <- ggplot(dados2, aes(y = temp, x = cod_sensor, fill = cod_sensor)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(15,40)) +
  theme_classic(base_size = 14) +
  xlab("Temperatura") +
  ylab("°C")
b1
b2 <- ggplot(dados2, aes(y = umid, x = cod_sensor, fill = cod_sensor)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(35,98)) +
  theme_classic(base_size = 14) +
  xlab("Umidade") +
  ylab("%")
b3 <- ggplot(dados2, aes(y = CO2, x = cod_sensor, fill = cod_sensor)) +
  geom_boxplot(show.legend = F, alpha = .5) +
  scale_y_continuous(limits = c(395,600)) +
  theme_classic(base_size = 14) +
  xlab("CO2") +
  ylab("ppm")
#juntando os gráficos
#plot_grid(b1,b2,b3, ncol = 1)
plot_grid(b1,b2,b3, ncol = 1)
#####
##gráficos de linha TEMPERATURA
#####
linhaTDataLogger <- ggplot(line,
  aes(x = coleta))+
  geom_line(aes(y = tempDataLogger),
  col="red")+
  scale_y_continuous(limits = c(0,100)) +
  #scale_x_continuous(limits = c(1,500)) +
  xlab("medições")
linhaTDataLogger
linhaTInob <- ggplot(line,
  aes(x = coleta2))+

```

```

geom_line(aes(y = tempInobram),
  col="blue")+
scale_y_continuous(limits = c(0,100)) +
#scale_x_continuous(limits = c(1,500)) +
xlab("medições")
linhaTInob
#juntando
juntoT <- (plots=ggplot(line, aes(x = coleta)) +
  geom_line(aes(y = tempDataLogger,
    colour = "Data Logger"))+
  scale_x_continuous(limits= c(1,45)) +
  scale_y_continuous(limits = c(22,37)) +
  geom_line(aes(y = tempInobram,
    colour = "Sistema comercial")))+
xlab("Dias") +
ylab("Temperatura °C")+
scale_colour_manual("",
  breaks = c("Data Logger", "Sistema comercial"),
  values = c("red", "blue"))+
theme_classic(base_size = 14) +
theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank())
#####
#####
#####
##gráficos de linha Umidade
#####
linhaUDataLogger <- ggplot(line,
  aes(x = coleta))+
geom_line(aes(y = umidDataLogger),
  col="red")+
scale_y_continuous(limits = c(0,100)) +
#scale_x_continuous(limits = c(1,500)) +
xlab("medições")
linhaUDataLogger
linhaUInob <- ggplot(line,
  aes(x = coleta2))+
geom_line(aes(y = UmidInobram),
  col="blue")+
scale_y_continuous(limits = c(0,100)) +
#scale_x_continuous(limits = c(1,500)) +
xlab("medições")
linhaUInob
#juntando
juntoU <- (plots=ggplot(line, aes(x = coleta)) +
  geom_line(aes(y = umidDataLogger,
    colour = "Data Logger"))+
  scale_x_continuous(limits= c(1,45)) +
  scale_y_continuous(limits = c(33,100)) +
  geom_line(aes(y = UmidInobram,
    colour = "Sistema comercial")))+
xlab("Dias") +
ylab("Umidade %")+
scale_colour_manual("",
  breaks = c("Data Logger", "Sistema comercial"),
  values = c("red", "blue"))+
theme_classic(base_size = 14) +
theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank())
juntoU
#####
#####
#####
##gráficos de linha CO2
#####
linhaCO2DataLogger <- ggplot(line,
  aes(x = coleta))+
geom_line(aes(y = CO2DataLogger),
  col="red")+
scale_y_continuous(limits = c(400,600)) +
#scale_x_continuous(limits = c(1,500)) +
xlab("medições")
linhaCO2DataLogger
linhaTInob <- ggplot(line,
  aes(x = coleta2))+
geom_line(aes(y = CO2Inobram),

```

