

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA – PPGEEL**

TATIANA PEREIRA FILGUEIRAS

**SMACARE - SISTEMA MULTIAGENTE COM INFERÊNCIA NEURAL PARA APOIO
DIAGNÓSTICO EM TEMPO REAL E PROGNÓSTICO CLÍNICO**

JOINVILLE

2025

TATIANA PEREIRA FILGUEIRAS

**SMACARE - SISTEMA MULTIAGENTE COM INFERÊNCIA NEURAL PARA APOIO
DIAGNÓSTICO EM TEMPO REAL E PROGNÓSTICO CLÍNICO**

Tese apresentada como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGEEL, da Universidade do Estado de Santa Catarina – UDESC.

Orientador: Prof. Dr. Pedro Bertemes Filho

**JOINVILLE
2025**

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Universitária Udesc,
com os dados fornecidos pelo(a) autor(a)**

FILGUEIRAS, Tatiana Pereira
SMACARE - SISTEMA MULTIAGENTE COM
INFERÊNCIA NEURAL PARA APOIO DIAGNÓSTICO EM
TEMPO REAL E PROGNÓSTICO CLÍNICO / Tatiana Pereira
FILGUEIRAS. -- 2025.
169 p.

Orientador: Pedro Bertemes-Filho
Tese (doutorado) -- Universidade do Estado de Santa
Catarina, Centro de Ciências Tecnológicas, Programa de
Pós-Graduação em Engenharia Elétrica, Joinville, 2025.

1. Sistema Multiagente. 2. Internet das Coisas. 3.
Inteligência Artificial. 4. Wearable. 5. Diagnóstico/Prognóstico.
I. Bertemes-Filho, Pedro. II. Universidade do Estado de Santa
Catarina, Centro de Ciências Tecnológicas, Programa de
Pós-Graduação em Engenharia Elétrica. III. Título.

TATIANA PEREIRA FILGUEIRAS

SMACARE - SISTEMA MULTIAGENTE COM INFERÊNCIA NEURAL PARA APOIO DIAGNÓSTICO EM TEMPO REAL E PROGNÓSTICO CLÍNICO

Tese apresentada como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGEEL, da Universidade do Estado de Santa Catarina – UDESC.

Orientador: Prof. Dr. Pedro Bertemes Filho

BANCA EXAMINADORA

Membros:

Prof. Dr. Pedro Bertemes Filho
Universidade do Estado de Santa Catarina
Presidente/Orientador

Prof. Dr. Charles Christian Miers
Universidade do Estado de Santa Catarina
Membro Interno

Prof. Dr. Fabrício Noveletto
Universidade do Estado de Santa Catarina
Membro Interno

Profa. Dra. Sandra Cossul
Universidade Estadual de Maringá
Membro Externo

Prof. Dr. Roberto Heinze
Centro Universitário de Brusque
Membro Externo

Joinville, 24 de Outubro de 2025

À minha querida mãe, Florice Custodio Pereira
Filgueiras. Sinto sua falta. Com amor.

AGRADECIMENTOS

Agradeço, primeiramente, a Jeová, meu Deus, por cada dia de vida, saúde, e pelo conhecimento que Ele me concedeu para concluir esta pesquisa. Também expresso minha gratidão a seu filho, Jesus Cristo, pelo resgate pago por mim.

À minha querida mãe, Florice, minha eterna gratidão. Seu apoio incondicional e encorajamento foram o que me sustentou e motivou a superar meus próprios limites. Ainda que ela tenha falecido nas etapas finais deste trabalho, sua perseverança e seu amor foram o impulso para cada passo que dei. Esta conquista é tanto minha quanto dela.

Agradeço ao meu orientador, Prof. Dr. Pedro Bertemes Filho por aceitar conduzir esta pesquisa e por todo o suporte técnico e acadêmico ao longo do desenvolvimento deste trabalho. Também agradeço ao Prof. Dr. Fabrício Noveletto pelas contribuições que permitiram que esse trabalho ficasse pronto. Agradeço também ao Prof. Dr. André Bittencourt Leal pelo constante apoio na resolução de dúvidas ao longo desta jornada e, em especial, por viabilizar os recursos necessários para a publicação do meu primeiro artigo em periódico científico. Sua colaboração foi fundamental para esse importante marco na minha trajetória acadêmica.

Estendo meus agradecimentos a todos os professores do Programa de Doutorado em Engenharia Elétrica da Universidade do Estado de Santa Catarina – UDESC, pela excelência na formação oferecida e pela qualidade técnica presente em cada disciplina ministrada, que contribuíram significativamente para minha formação como pesquisadora.

Agradeço, de forma muito especial, ao meu cunhado, Stefano La Porta, por seu apoio constante, incentivo incondicional e presença fundamental em todas as etapas desta jornada. Sua ajuda foi decisiva para que eu pudesse chegar até aqui.

Sou também profundamente grata ao meu pai e à minha irmã, cujo apoio emocional e motivação foram essenciais para que eu mantivesse a perseverança e a confiança ao longo do percurso.

À Profa. Dra. Vivian Siffert Wildner, deixo um agradecimento sincero pelo auxílio voluntário prestado ao longo do curso, especialmente nas etapas finais desta tese. Sua orientação e disponibilidade foram absolutamente indispensáveis e marcaram de forma significativa este processo. Sem sua colaboração, este trabalho não teria sido possível.

Registro minha gratidão ao Sr. Rafael Marchan, pelo auxílio em viabilizar a parceria com o Lar Divina Providência, que foi essencial para a realização da etapa experimental da pesquisa.

Agradeço também ao Lar Divina Providência, localizado em Navegantes, SC, por disponibilizar seu espaço e acolher com generosidade os testes necessários à validação deste trabalho.

Por fim, expresso minha sincera gratidão a todos os meus irmãos na fé da Congregação Gravatá das Testemunhas de Jeová, em Navegantes, SC, que me apoiaram com ajuda prática e amizade sincera em todas as fases desta jornada acadêmica.

A todos, o meu mais profundo agradecimento. Muito obrigada!

“I have not failed. I’ve just found 10,000 ways that won’t work.” (EDISON, Thomas [1910])

RESUMO

O envelhecimento populacional e o aumento da prevalência de doenças crônicas têm ampliado a demanda por soluções capazes de realizar o monitoramento remoto e contínuo de sinais vitais. No entanto, os sistemas atuais apresentam limitações estruturais significativas, como arquiteturas centralizadas pouco escaláveis, baixa adaptabilidade frente à heterogeneidade dos dados e ausência de mecanismos inteligentes capazes de integrar múltiplos sinais em tempo real. Tais restrições comprometem a precisão diagnóstica e a personalização de prognósticos, especialmente em cenários dinâmicos como o cuidado domiciliar. Com o propósito de superar essas lacunas, esta tese propõe o SMACare, um Sistema Multiagente que integra dados coletados por um dispositivo vestível de baixo custo desenvolvido com modelos de redes neurais artificiais. A arquitetura emprega comunicação de baixo consumo energético por meio dos protocolos BLE e MQTT, organiza agentes especializados para as etapas de coleta, pré-processamento, análise e tomada de decisão, e utiliza um banco de dados PostgreSQL para o armazenamento e gerenciamento histórico das informações. Essa abordagem descentralizada proporciona maior robustez, flexibilidade e capacidade adaptativa diante da variabilidade inerente aos sinais fisiológicos. Os experimentos foram conduzidos com 30 participantes, abrangendo duas etapas principais: (i) diagnóstico em tempo real a partir dos sinais de frequência cardíaca, SpO₂, temperatura e pressão arterial estimada, e (ii) geração de prognósticos clínicos baseados no histórico de dados coletados. Os modelos de aprendizado profundo empregados alcançaram uma acurácia média de 95,6% na classificação de sinais vitais, 93,4% no diagnóstico multi-label e 91,8% na previsão prognóstica, com tempo médio de inferência inferior a 0,15 segundos por modelo. Além disso, a arquitetura do SMACare apresentou latência média inferior a 2 segundos para sinais individuais e 3 segundos para diagnósticos e prognósticos, possibilitando respostas rápidas e contextualizadas. Os resultados demonstraram a viabilidade e relevância do SMACare para a integração de monitoramento inteligente e geração de prognósticos personalizados, evidenciando avanços em termos de custo, modularidade e aplicabilidade em ambientes domiciliares, contribuindo para a descentralização e agilização do processo de tomada de decisão clínica.

Palavras-chave: Sistema Multiagente. Internet das Coisas (IoT). Dispositivos Vestíveis. Inteligência Artificial. Redes Neurais Artificiais. Sinais Vitais. Diagnóstico em Tempo Real. Prognóstico Clínico.