```

col="blue")+
scale_y_continuous(limits = c(400,610)) +
#scale_x_continuous(limits = c(1,500)) +
xlab("medições")
linhaTinob
#juntando
juntoCO2 <- (plots=ggplot(line, aes(x = coleta)) +
  geom_line(aes(y = CO2DataLogger,
    colour = "Data Logger"))+
  scale_x_continuous(limits= c(1,45)) +
  scale_y_continuous(limits = c(400,610)) +
  geom_line(aes(y = CO2Inobram,
    colour = "Sistema comercial")))+
xlab("Dias") +
ylab("CO2 ppm")+
scale_colour_manual("",
  breaks = c("Data Logger", "Sistema comercial"),
  values = c("red", "blue"))+
theme_classic(base_size = 14) +
theme(panel.grid.major = element_blank(),
  panel.grid.minor = element_blank())
juntoCO2
#####
#####
#####
#plotando todos os gráficos juntos
#####
plot_grid(juntoT, juntoU, juntoCO2, ncol = 1)
##cruzamento de dados
##produtor 2 sem iot x produtor 2 com iot
#separando os grupos
controle01IoT <- sqldf("SELECT * FROM dados WHERE id_lote = 'LT_LUCAS'")
controle01gr01 <- sqldf("SELECT * FROM dados WHERE id_lote = 'GR_01'")
controle01gr02 <- sqldf("SELECT * FROM dados WHERE id_lote = 'GR_02'")
controle01gr03 <- sqldf("SELECT * FROM dados WHERE id_lote = 'GR_03'")
controle01gr04 <- sqldf("SELECT * FROM dados WHERE id_lote = 'GR_04'")
#####
##normalidade e homogeneidade da variância
#####
shapiro.test(controle01$pag_ave)
qqnorm(controle01$pag_ave)
qqline(controle01$pag_ave)
bartlett.test(mann$pag_ave~mann$lote)
###cruzamento dos dados somente produtor 2
wilcox.test(mann$pag_ave~mann$lote)
#####
##cruzamento de dados
##produtor 2 com iot x todos os produtores sem IoT
#####
##normalidade e homogeneidade da variância
#####
shapiro.test(mann$pag_ave)
qqnorm(mann$pag_ave)
qqline(mann$pag_ave)
###cruzamento dos dados somente dos pequenos produtores
wilcox.test(mann$pag_ave~mann$lote)
##normalidade data logger x inobram
#temperatura
shapiro.test(dados2$temp)
qqnorm(dados2$temp)
qqline(dados2$temp)
shapiro.test(dados2$umid)
qqnorm(dados2$umid)
qqline(dados2$umid)
shapiro.test(dados2$CO2)
qqnorm(dados2$CO2)
qqline(dados2$CO2)
hist(dados2$CO2)
bartlett.test(dados2$temp~dados2$cod_sensor)
t.test(dados2$temp~dados2$cod_sensor)

```

APÊNDICE 9 – DOCUMENTAÇÃO API

API de integração de dados IoT - FARM UFPR 4.0.

0.0.1 - alfa version **GA3**

Projeto de tese do PPGGI UFPR - Discente de doutorado Lucas José de Souza sob a orientação do Professor Dr. Egon Walter Wildauer. A API tem por objetivo a manipulação de dados advindos do sistema embarcado de coleta de dados e demais funcionalidades prototipado para a pesquisa. De forma a permitir a gestão das informações com a extração de dados e construção de conhecimento.

[Terms of service](#)
[Contact the developer](#)

Servers

<https://apiagro-backend.herokuapp.com/> - API de produção
<https://apiagro-backend.herokuapp.com/> - API de produção
<http://localhost:3000/v1/> - API de teste - RaspberryPI

POST	/session	validar um usuário para acessar a aplicação	▼
POST	/users	cadastro de usuários da aplicação	▼
GET	/users	Listar usuários	▼
POST	/farms	Cadastro dos produtores	▼
GET	/farms	Listagem das fazendas	▼
PUT	/farms	Alteração de dados	▼
DELETE	/farms	Exclusão de fazendas	▼
POST	/sensores	Cadastro dos sensores IoT	▼
GET	/sensores	Busca de dados dos sensores	▼
POST	/actuator	cadastro dos atuadores IoT	▼
GET	/actuator	Busca de dados dos atuadores	▼
PUT	/actuator	Alteração de dados	▼
POST	/helps	Cadastro chamados do usuário para o especialista	▼
GET	/helps	Listar chamados	▼
POST	/notification	Envio de notificações para o aplicativo	▼
GET	/notification	Lista de notificações	▼

Schemas