ABSTRACT

Population aging and the rising prevalence of chronic diseases have increased the demand for solutions capable of performing remote and continuous monitoring of vital signs. However, current systems present significant structural limitations, such as centralized architectures with low scalability, limited adaptability to data heterogeneity, and the absence of intelligent mechanisms capable of integrating multiple signals in real time. These constraints compromise diagnostic accuracy and the personalization of prognostics, especially in dynamic scenarios such as home care. To address these gaps, this thesis proposes SMACare, a Multi-Agent System that integrates data collected by a low-cost wearable device developed in this work with artificial neural network models. The architecture employs low-power communication protocols (BLE and MQTT), organizes specialized agents for the stages of data collection, preprocessing, analysis, and decision-making, and uses a PostgreSQL database for historical data storage and management. This decentralized approach offers greater robustness, flexibility, and adaptive capacity in response to the inherent variability of physiological signals. Experiments were conducted with 30 participants, covering two main stages: (i) real-time diagnosis based on heart rate, SpO₂, temperature, and estimated blood pressure signals, and (ii) clinical prognostic generation based on the historical data collected. The deep learning models employed achieved an average accuracy of 95.6% in vital sign classification, 93.4% in multi-label diagnosis, and 91.8% in prognostic prediction, with an average inference time of less than 0.15 seconds per model. Furthermore, the SMACare architecture demonstrated an average latency of less than 2 seconds for individual signals and 3 seconds for diagnoses and prognostics, enabling fast and contextualized responses. The results demonstrated the feasibility and relevance of SMACare for integrating intelligent monitoring and generating personalized prognostics, highlighting advancements in terms of cost, modularity, and applicability in home environments, thereby contributing to the decentralization and acceleration of clinical decision-making processes.

Keywords: Multi-Agent System. Internet of Things (IoT). Wearable Devices. Artificial Intelligence. Artificial Neural Networks. Vital Signals. Real-Time Diagnosis. Clinical Prognosis.

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Funcionamento do Protocolo AMQP | 30 |
| Figura 2 – Funcionamento do Protocolo CoAP | 31 |
| Figura 3 – Funcionamento do Protocolo MQTT | 32 |
| Figura 4 – Exemplo de Arquitetura de Sistemas Embarcados | 39 |
| Figura 5 – Módulo display Oled 128x64 nas cores de led amarelo e azul. | 43 |
| Figura 6 – Vistas frontal e traseira do módulo H8-003. | 44 |
| Figura 7 – Vista frontal do módulo DS3231. | 45 |
| Figura 8 – Vistas frontal e traseira do módulo TP4056. | 46 |
| Figura 9 – Vista frontal do módulo PC-MED-0411. | 47 |
| Figura 10 – Estrutura de um SMA. | 49 |
| Figura 11 – Ilustrações do dispositivo vestível fabricado. (a) Vista traseira da placa, mostrando a interface de 6 pinos do módulo MAX32664 (1), o módulo de gerenciamento de energia (2), a bateria (3), o sensor infravermelho (IR) (4) e a interface para cartão microSD (5). (b) Vista frontal da placa, incluindo o relógio em tempo real (RTC) (6), o microcontrolador nRF52840 (7), o botão para reativar o visor a partir do modo de espera (8), o interruptor liga/desliga (9) e o visor (10). | 59 |
| Figura 12 – Funcionamento do SMACare - Cenário 1 | 63 |
| Figura 13 – Diagrama de sequência das etapas de login e cadastro, válido para ambos os Cenários | 63 |
| Figura 14 – Diagrama de sequência das etapas pós login e preparação do sistema para as inferências | 64 |
| Figura 15 – Diagrama de sequência das etapas de Inferência e Notificação | 65 |
| Figura 16 – Diagrama de sequência da etapa de Prognóstico | 66 |
| Figura 17 – Funcionamento do SMACare - Cenário 2 | 68 |
| Figura 18 – Diagrama de sequência das etapas pós login e preparação do sistema para as inferências | 69 |
| Figura 19 – Diagrama de sequência das etapas de Inferência | 70 |
| Figura 20 – Diagrama de sequência das etapas de inferência por linha de arquivo | 71 |
| Figura 21 – Diagrama de sequência da etapa de Prognóstico | 72 |
| Figura 22 – Modelo Entidade-Relacionamento (MER) do banco de dados do sistema SMACare, ilustrando as entidades estáticas, as tabelas geradas dinamicamente para cada par dispositivo-paciente e a integração com os módulos responsáveis por armazenamento, diagnóstico, prognóstico e notificações. | 74 |
| Figura 23 – Tempo médio de inferência e acurácia dos modelos para classificação de frequência cardíaca. | 81 |

| | |
|--|-----|
| Figura 24 – Tempo médio de inferência e acurácia dos modelos para classificação de pressão arterial estimada. | 81 |
| Figura 25 – Tempo médio de inferência e acurácia dos modelos para classificação de SpO2. | 82 |
| Figura 26 – Tempo médio de inferência e acurácia dos modelos para classificação de temperatura. | 82 |
| Figura 27 – Tempo médio de inferência e acurácia dos modelos para classificação de prognóstico. | 84 |
| Figura 28 – Tempo médio de inferência e acurácia dos modelos para classificação de diagnóstico. | 84 |
| Figura 29 – Diagrama da Rede Neural Feedforward (MLP) para o sinal de pressão arterial estimada (BPT), com três camadas densas: 16, 8 e 3 neurônios, respectivamente. | 90 |
| Figura 30 – Diagrama da Rede Neural Feedforward (MLP) para classificação de SpO2 com camadas Dense(8) e Dense(4). | 93 |
| Figura 31 – Diagrama da Rede Neural Feedforward (MLP) para classificação de temperatura corporal com camadas Dense(16), Dense(16) e Dense(8). | 95 |
| Figura 32 – Diagrama da Rede Neural Feedforward (MLP) para classificação de frequência cardíaca com camadas Dense(16), Dense(16) e Dense(8). | 97 |
| Figura 33 – Arquitetura da rede neural para prognóstico clínico com entrada multivariada, camadas densas com normalização e saída com classificação em categorias clínicas. | 102 |
| Figura 34 – Protótipo do dispositivo vestível desenvolvido neste estudo: (a) o sistema vestível em uso; (b) o protótipo vestível e a cinta de fixação; e (c) a interface gráfica do aplicativo móvel personalizado. | 105 |
| Figura 35 – <i>Heatmap</i> dos tempos médios de notificação (s) para BPM (Frequência Cardíaca), BPT (Pressão Arterial Estimada), SpO2 e Temperatura em diferentes sujeitos monitorados pelo SMA. | 109 |
| Figura 36 – <i>Heatmap</i> do desvio padrão dos tempos de notificação (s) para BPM (Frequência Cardíaca), BPT (Pressão Arterial Estimada), SpO2 e Temperatura em diferentes sujeitos monitorados no Sistema Multiagente (SMA). | 112 |
| Figura 37 – <i>Heatmap</i> dos tempos de notificação (s) para Prognóstico e Diagnóstico Multi-label em diferentes sujeitos monitorados no Sistema Multiagente (SMA). | 113 |
| Figura 38 – <i>Heatmap</i> do desvio padrão dos tempos de notificação(s) para Diagnóstico Multi-label e Prognóstico, em diferentes sujeitos monitorados no Sistema Multiagente (SMA). | 114 |
| Figura 39 – Matriz de confusão para a classificação do sinal BPM (frequência cardíaca) em três categorias: Bradicardia, Normal e Taquicardia | 116 |
| Figura 40 – Matriz de confusão para a classificação do sinal BPT (pressão arterial estimada) em três categorias: Hipertensão, Hipotensão e Normal. | 117 |
| Figura 41 – Matriz de confusão para a classificação do sinal SpO2 em seis categorias. | 119 |

| | |
|---|-----|
| Figura 42 – Matriz de confusão para a classificação do sinal Temperatura Corporal em oito categorias. | 120 |
| Figura 43 – Matriz de confusão para a classificação de Diagnóstico Multi-label. | 122 |
| Figura 44 – Matriz de confusão para a classificação de Prognóstico. | 123 |

LISTA DE TABELAS

| | |
|---|-----|
| Tabela 1 – Alguns sensores utilizados na IoT. | 35 |
| Tabela 2 – Resumo dos valores médios de acurácia e dos desvios padrão observados no treinamento dos modelos de classificação para os sinais fisiológicos. | 77 |
| Tabela 3 – Resumo dos valores médios de acurácia e dos desvios padrão observados no treinamento dos modelos de classificação para diagnóstico multi-label e prognóstico. | 78 |
| Tabela 4 – Resumo das métricas de acurácia e do tempo médio de inferência por modelo e sinal fisiológico. | 80 |
| Tabela 5 – Resultados comparativos de tempo de inferência, variabilidade e acurácia dos modelos de diagnóstico e prognóstico. | 83 |
| Tabela 6 – Características demográficas, antropométricas e clínicas dos trinta participantes do estudo. | 104 |
| Tabela 7 – Percentual de acertos por participante nos diagnósticos baseados em sinais fisiológicos e na saúde autorelatada. | 107 |
| Tabela 8 – Valores médios do tempo de notificação (em segundos) por participante, considerando sinais fisiológicos (BPM, BPT, SpO ₂ , Temperatura) e os tempos de resposta de Diagnóstico Multi-label e Prognóstico. | 110 |
| Tabela 9 – Resultados dos desvios padrão do tempo de notificação (em segundos) por participante para cada sinal fisiológico e para as tarefas de Diagnóstico Multi-label e Prognóstico no sistema proposto. | 111 |
| Tabela 10 – Comparação entre trabalhos correlatos e o trabalho desta tese – Parte 1 . . . | 125 |
| Tabela 11 – Comparação entre trabalhos correlatos e o trabalho desta tese – Parte 2 . . . | 127 |
| Tabela 12 – Quadro-resumo do escopo funcional e conceitual do sistema SMACare: delimitação entre o que o sistema é/não é | 128 |
| Tabela 13 – Resumo das funcionalidades implementadas e das ações não contempladas pelo sistema SMACare, com o objetivo de esclarecer suas capacidades e delimitações práticas. | 129 |
| Tabela 14 – Explicação dos principais termos técnicos usados no treinamento do modelo de prognóstico | 145 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|--|
| AI | Artificial Intelligence (Inteligência Artificial) |
| AMQP | Advanced Message Queuing Protocol |
| ANN | Artificial Neural Network (Rede Neural Artificial) |
| API | Application Programming Interface |
| BLE | Bluetooth Low Energy |
| BPM | Beats Per Minute (Batimentos por Minuto) |
| BPT | Blood Pressure Trend (Pressão Arterial Estimada) |
| CoAP | Constrained Application Protocol |
| CSV | Formato de Planilha de Arquivo |
| DCN | Deep & Cross Network |
| DNN | Deep Neural Network |
| ECG | Electrocardiogram (Eletrocardiograma) |
| GNN | Graph Neural Network |
| GPU | Graphics Processing Unit |
| HTTP | HyperText Transfer Protocol |
| IA | Inteligência Artificial |
| IN | Instrução Normativa |
| IoT | Internet of Things (Internet das Coisas) |
| JADE | Java Agent DEvelopment Framework |
| LSTM | Long Short-Term Memory |
| MAC | Medium Access Control |
| MAPE | Mean Absolute Percentage Error |
| MLP | Multilayer Perceptron |
| MQTT | Message Queuing Telemetry Transport |
| NBR | Normas Técnicas Brasileiras |
| PAD | Pressão Arterial Diastólica |
| PAS | Pressão Arterial Sistólica |
| RAM | Random Access Memory |
| ReLU | Rectified Linear Unit |
| RNA | Rede Neural Artificial |

| | |
|------------------|---|
| RNN | Recurrent Neural Network |
| RMSE | Root Mean Square Error |
| SMA | Sistema Multiagente |
| SMACare | Sistema Multiagente para Cuidados em Saúde |
| SpO ₂ | Saturação de Oxigênio no Sangue |
| SSD | Solid State Drive |
| TCP/IP | Transmission Control Protocol / Internet Protocol |
| TPU | Tensor Processing Unit |
| Udesc | Universidade do Estado de Santa Catarina |
| UTC | Coordinated Universal Time |
| W&D | Wide & Deep Network |

LISTA DE SÍMBOLOS

| | |
|--------------------|---|
| $\%$ | Porcento |
| $^{\circ}\text{C}$ | Graus Celsius |
| t | Tempo ou instante de tempo |
| x | Variável de entrada ou vetor de características |
| y | Saída real (rótulo verdadeiro) |
| \hat{y} | Saída predita pelo modelo |
| μ | Média aritmética |
| σ | Desvio padrão |
| ϵ | Erro de predição |
| δ | Variação ou diferença entre valores |
| N | Número total de amostras ou participantes |
| n | Número de observações por grupo |
| Σ | Somatório |
| $\frac{x}{y}$ | Razão entre duas variáveis |
| RMSE | Raiz do Erro Quadrático Médio |
| MAPE | Erro Percentual Absoluto Médio |
| α | Taxa de aprendizado (learning rate) |
| λ | Parâmetro de regularização |
| ∇ | Gradiente |
| θ | Parâmetro do modelo (peso) |
| $f(x)$ | Função de ativação ou função de mapeamento |
| $\text{ReLU}(x)$ | Função de ativação ReLU: $\max(0, x)$ |
| \mathbb{R} | Conjunto dos números reais |
| \rightarrow | Implica ou converge para |
| \approx | Aproximadamente igual |
| \pm | Mais ou menos (média \pm desvio padrão) |
| \cdot | Produto escalar ou multiplicação |
| \in | Pertence ao conjunto |
| \forall | Para todo (quantificador universal) |
| \exists | Existe (quantificador existencial) |

SUMÁRIO

| | | |
|----------|--|-----------|
| 1 | INTRODUÇÃO | 19 |
| 1.1 | JUSTIFICATIVA E MOTIVAÇÃO | 20 |
| 1.2 | OBJETIVOS | 21 |
| 1.3 | ESTRUTURA DA TESE | 22 |
| 2 | ESTADO DA ARTE | 23 |
| 3 | SISTEMAS DISTRIBUÍDOS | 26 |
| 3.1 | COMPUTAÇÃO MÓVEL E UBÍQUA | 26 |
| 3.2 | INTERNET OF THINGS | 27 |
| 3.3 | PROTOCOLOS DE COMUNICAÇÃO | 28 |
| 3.4 | PROTOCOLOS DE DADOS | 29 |
| 3.4.1 | Advanced Message Queuing Protocol - AMQP | 29 |
| 3.4.2 | Constrained Application Protocol - CoAP | 30 |
| 3.4.3 | Message Queue Telemetry Transport - MQTT | 30 |
| 4 | SISTEMAS DE MEDIÇÃO | 33 |
| 4.1 | SENSORES DE SPO ₂ | 35 |
| 4.2 | SENSOR DE TEMPERATURA | 38 |
| 4.3 | SISTEMAS EMBARCADOS | 39 |
| 4.3.1 | Módulo Display Oled 128x64 | 43 |
| 4.3.2 | Módulo Mini MicroSD H8-003 | 43 |
| 4.3.3 | Módulo RTC DS3231 | 44 |
| 4.3.4 | Módulo Carregador de Bateria de Li-Ion TP4056 | 45 |
| 4.3.5 | Módulo ProtoCentral Pulse Express PC-MED-0411 | 46 |
| 5 | AGENTES INTELIGENTES E APRENDIZADO DE MÁQUINA | 48 |
| 5.1 | SISTEMA MULTIAGENTE | 49 |
| 5.2 | REDES NEURAIS ARTIFICIAIS | 49 |
| 5.2.1 | Arquitetura MultiLayer Perceptron | 50 |
| 5.2.2 | Treinamento e Otimização | 51 |
| 5.2.3 | Regularização: Batch Normalization e Dropout | 52 |
| 5.3 | MACHINE LEARNING NO SMA | 53 |
| 5.3.1 | TensorFlow | 53 |
| 5.3.2 | Keras | 55 |
| 5.3.3 | Scikit-Learn | 56 |
| 6 | DESENVOLVIMENTO DO SISTEMA SMACARE | 57 |
| 6.1 | HARDWARE | 57 |
| 6.2 | SISTEMA MULTIAGENTE | 59 |