```
Users {
  name      string
  e-mail    string
  createdat timestamp
}
```

```
Session {
  id        string
  name     string
  e-mail    string
}
```

```
Farm {
  user          string
  fazenda      string
  cidade       string
  responsavel  string
  cultivos     string
  bio          string
  avatar_url   string
  coordinates  [ {
    } ]
}
```

```
Sensor {
  user_id      string
  timestamp   timestamp
  tipoMedida1 string
  tipoMedida2 string
  unidadeMedida1 string
  unidadeMedida2 string
  modeloSensor string
  valor1      number
  valor2      number
}
```

```
Actuator {
  user          string
  actuator_name string
  timestamp    timestamp
  state        boolean
}
```

```
Help {
  user          string
  assuntoProblema string
  descricaoProblema string
}
```

```
Notification {
  user          string
  content       string
}
```



API de integração de dados IoT - FARM UFPR 4.0.

0.0.1 - alfa version OAS3

Projeto de tese do PPGGI UFPR - Discente de doutorado Lucas José de Souza sob a orientação do Professor Dr. Egon Walter Wildauer. A API tem por objetivo a manipulação de dados advindos do sistema embarcado de coleta de dados e demais funcionalidades prototipado para a pesquisa. De forma a permitir a gestão das informações com a extração de dados e construção de conhecimento.

[Terms of service](#)

[Contact the developer](#)

Servers

<https://apiagro-backend.herokuapp.com/> - API de produção

default



POST /session validar um usuário para acessar a aplicação



Rota para validação de usuário

Parameters

Try it out

No parameters

Request body

application/json

Examples:

session

Example Value Schema

```
{
  "id": "345a7s6d5d426e",
  "name": "Lucas José",
  "e-mail": "lucas@gmail.com"
}
```

Responses

Code	Description	Links
200	Token gerado com sucesso!	No links
400	Usuário e/ou Senha incorreto(s)	No links
401	Usuário não existe	No links

POST /users cadastro de usuários da aplicação

Rota para o cadastro de usuários do sistema

Parameters[Try it out](#)

No parameters

Request body

[application/json](#)**Examples:**[users](#)**Example Value** Schema

```
{
  "name": "Lucas José",
  "e-mail": "lucas@gmail.com",
  "createdAt": "05/12/2022 08:30"
}
```

Responses

Code	Description	Links
200	Usuário cadastrado com com sucesso!	No links

Code	Description	Links
400	Usuário já cadastrado	No links
401	Falha na validação	No links

GET /users Listar usuários ^

Rota para a listagem dos usuários cadastrados na plataforma

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Busca realizada com sucesso	No links
400	Falha ao realizar a busca! Contate o administrador ou redefina os parâmetros.	No links
401	Não autorizado	No links

POST /farms Cadastro dos produtores ^

Rota para o usuário cadastrar sua fazenda e incluir demais informações relevantes

Parameters Try it out

No parameters

Request body

application/json

Examples:

farms

Example Value Schema

```
{
  "user": "Professor Egon",
  "fazenda": "Fazenda 4.0 UFPR - Palotina",
  "cidade": "Palotina-PR",
  "responsavel": "Professor Mauricio",
  "cultivos": "Frango de corte",
  "bio": "Fazenda criada em 2022 para o desenvolvimento de sistemas IoT de baixo custo",
  "avatar_ur": "colocar a URL do avatar do usuário",
  "coordinates": [
    -53.8432086,
    -24.2961452
  ]
}
```

Responses

Code	Description	Links
200	Fazenda Cadastrada com Sucesso	No links
400	Essa fazenda já está cadastrada	No links
401	Usuário não autorizado	No links

GET /farms Listagem das fazendas

Rota para listar todas as fazendas cadastradas por produtores

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Busca realizada com sucesso!	No links
400	Falha ao realizar a busca! Contate o administrador ou redefina os parâmetros.	No links
401	Usuário não autorizado	No links

PUT /farms Alteração de dados



Rota para editar os dados das fazendas cadastradas

Parameters

[Try it out](#)

No parameters

Request body

[application/json](#)

Examples:

[farms](#)

Example Value Schema