| | | |
|--------------|--|------------|
| 6.2.1 | Estrutura do Banco de Dados e Integração com o Sistema Multiagente | 73 |
| 6.3 | REDES NEURAIS ARTIFICIAIS | 74 |
| 6.3.1 | Análise Comparativa dos Modelos para Integração ao SMA | 75 |
| 6.4 | ARQUITETURA DO SMACARE | 84 |
| 6.4.1 | Camadas e Transformações | 85 |
| 6.4.2 | Pré-Processamento de Dados | 85 |
| 6.4.3 | Treinamento do MLP | 86 |
| 6.5 | ARQUITETURA DE REDES NEURAIS PARA PRESSÃO ARTERIAL ESTIMADA | 88 |
| 6.6 | ARQUITETURA DE REDES NEURAIS PARA SATURAÇÃO PERIFÉRICA DE OXIGÊNIO | 91 |
| 6.7 | ARQUITETURA DA REDE NEURAL PARA CLASSIFICAÇÃO DA TEMPERATURA CORPORAL | 93 |
| 6.8 | ARQUITETURA DA REDE NEURAL PARA CLASSIFICAÇÃO DA FREQUÊNCIA CARDÍACA | 95 |
| 6.9 | ARQUITETURA DA REDE NEURAL PARA PROGNÓSTICO CLÍNICO | 97 |
| 6.10 | ARQUITETURA DA REDE NEURAL PARA DIAGNÓSTICO MULTILABEL | 102 |
| 6.11 | VALIDAÇÃO DA ACURÁCIA DO SMACARE EM SINAIS INDIVIDUAIS E SAÚDE AUTORREFERIDA | 103 |
| 6.12 | TESTE DE LATÊNCIA NA DETECÇÃO E NOTIFICAÇÃO DE ANOMALIAS | 107 |
| 6.13 | VALIDAÇÃO EXPERIMENTAL E ANÁLISE POR MATRIZ DE CONFUSÃO | 115 |
| 7 | CONSIDERAÇÕES FINAIS | 124 |
| 7.1 | DISCUSSÕES GERAIS | 124 |
| 7.2 | CONCLUSÃO | 129 |
| 7.3 | PRINCIPAIS CONTRIBUIÇÕES DA TESE | 129 |
| 7.4 | TRABALHOS FUTUROS | 130 |
| | REFERÊNCIAS | 132 |
| | APÊNDICES | 141 |
| | APÊNDICE A – PSEUDO-CÓDIGOS DE TREINAMENTO DAS RNAS | 142 |
| | APÊNDICE B – TERMOS TÉCNICOS DOS TREINAMENTOS DO APÊNDICE A | 145 |
| | APÊNDICE C – DIAGRAMAS DA PLACA DO DISPOSITIVO UTILIZADO NESSA TESE | 146 |
| | APÊNDICE D – DIAGRAMAS DA CASE DO DISPOSITIVO UTILIZADO NESSA TESE | 149 |

| | |
|--|------------|
| APÊNDICE E – QUESTIONÁRIO DE TRIAGEM | 153 |
| APÊNDICE F – TERMO DE CONSENTIMENTO LIVRE E ESCLAFECIDO | 164 |
| APÊNDICE G – ATESTADO DE UM DOS PARTICIPANTES DA PESQUISA | 167 |

1 INTRODUÇÃO

A integração de tecnologias de computação ubíqua, Internet das Coisas (IoT) e inteligência artificial (IA) tem promovido transformações na área da saúde, viabilizando novos paradigmas para o cuidado, o monitoramento remoto e o diagnóstico clínico (Coulouris et al., 2011; Perera et al., 2014). Em um contexto marcado pelo envelhecimento populacional e pelo aumento da prevalência de doenças crônicas, cresce a necessidade de soluções tecnológicas capazes de monitorar sinais vitais em tempo real, favorecendo a detecção precoce de anomalias e a otimização dos recursos médicos disponíveis (Mouha et al., 2021). No entanto, as soluções atualmente disponíveis apresentam limitações importantes: a maioria concentra-se apenas na coleta de dados, sem oferecer análise automática ou suporte à decisão clínica em tempo real; outras abordagens utilizam dados simulados em vez de sinais reais; e poucas contemplam a integração entre monitoramento contínuo, diagnóstico inteligente e geração de prognósticos personalizados. Também, as arquiteturas existentes costumam ser centralizadas, pouco escaláveis e desprovidas de mecanismos para registro histórico e notificações.

A IoT se caracteriza pela interconexão de dispositivos inteligentes que coletam, processam e compartilham dados continuamente, de forma autônoma (Dauer, 2021). No cenário da saúde digital, dispositivos vestíveis (*wearables*) têm se destacado por permitir medições constantes de parâmetros fisiológicos como frequência cardíaca, saturação periférica de oxigênio (SpO₂), temperatura corporal e pressão arterial estimada (Park et al., 2021; Kim et al., 2023). Equipados com sensores ópticos, iniciais e de temperatura, esses sistemas embarcados de baixo custo conseguem operar com alta eficiência energética, tornando-se adequados para monitoramento domiciliar e aplicações móveis (Kim et al., 2023). A conectividade desses dispositivos é viabilizada por protocolos leves, como o Bluetooth Low Energy (BLE), que permite comunicação com baixo consumo energético (Bluetooth SIG, 2014; Cho et al., 2015). Protocolos de mensagens como MQTT, AMQP e CoAP possibilitam arquiteturas assíncronas e desacopladas, essenciais para aplicações IoT que demandam escalabilidade, flexibilidade e confiabilidade na troca de dados (Mukherji et al., 2019; Hamdani; Sbeyti, 2019; Johansson; Dossot, 2020).

Neste contexto, a integração entre hardware vestível e sistemas multiagentes inteligentes desponta como uma solução promissora para superar as limitações existentes. Sistemas Multiagentes (SMA) oferecem flexibilidade e modularidade ao organizar componentes de software em agentes autônomos que colaboram para alcançar objetivos globais, como o diagnóstico em tempo real e a geração de prognósticos personalizados (Huhns; Stephens, 1999). Quando associados a técnicas de aprendizado de máquina, esses agentes tornam-se capazes de inferir estados clínicos a partir de séries temporais, identificar anomalias e emitir alertas imediatos, além de manter históricos para análises preditivas (Russell; Norvig, 2021).

Com base nessas necessidades e lacunas, a presente tese propõe o SMACare, um sistema multiagente que integra redes neurais artificiais a dados coletados por dispositivos vestíveis, com o objetivo de aprimorar a capacidade de diagnóstico em tempo real e a geração de prognósticos

clínicos personalizados. A arquitetura proposta foi validada experimentalmente com voluntários, demonstrando viabilidade para aplicações em *home care* e contribuindo para a descentralização e a personalização do cuidado em saúde.

1.1 JUSTIFICATIVA E MOTIVAÇÃO

Após a análise do estado da arte e da literatura correlata, foram identificadas lacunas significativas que motivaram o desenvolvimento do SMACare. Os principais desafios encontrados dizem respeito à dificuldade de acompanhar sinais vitais em tempo real fora do ambiente hospitalar; à limitação de soluções existentes, que na maioria das vezes apenas coletam sinais fisiológicos e os encaminham para análise humana, sem incorporar mecanismos automáticos para análise inteligente, diagnóstico em tempo real e prognóstico individualizado; e à carência de estudos que avaliem latência e confiabilidade desses sistemas, fatores críticos para evitar atrasos na detecção de eventos clínicos. Com o objetivo de superar essas limitações, o SMACare foi concebido para atuar tanto na detecção precoce de anomalias quanto na geração de prognósticos personalizados, combinando dados coletados em tempo real com análises baseadas no histórico clínico de cada participante. Para isso, o sistema opera em dois cenários complementares: no primeiro, realiza inferências contínuas em tempo real a partir de sinais adquiridos pelo dispositivo vestível; no segundo, utiliza os registros históricos armazenados para gerar diagnósticos e prognósticos de forma preditiva.

Grande parte das soluções atuais não conseguiu resolver adequadamente esses desafios por diferentes motivos. A maioria limita-se à coleta de dados, sem suporte à decisão em tempo real. Abordagens que utilizam inteligência artificial geralmente empregam dados simulados ou bases públicas, sem tratar da análise de dados provenientes de dispositivos vestíveis em tempo real, impossibilitando a emissão de alertas imediatos. Algumas pesquisas se concentram na previsão de eventos populacionais, como no contexto da COVID-19, mas não consideram o prognóstico individualizado a partir do histórico de cada paciente. De modo geral, tais sistemas não apresentam mecanismos de organização e armazenamento dos dados coletados, comprometendo a construção de um histórico clínico detalhado, e empregam arquiteturas estáticas e pouco escaláveis, que não contemplam cenários reais com múltiplos usuários e múltiplos dispositivos conectados simultaneamente.

O SMACare supera essas limitações ao propor uma arquitetura modular e escalável, que integra tecnologias complementares: protocolos IoT (BLE e MQTT) para coleta e transmissão de sinais vitais, Sistemas Multiagentes (JADE) para coordenação e processamento distribuído, e redes neurais artificiais profundas (TensorFlow) para inferência diagnóstica e prognóstica. Os sinais coletados pelo dispositivo vestível são armazenados em um banco de dados PostgreSQL, que organiza dinamicamente os registros por dispositivo e paciente, garantindo persistência, rastreabilidade e auditabilidade do histórico clínico. A partir desses dados, o sistema realiza inferências diagnósticas em tempo real e, periodicamente, gera prognósticos baseados no histó-

rico acumulado, notificando automaticamente o usuário sempre que anomalias são identificadas. Diferentemente de abordagens baseadas apenas em dados simulados, o SMACare foi validado experimentalmente com trinta voluntários utilizando um dispositivo vestível de baixo custo, desenvolvido nesta pesquisa com base no Arduino Nano BLE Sense Rev2 e sensores ópticos/infravermelho. A avaliação experimental incluiu a análise de tempos de resposta, confirmando baixa latência na inferência e na emissão de notificações.

Dessa forma, o SMACare responde à ausência de soluções integradas para diagnóstico automático e prognóstico em sistemas vestíveis, apresentando uma proposta inovadora que alia baixo custo (BoM), inteligência distribuída e validação com dados reais, superando limitações observadas em métodos anteriores.

1.2 OBJETIVOS

Este trabalho tem como objetivo geral desenvolver um Sistema Multiagente (SMA) para o diagnóstico em tempo real com base nos dados coletados por um dispositivo vestível destinado ao monitoramento de sinais vitais. Além dessa função, o SMA também deverá armazenar o histórico do participante da pesquisa em um banco de dados e gerar prognósticos a partir dessas informações.

Os objetivos específicos são:

- Analisar o estado da arte em sistemas multiagente, Internet das Coisas (IoT) e seus protocolos de comunicação;
- Projetar e desenvolver um sistema embarcado baseado em Arduino Nano BLE Sense Rev2, incorporando módulos para medição de frequência cardíaca, saturação periférica de oxigênio (SpO_2), temperatura corporal e pressão arterial estimada;
- Implementar e configurar protocolos de comunicação de dados (BLE, MQTT) para garantir conectividade eficiente e de baixo consumo energético entre o dispositivo vestível e os sistemas de processamento remoto;
- Desenvolver um Sistema Multiagente (SMA) utilizando o *framework* JADE para orquestração de agentes especializados em coleta, armazenamento, inferência e notificação de dados clínicos;
- Projetar, treinar e validar modelos de redes neurais artificiais para classificação em tempo real de sinais fisiológicos individuais (frequência cardíaca, SpO_2 , temperatura, pressão arterial) e integra-los ao SMA;
- Desenvolver e integrar ao SMA modelos de redes neurais para inferência prognóstica multiclasse com base em dados históricos dos sinais vitais dos participantes;

- Avaliar comparativamente o desempenho dos modelos propostos, considerando métricas de acurácia, tempo de inferência e confiabilidade para uso em cenários home-care;
- Validar experimentalmente o sistema completo com pacientes, analisando sua viabilidade para apoio ao diagnóstico em tempo real, diagnóstico multi-label e geração de prognósticos clínicos personalizados.

1.3 ESTRUTURA DA TESE

Considerando o Capítulo 1 como parte introdutória da tese, sua estruturada baseia-se em outros 6 capítulos. O Capítulo 2 realiza uma revisão de trabalhos correlatos, analisando soluções existentes em monitoramento remoto de saúde, dispositivos vestíveis e aplicações de IA para diagnóstico e prognóstico clínico. São comparadas abordagens evidenciando lacunas e motivando as decisões de projeto desta tese.

O Capítulo 3 aborda os fundamentos teóricos relacionados a sistemas distribuídos, computação ubíqua e IoT, incluindo os principais protocolos de comunicação utilizados em aplicações de saúde digital, como Bluetooth Low Energy (BLE) e MQTT. São discutidos conceitos de arquiteturas em nuvem, com ênfase em sua aplicabilidade para conectividade e escalabilidade em sistemas de saúde.

O Capítulo 4 descreve os sensores e módulos embarcados selecionados para o dispositivo vestível, detalhando o funcionamento de sensores ópticos para SpO₂ e frequência cardíaca, sensores de temperatura e métodos de estimativa de pressão arterial. São apresentados também os módulos de comunicação e a escolha do Arduino Nano BLE Sense Rev2 como plataforma de hardware, com justificativas técnicas e econômicas para sua adoção.

O Capítulo 5 apresenta os conceitos fundamentais de sistemas multiagentes (SMA) e aprendizado de máquina, abordando aspectos como modelos supervisionados, função de perda, regularização, normalização e arquiteturas de redes neurais do tipo *Multilayer Perceptron* (MLP). O capítulo discute a escolha do framework JADE para o desenvolvimento dos agentes e a integração com os modelos de IA, destacando estratégias de inferência em tempo real e organização modular do sistema.

O Capítulo 6 descreve o desenvolvimento e a arquitetura completa do sistema proposto, incluindo o projeto do hardware vestível, a configuração dos sensores, o desenvolvimento do Sistema Multiagente e o detalhamento das redes neurais artificiais implementadas para classificação dos sinais vitais e inferência prognóstica multiclasse. Este capítulo também discute o processo de coleta de dados com voluntários, os métodos de pré-processamento, o treinamento e a validação dos modelos propostos.

Sendo o último capítulo desta tese, no Capítulo 7 são apresentadas as discussões finais e a conclusão, também as contribuições do trabalho, além de sugestões para pesquisas futuras que possam dar continuidade ao desenvolvimento de soluções de monitoramento remoto em saúde baseadas em IoT e inteligência artificial.

2 ESTADO DA ARTE

Este capítulo apresenta a revisão do estado da arte realizada com o objetivo de fundamentar e justificar a proposta desta tese. O levantamento bibliográfico conduzido evidencia um crescente interesse da comunidade científica na aplicação de sistemas distribuídos inteligentes e técnicas de inteligência artificial voltadas para o monitoramento de saúde. Diversas abordagens recentes destacam-se nesse campo, cada uma contribuindo com soluções específicas, mas também apresentando limitações importantes que motivam a proposta desenvolvida nesta pesquisa.

Um dos desafios mais urgentes está relacionado ao monitoramento eficaz de pacientes idosos e portadores de doenças crônicas, especialmente em um cenário de envelhecimento populacional. O trabalho apresentado por (Humayun et al., 2022) propõe uma arquitetura baseada em sistemas multiagentes (SMA) para o monitoramento remoto de saúde, visando superar limitações dos sistemas tradicionais de *e-health*, que frequentemente se restringem à coleta de sinais vitais sem oferecer análise ou suporte à decisão clínica em tempo real. A estrutura proposta, denominada *Agent-Based Healthcare Monitoring Framework* (ABHCMF), é composta por agentes inteligentes distribuídos que colaboram para garantir a prestação eficiente e pontual de cuidados médicos a pacientes localizados em áreas urbanas ou remotas. O sistema integra sensores vestíveis para coleta contínua de dados fisiológicos, conectados a dispositivos móveis que comunicam essas informações a um Agente Médico Móvel Externo ao Hospital (MADO). Esse agente realiza um primeiro filtro, transmitindo apenas sinais vitais anormais ao Agente Médico no Hospital (DAH), reduzindo o volume de dados desnecessários. O DAH interpreta os dados recebidos à luz do histórico clínico do paciente e, em casos leves, pode emitir prescrições ou orientações diretamente; em situações mais críticas, encaminha os dados para o Agente Especialista (SA), que define condutas específicas, podendo incluir o envio de prescrições para o Agente Farmácia (PA) ou o acionamento de serviços de emergência. O Agente Enfermeiro (NA) atua no contexto hospitalar para gerenciar casos que necessitam de cuidado presencial, seguindo as orientações do DAH e do SA. A integração com redes 5G garante baixa latência na comunicação entre dispositivos móveis, sensores vestíveis e sistemas hospitalares, viabilizando o monitoramento em tempo real. O estudo inclui uma avaliação por meio de estudo de caso fictício com pacientes idosos portadores de doenças crônicas, demonstrando a capacidade do sistema de adaptar suas recomendações e acionar especialistas ou serviços de emergência de forma proativa.