```
{
  "user": "Professor Egon",
  "fazenda": "Fazenda 4.0 UFPR - Palotina",
  "cidade": "Palotina-PR",
  "responsavel": "Professor Mauricio",
  "cultivos": "Frango de corte",
  "bio": "Alteração da bio realizada com sucesso (teste)",
  "avatar_ur": "colocar a URL do avatar do usuário",
  "coordinates": [
    -53.8432086,
    -24.2961452
  ]
}
```

Responses

Code	Description	Links
200	Dados atualizados com sucesso.	No links
400	Não foi possível realizar a alteração.	No links
401	Usuário não autorizado	No links

DELETE /farms Exclusão de fazendas

Rota para excluir as fazendas

Parameters

Try it out

No parameters

Request body

application/json

Examples:

farms

Example Value Schema

```
{
  "user": "Professor Egon",
  "fazenda": "Fazenda 4.0 UFPR - Palotina",
  "cidade": "Palotina-PR",
  "responsavel": "Professor Mauricio",
  "cultivos": "Frango de corte",
  "bio": "Fazenda criada em 2022 para o desenvolvimento de sistemas IoT de baixo custo",
  "avatar_ur": "colocar a URL do avatar do usuário",
  "coordinates": [
    -53.8432086,
    -24.2961452
  ]
}
```

Responses

Code	Description	Links
200	Exclusão realizada com sucesso!	No links
400	Não foi possível excluir essa fazenda.	No links
401	Usuário não autorizado	No links

POST /sensores Cadastro dos sensores IoT

Rota para o usuário cadastrar os sensores e suas especificações

Parameters

Try it out

No parameters

Request body

application/json

Examples:

sensors

Example Value Schema

```
{
  "user_id": "bw23a6751326a6516537a65135237",
  "timestamp": "10/23/2022 17:45",
  "tipoMedida1": "Temperatura",
  "tipoMedida2": "Umidade ambiente",
  "unidadeMedida1": "°C",
  "unidadeMedida2": "%",
  "modeloSensor": "Adafruit bmht11",
  "valor1": 23,
  "valor2": 74,
  "ipArduino": "192.168.1.1"
}
```

Responses

Code	Description	Links
200	Sensor cadastrado com sucesso!	No links

Code	Description	Links
400	Esse sensor já está cadastrado	No links
401	Usuário não autorizado	No links

GET /sensores Busca de dados dos sensores

Rota para filtrar os dados dos sensores com diversos parâmetros

Parameters

Try it out

No parameters

Request body

application/json

Examples:

sensors

Example Value Schema

```
[
  {
    "user_id": "bw23a6751326a6516537a65135237",
    "timestamp": "10/23/2022 17:45",
    "tipoMedida1": "Temperatura",
    "tipoMedida2": "Umidade ambiente",
    "unidadeMedida1": "°C",
    "unidadeMedida2": "%",
    "modeloSensor": "Adafruit bmht11",
    "valor1": 23,
    "valor2": 74,
    "ipArduino": "192.168.1.1"
  },
  {
    "user_id": "bw23a6751326a6516537a65135237",
    "timestamp": "11/23/2022 17:45",
    "tipoMedida1": "Temperatura",
    "tipoMedida2": "Umidade ambiente",
    "unidadeMedida1": "°C",
    "unidadeMedida2": "%",
    "modeloSensor": "Adafruit bmht11",
    "valor1": 21,
    "valor2": 87,
    "ipArduino": "192.168.1.1"
  },
  {
    "user_id": "bw23a6751326a6516537a65135237",
    "timestamp": "12/23/2022 17:45",
```

Responses

Code	Description	Links
200	Busca realizada com sucesso!	No links
400	Falha ao realizar a busca! Contate o administrador ou redefina os parâmetros.	No links
401	Usuário não autorizado	No links

POST /actuator cadastro dos atuadores IoT

Rota para o usuário cadastrar os atuadores e suas especificações

Parameters[Try it out](#)

No parameters

Request body

[application/json](#)**Examples:**[actuators](#)

Example Value Schema