O monitoramento contínuo de pacientes em ambientes clínicos dinâmicos também é essencial para intervenções oportunas e para a melhoria dos desfechos em saúde. Nesse sentido, o trabalho de (Shaik; Kumar; Kaur, 2023) propõe uma estrutura de monitoramento baseada em *Deep Reinforcement Learning* (DRL) multiagente para superar as limitações dos sistemas tradicionais, que frequentemente falham em lidar com a complexidade e a incerteza dos sinais vitais flutuantes. A proposta é composta por múltiplos agentes inteligentes, cada um especializado no acompanhamento de um parâmetro fisiológico específico, operando de forma cooperativa ou

competitiva dentro de um ambiente de monitoramento genérico. Cada agente DRL é modelado como uma política ótima que maximiza recompensas definidas a partir de pontuações de alerta clínico (*Modified Early Warning Scores* – MEWS). Esses agentes aprendem, por meio de interações sequenciais com o ambiente, a identificar padrões comportamentais do paciente e a emitir alertas direcionados aos *Medical Emergency Teams* (METs) em níveis graduados de criticidade. A formulação do problema adota o *Markov Decision Processes* (MDP), no qual o estado corresponde ao conjunto de leituras fisiológicas, as ações mapeiam os alertas para diferentes níveis MET, e as recompensas são calibradas para reforçar detecções precoces e precisas de deterioração clínica. A arquitetura multiagente inclui técnicas de *Deep Q-Networks* (DQN), com mecanismos de memorization e replay e estratégias de exploração balanceadas via algoritmo ϵ -greedy. Os agentes utilizam dados reais coletados de bases públicas como PPG-DaLiA e WESAD. Experimentos demonstraram que a abordagem multiagente DRL supera modelos como *Q-Learning*, PPO, A2C, Double DQN, DDPG, MADDPG e QMIX em métricas de recompensa, taxa de aprendizado, uso de memória e complexidade computacional, confirmando sua adequação para aplicações em cenários de saúde com restrições de recursos.

A pandemia de COVID-19 impôs desafios à saúde pública mundial e exigiu abordagens inovadoras para prever, simular e gerenciar sua evolução. (Haitam; Najat; Jaafar, 2023) apresenta uma estrutura integrada que combina aprendizado de máquina (ML) e SMA para modelar a dinâmica da pandemia no Marrocos, com o objetivo de apoiar decisões governamentais e políticas de saúde. Modelos como *Random Forests*, *Decision Trees* e *Support Vector Machines* foram treinados e validados para prever casos e óbitos diários de COVID-19, alcançando desempenho elevado, especialmente o *Random Forest*, que capturou padrões sazonais e identificou variáveis críticas. Em complemento, o SMA simula interações entre indivíduos, grupos sociais e o governo. Cada agente possui atributos específicos e toma decisões influenciadas por fatores sociais e econômicos. O agente governamental implementa e ajusta medidas de contenção (*lockdowns*, vacinação) em resposta aos indicadores de evolução da pandemia. As previsões geradas pelos modelos de ML alimentam as simulações multiagentes, criando um ciclo de realimentação que permite avaliar o impacto de diferentes políticas públicas. Os resultados demonstraram, por exemplo, que aumentar a cobertura vacinal em 10% poderia reduzir a mortalidade em até 25%, enquanto o relaxamento prematuro de restrições implica aumento significativo de casos e óbitos.

No campo das doenças neurodegenerativas, (Hou et al., 2025) propõe o ADAgent, um agente inteligente especializado para análise da Doença de Alzheimer, construído sobre modelos de linguagem de grande escala (*Large Language Models* – LLMs). O ADAgent integra raciocínio, planejamento e uso de ferramentas médicas especializadas para diagnóstico e prognóstico multimodais. A arquitetura integra um motor de raciocínio baseado em LLM para decompor consultas médicas em etapas sequenciais, um conjunto de ferramentas médicas especializadas capazes de processar entradas multimodais (MRI, PET) ou unimodais, e um coordenador colaborativo que agrupa e interpreta os resultados. A lógica segue um ciclo de quatro etapas: observação, pensamento, ação e coordenação colaborativa. O sistema foi testado com dados do *Alzheimer's*

Disease Neuroimaging Initiative (ADNI), incluindo MRI-T1 e PET-FDG, demonstrando ganhos de 2,7% em diagnóstico multimodal e 0,7% em prognóstico multimodal, além de robustez em cenários de modalidade ausente. A integração de múltiplos modelos (MedicalNet, nnMamba, ResNet50, MCAD, CMViM) em ferramentas especializadas permite ao ADAgent maior precisão e adaptabilidade, superando abordagens tradicionais.

Outra proposta relevante é apresentada por (Mutlag et al., 2021), que introduz o *Critical Healthcare Task Management* (CHTM), uma arquitetura distribuída baseada em *fog–cloud computing* e SMA para otimizar o gerenciamento de tarefas críticas em saúde, especialmente no monitoramento de ECG. O modelo estrutura-se em três níveis de processamento e dois de controle. Os agentes inteligentes realizam priorização dinâmica e alocação adaptativa de recursos, garantindo balanceamento de carga global e local, interoperabilidade entre nós de névoa e tomada de decisão em tempo real. Avaliações com dados reais de ECG utilizando *iFogSim* demonstraram reduções expressivas em métricas como uso de rede, tempo médio de resposta, latência e consumo de energia, reforçando a importância desse modelo para aplicações críticas.

O trabalho de Marcon e Colaboradores (2017) aborda os desafios do aumento da demanda por serviços de *Home Health Care* (HHC), propondo uma abordagem baseada em SMA e regras de decisão reativas para resolver o problema de roteamento de cuidadores de forma distribuída e dinâmica. A arquitetura integra um módulo off-line centralizado e um módulo on-line distribuído, no qual os agentes cuidadores tomam decisões locais e ajustam suas rotas em tempo real em resposta a eventos imprevisíveis, como atrasos ou mudanças de disponibilidade. Essa abordagem mostrou-se escalável e robusta, reduzindo os tempos de espera dos pacientes e melhorando a qualidade do atendimento.

Os estudos analisados demonstram avanços relevantes no uso de SMA e técnicas de inteligência artificial para a área da saúde. No entanto, nenhuma das abordagens existentes contempla de forma integrada todas as funcionalidades e requisitos que o SMACare busca atender, especialmente no que se refere à coleta em tempo real a partir de um *wearable*, análise distribuída entre agentes e geração combinada de diagnósticos e prognósticos. Esses trabalhos, entretanto, fornecem subsídios conceituais e metodológicos que fundamentam e justificam a proposta desta tese.

3 SISTEMAS DISTRIBUÍDOS

Este capítulo apresenta os fundamentos teóricos e conceituais que embasam o uso de sistemas distribuídos em aplicações de saúde digital, com ênfase na infraestrutura tecnológica necessária para viabilizar a coleta e transmissão de dados fisiológicos em tempo real. Também são descritos os principais conceitos relacionados à computação móvel e ubíqua, à Internet das Coisas (IoT) e à integração de dispositivos vestíveis com redes de sensores e protocolos de comunicação eficientes.

As seções subsequentes exploram, de forma sistemática, as características, limitações e aplicações de protocolos como *Bluetooth Low Energy* (BLE), Redes de Sensores Sem Fio (RSSF), AMQP, CoAP e MQTT, destacando sua adequação a ambientes de saúde distribuídos. Essa base teórica oferece subsídios técnicos essenciais para as decisões de projeto deste trabalho, assegurando uma integração consistente entre hardware vestível, comunicação de dados e processamento inteligente no SMA proposto.

3.1 COMPUTAÇÃO MÓVEL E UBÍQUA

A computação móvel e ubíqua têm sido impulsionadas pelos avanços tecnológicos na miniaturização de dispositivos e na interligação em rede sem fio, permitindo a crescente integração de equipamentos de computação pequenos e portáteis com sistemas distribuídos (Satyanarayanan, 2017). Esses equipamentos incluem notebooks, aparelhos portáteis, dispositivos acoplados ao corpo e aparelhos incorporados em outros equipamentos (Perera et al., 2014). A portabilidade de muitos desses dispositivos, junto à sua capacidade de se conectar convenientemente com redes em diferentes lugares, torna a computação móvel possível (Weiser; Brown, 2015) (Dourish; Bell, 2014).

A computação móvel refere-se à execução de tarefas de computação enquanto o usuário se desloca de um lugar a outro ou visita locais fora de seu ambiente usual. Usuários que estão longe de suas intranets "de base" (como a do trabalho ou de casa) podem acessar recursos através dos dispositivos que carregam consigo. Eles podem continuar a acessar a Internet, os recursos de sua intranet de base e, cada vez mais, utilizar recursos como impressoras ou pontos de venda que estão próximos, enquanto se movimentam (Coulouris et al., 2011).

A computação ubíqua, também conhecida como computação pervasiva, envolve o uso de vários dispositivos computacionais pequenos e baratos, presentes nos ambientes físicos dos usuários, incluindo suas casas, escritórios e até nas ruas. O termo "pervasivo" sugere que pequenos equipamentos de computação se tornarão tão integrados aos objetos diários que serão quase imperceptíveis, com seu comportamento computacional sendo transparente e intimamente vinculado à sua função física. Por sua vez, o termo "ubíquo" transmite a ideia de que o acesso a serviços de computação está sempre disponível, em qualquer lugar (Coulouris et al., 2011).

O usuário tem acesso a três formas diferentes de conexão sem fio. Seu notebook pode se conectar com a rede local sem fio da organização anfitriã, que cobre algumas centenas de

metros (por exemplo, um andar de um edifício) e se conecta ao restante da intranet através de um gateway ou ponto de acesso. O usuário também pode utilizar um telefone móvel que tem acesso à Internet, permitindo o acesso à Web e a outros serviços, limitado apenas pelo que pode ser apresentado em sua pequena tela. Por fim, o usuário pode portar uma câmera digital que se comunica por meio de uma rede sem fio pessoal (com alcance de cerca de 10 metros) com outros equipamentos, como uma impressora.

3.2 INTERNET OF THINGS

A Internet das Coisas (IoT, do inglês *Internet of Things*) representa um avanço na convergência entre o mundo físico e os sistemas computacionais, ao permitir a interconexão de dispositivos físicos inteligentes à internet. Essa integração envolve sensores, atuadores, software embarcado e conectividade de rede, criando um ecossistema digital capaz de coletar, processar e compartilhar dados de forma autônoma e contínua, sem intervenção humana direta (Coulouris et al., 2011)(Dauer, 2021)(Yasar; Gillis, 2024).

De acordo com Mouha et al. (2021), a IoT refere-se à interconexão, via rede internet, de dispositivos físicos equipados com componentes eletrônicos, sensores e atuadores, possibilitando que esses objetos sejam detectados, monitorados e controlados remotamente. Essa configuração é especialmente relevante no contexto da saúde digital, onde tecnologias como os dispositivos vestíveis (*wearables*) têm ganhado destaque por sua capacidade de realizar o monitoramento contínuo de sinais vitais em tempo real.

Dispositivos vestíveis consistem em tecnologias inteligentes posicionadas no corpo humano — como pulseiras, relógios, cintas torácicas e tecidos inteligentes — projetadas para coletar dados fisiológicos relevantes, tais como frequência cardíaca, pressão arterial, temperatura corporal, saturação de oxigênio e padrões de sono ou movimento. Esses dados são transmitidos a sistemas computacionais por meio de redes sem fio e posteriormente analisados, possibilitando diagnósticos mais precisos, acompanhamento remoto de pacientes e intervenções clínicas baseadas em evidências (Mouha et al., 2021) (Yasar; Gillis, 2024).

A estrutura funcional da IoT na saúde requer a implementação coordenada de três componentes fundamentais: *Hardware*, *Middleware* e Aplicação. O *Hardware* compreende os elementos físicos do sistema, como os sensores embutidos nos *wearables*, os atuadores e os sistemas embarcados que processam os dados localmente. O *Middleware* é responsável por viabilizar a comunicação entre os dispositivos, o armazenamento seguro das informações e o uso de recursos de computação em nuvem. A Aplicação envolve o desenvolvimento de interfaces amigáveis e adaptáveis às necessidades dos usuários, como os painéis interativos (dashboards), que facilitam a visualização dos dados de saúde por médicos e pacientes (Mouha et al., 2021) (Perera et al., 2014).

A adoção de *wearables* tem transformado o paradigma tradicional da saúde, promovendo um modelo mais preventivo, personalizado e centrado no paciente. Em ambientes hospitalares e

domiciliares, essas tecnologias contribuem para a redução de internações, detecção precoce de anomalias, otimização do uso de recursos médicos e melhoria da qualidade de vida, especialmente em populações com doenças crônicas ou em condição de risco elevado (Dourish; Bell, 2014)(Ho; Tsai; Lin, 2023). Para garantir a comunicação eficiente na IoT, é crucial o uso de protocolos de comunicação e de dados, projetados para minimizar o consumo de energia e adequar-se aos recursos limitados do *Hardware*.

3.3 PROTOCOLOS DE COMUNICAÇÃO

Dispositivos computacionais geralmente se conectam à rede e à internet por meio do protocolo IP (*Internet Protocol*). No entanto, esse protocolo exige um consumo significativo de energia e memória, características que não se adequam aos dispositivos IoT, que possuem restrições quanto à memória e ao consumo energético. Além disso, objetos IoT podem se comunicar localmente com outros objetos na mesma rede ou a curta distância, não necessitando necessariamente de uma conexão com a internet (Coulouris et al., 2011).

Diversas tecnologias de computação têm sido sugeridas e empregadas em ambientes IoT para atender a essas necessidades. Nas subseções seguintes, serão apresentados alguns dos protocolos não-IP mais utilizados.

O *Bluetooth Low Energy* (BLE) é uma extensão do protocolo Bluetooth tradicional (*Bluetooth Classic*), concebida para permitir comunicações sem fio com consumo energético reduzido, sem comprometer o alcance da transmissão. Introduzido pela especificação Bluetooth 4.0, o BLE foi projetado para aplicações em que a eficiência energética é crucial, especialmente no contexto da Internet das Coisas (IoT), onde sensores e dispositivos de monitoramento operam por longos períodos com alimentação limitada (Garcia-Espinosa et al., 2018).

Uma das principais características do BLE é sua arquitetura orientada a eventos e ciclos curtos de transmissão. Os dispositivos permanecem a maior parte do tempo em modo de inatividade (*idle* ou *sleep*), sendo ativados apenas para realizar conexões rápidas que envolvem a troca de pequenas quantidades de dados. Essa abordagem de comunicação intermitente permite que sensores e atuadores operem com melhor eficiência energética, tornando o BLE ideal para dispositivos vestíveis, monitoramento ambiental e automação residencial (Ho; Tsai; Lin, 2023).

Com o lançamento da versão 4.2 da especificação Bluetooth, o BLE passou a oferecer suporte ao protocolo IPv6 por meio do *Internet Protocol Support Profile* (IPSP), implementado sobre o protocolo 6LoWPAN. Essa inovação permitiu a integração nativa dos dispositivos BLE com redes baseadas em IP, ampliando sua interoperabilidade e facilitando sua adoção em arquiteturas IoT que utilizam protocolos de internet (Bluetooth SIG, 2014).

Em termos de arquitetura funcional, um dispositivo BLE pode operar em dois modos: *central* (mestre) ou *peripheral* (escravo). O dispositivo central tem a capacidade de gerenciar múltiplas conexões simultâneas com diversos periféricos, enquanto cada periférico está limitado a uma única conexão com um central. Essa estrutura mestre-escravo é particularmente eficaz em

sistemas de monitoramento distribuído, nos quais uma estação central coleta dados de vários sensores periféricos (Cho et al., 2015).

As operações do BLE podem ser divididas em duas fases principais: pré-conexão e pós-conexão. Durante a pré-conexão, o periférico anuncia sua presença em canais específicos de publicidade, aguardando uma requisição de conexão por parte de um dispositivo central. A central, por sua vez, realiza uma varredura ativa nesses canais para identificar os dispositivos disponíveis. Após o estabelecimento da conexão, inicia-se a fase de pós-conexão, na qual os dispositivos trocam dados por meio de eventos sincronizados. Durante os intervalos entre esses eventos, ambos os dispositivos retornam ao modo de baixo consumo, o que reforça ainda mais a economia de energia proporcionada pela tecnologia (Nair et al., 2015).

3.4 PROTOCOLOS DE DADOS

O principal protocolo de dados da internet é o *HyperText Transfer Protocol* (HTTP), porém, este não é adequado para ambientes com recursos limitados devido a sua verbose mais complexa, que acaba por incidir em um consumo maior de energia e vai de encontro com um dos requisitos base da IoT.

Muitos protocolos alternativos foram desenvolvidos visando ambientes IoT. Nesta seção, serão apresentados alguns deles.

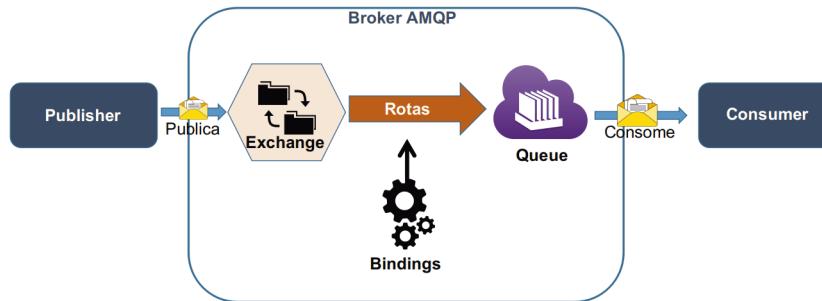
3.4.1 Advanced Message Queuing Protocol - AMQP

O Advanced Message Queuing Protocol (AMQP) é um protocolo baseado na comunicação *Machine-to-Machine* (M2M), desenvolvido por John O’Hara no JPMorgan Chase em Londres, Reino Unido, em 2003. Este protocolo foi projetado para suportar arquiteturas do tipo *request/response* e *publish/subscribe*, proporcionando um padrão robusto e flexível para publicação, enfileiramento, armazenamento e consumo de mensagens (Adiono et al., 2020). Os principais componentes do AMQP são *publishers*, *consumers*, *servers/brokers*, *exchanges*, *queues* e *bindings*. Cada um desses componentes desempenha um papel fundamental na operação do protocolo (Adiono et al., 2020)(Johansson; Dossot, 2020).