```
{
  "user_id": "b32yt2f3y12t123tf2",
  "actuator_name": "Motor Weg - MD13E03",
  "timestamp": "11/11/2022 11:35",
  "state": true
}
```

Responses

Code	Description	Links
200	Atuador cadastrado com sucesso!	No links

Code	Description	Links
400	Esse atuador já está cadastrado	No links
401	Usuário não autorizado	No links

GET /actuator Busca de dados dos atuadores ^

Rota para filtrar os dados dos atuadores cadastrados e seus estados

Parameters Try it out

No parameters

Responses

Code	Description	Links
200	Busca realizada com sucesso!	No links
400	Falha ao realizar a busca! Contate o administrador ou redefina os parâmetros.	No links
401	Usuário não autorizado	No links

PUT /actuator Alteração de dados ^

Rota para editar o estado dos atuadores cadastrados

Parameters Try it out

No parameters

Request body

application/json

Examples:

actuators

Example Value Schema

```
[ {  
  "user_id": "b32yt2f3y12t123tf2",  
  "actuator_name": "Teste de mudança de nome",  
  "timestamp": "18/12/2022 11:35",  
  "state": true  
}
```

Responses

Code	Description	Links
200	Dados atualizados com sucesso.	No links
400	Não foi possível realizar a alteração.	No links
401	Usuário não autorizado	No links

POST /helps Cadastro chamados do usuário para o especialista

Rota para o usuário cadastrar chamados

Parameters

Try it out

No parameters

Request body

application/json

Examples:

helps

Example Value Schema

```
{
  "user_id": "bb28c1g241b1vb21b",
  "assuntoProblema": "Problema sensor",
  "descricaoProblema": "Houve um alagamento devido as chuvas do dia 01/10/2022 o qual ocasiono
u a falha no sensor dht11"
}
```

Responses

Code	Description	Links
200	Chamado cadastrado com sucesso!	No links
401	Usuário não autorizado	No links

GET /helps Listar chamados



Rota para a listagem de chamados registrados

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Busca realizada com sucesso	No links
400	Falha ao realizar a busca! Contate o administrador ou redefina os parâmetros.	No links
401	Usuário não autorizado	No links

POST /notification Envio de notificações para o aplicativo



Rota para o cadastro e envio de notificação para os usuários

Parameters

[Try it out](#)

No parameters

Request body

[application/json](#)

Examples:

[notification](#)

Example Value Schema

```
{
  "user_id": "bb28c1g241b1vb21b",
  "content": "A tendência é do NH3 subir os níveis nos próximos 30 minutos, a ventilação foi automaticamente acionada. Para mais informações acesse o Aplicativo"
}
```

Responses

Code	Description	Links
200	Notificação enviada com sucesso!	No links
401	Usuário não autorizado	No links

GET /notification Lista de notificações



Rota para listar o histórico de notificações enviadas aos usuários

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Notificação enviada com sucesso!	No links
401	Usuário não autorizado	No links

Schemas



```
Users {
  name      string
  e-mail    string
  createdAt timestamp
}

Session {
  id        string
  name      string
  e-mail    string
}

Farm {
  user        string
  fazenda    string
  cidade      string
  responsavel string
  cultivos   string
  bio        string
  avatar_url string
  coordinates [ U ]
}

Sensor {
  user_id      string
  timestamp    timestamp
  tipoMedida1  string
  tipoMedida2  string
  unidadeMedida1 string
  unidadeMedida2 string
  modeloSensor string
  valor1       number
  valor2       number
}
```

```
Actuator {  
  user           string  
  actuator_name string  
  timestamp      timestamp  
  state          boolean  
}
```

```
Help {  
  user           string  
  assuntoProblema string  
  descricaoProblema string  
}
```

```
Notification {  
  user           string  
  content        string  
}
```

APÊNDICE 10 – DIAGRAMA DE SEQUÊNCIA CADASTRO E LOGIN