Um *publisher* é responsável por criar e enviar mensagens para as *exchanges*. As *exchanges*, por sua vez, são responsáveis por distribuir as mensagens para as *queues* (ou filas) com base em um conjunto de regras definidas pelos *bindings*. Essas regras determinam como as mensagens são roteadas dentro do sistema. Dependendo da configuração dos *bindings*, uma *exchange* pode decidir enfileirar ou descartar uma mensagem(Wytrębowicz; Cabaj; Krawiec, 2021).

As *queues* armazenam as mensagens até que sejam consumidas. Os *servers/brokers* gerenciam essas filas e facilitam a comunicação entre *publishers* e *consumers*. Os *consumers* acessam os *servers/brokers* para recuperar as mensagens das filas (Wytrębowicz; Cabaj; Krawiec, 2021). Este modelo assegura que uma mensagem nunca seja enfileirada sem passar primeiro pela *exchange*, que é a responsável por determinar o destino final da mensagem (Figura 1).

Figura 1 – Funcionamento do Protocolo AMQP



Fonte: Adaptado de (Johansson; Dossot, 2020)

Um *publisher* cria uma mensagem e envia esta para as *exchanges*. As *exchanges* distribuem cópias da mensagem em *queues* (ou filas) utilizando um conjunto de regras denominadas *bindings*. Após isso, os *servers/brokers* são acessados (ou assinados) por *consumers*, que retiram as mensagens da fila. Sendo assim, uma mensagem jamais é enfileirada sem antes passar pela *exchange*, que decide enfileirar ou descartar a mensagem, dependendo do tipo de *binding* configurado(Wytrębowicz; Cabaj; Krawiec, 2021).

3.4.2 Constrained Application Protocol - CoAP

CoAP é um protocolo *Machine-to-Machine* (M2M) para transferências web com o objetivo de ser leve e utilizável por dispositivos restritos em aplicações IoT. Utiliza o modelo cliente/servidor compreendendo as mesmas funcionalidades do HTTP, tornando a integração web mais simples, além de atender especificações mais complexas do M2M (Wytrębowicz; Cabaj; Krawiec, 2021).

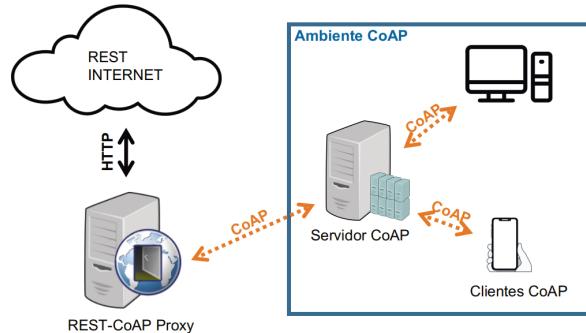
O modelo de troca de mensagens *request/response* baseia-se no UDP. Cada mensagem contém um ID (*MessageID*) usado para detectar duplicatas e para confiabilidade, sendo estes opcionais. Há ainda um campo opcional adicional denominado *Token*, que auxilia na detecção de duplicatas. O valor deste campo é definido de forma aleatória, portanto, é um número único para cada solicitação(Sarkar; Das; Jain, 2025).

Assim como o HTTP, o CoAP usa a arquitetura REST, onde os recursos são identificados pelo *Universal Resource Identifier* (URI) , por exemplo, `coap://`. Nós (ou nodos) HTTP podem não identificar nós CoAP, porém, um proxy intermediário pode disponibilizar um conjunto de recursos CoAP em URIs `http://` ou `https://` padrão, permitindo que clientes Web mais antigos acessem servidores CoAP de forma transparente (Figura 2) (Tariq et al., 2020).

3.4.3 Message Queue Telemetry Transport - MQTT

MQTT é um protocolo do tipo *publish/subscribe Machine-to-Machine* (M2M) que funciona sobre o protocolo TCP/IP e permite que dispositivos comuniquem-se uns com os outros através de redes wireless, por exemplo. Foi desenvolvido pela IBM no final dos anos 90, com

Figura 2 – Funcionamento do Protocolo CoAP



Fonte: Adaptado de (Islam; Lagutin; Fotiou, 2019)

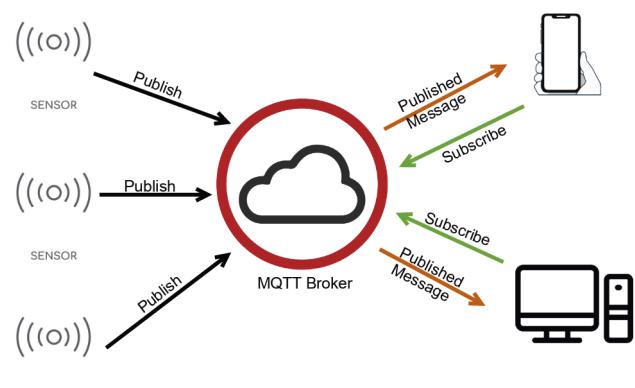
o objetivo de vincular sensores em *pipelines* de petróleo a satélites, suportando comunicação assíncrona entre as partes e desacoplando emissor e receptor. Sua implementação baseou-se no protocolo cliente/servidor, conectando redes e dispositivos com middlewares e aplicativos (Wytrębowicz; Cabaj; Krawiec, 2021).

O cliente (*publisher*) MQTT publica mensagens para um broker MQTT, estas podendo ser assinadas (*subscribed*) por outros *publishers* ou retidas, dependendo da configuração efetuada (Figura 3). Toda mensagem é publicada em um tópico, assim como é feito em um fórum da internet, por exemplo. Um cliente pode se inscrever em N tópicos e receber a mensagem de todos estes (Mukherji et al., 2019). Outra grande característica do MQTT são seus três níveis de Qualidade de Serviço (QoS) para entrega confiável de mensagens. Conforme Mukherji et al. (2019), são eles:

- a) QoS 0 (*At most once* ou Apenas uma Vez): A mensagem é enviada uma única vez, sem qualquer mecanismo de confirmação de recebimento. Assemelha-se ao protocolo de redes UDP;
- b) QoS 1 (*At least once* ou Pelo Menos uma Vez): A mensagem recebida é confirmada. Existe um armazenamento da mensagem por parte do emissor até sua posterior confirmação, possibilitando o reenvio da mensagem caso a confirmação não chegue.;
- c) QoS 2 (*Exactly once* ou Exatamente uma Vez): Nesse nível, a mensagem é enviada exatamente uma vez usando um *four way handshaking* semelhante ao do protocolo TCP.

Por ser um protocolo muito básico, o MQTT é mais adequado para redes com pequenos dispositivos que necessitem de monitoramento ou controle a partir de um servidor (Hamdani; Sbeyti, 2019).

Figura 3 – Funcionamento do Protocolo MQTT



Fonte: Adaptado de (Hamdani; Sbeyti, 2019)

4 SISTEMAS DE MEDIÇÃO

Este capítulo apresenta o detalhamento técnico dos sensores e módulos embarcados empregados no sistema de medição fisiológica, constituindo a base de hardware do dispositivo vestível proposto nesta tese. São descritos os sensores ópticos utilizados para a medição de SpO₂ e frequência cardíaca, o sensor infravermelho para a aferição de temperatura corporal, além dos principais componentes eletrônicos responsáveis pelo processamento local e armazenamento de dados. As seções subsequentes abordam a escolha da plataforma Arduino Nano BLE Sense Rev2, os módulos adicionais para expansão de funcionalidades e a integração dos sensores ao sistema embarcado. Esse embasamento técnico estabelece os requisitos essenciais para garantir a qualidade e a precisão na coleta dos sinais vitais, elementos fundamentais para o funcionamento adequado do Sistema Multiagente proposto nesse trabalho.

Sensores são dispositivos eletrônicos dotados de células sensíveis que medem parâmetros físicos específicos ou detectam variações no ambiente, fornecendo uma resposta correspondente (Biglari; Tang, 2023; Kumar, 2025). Esses dispositivos são capazes de transformar fenômenos físicos em sinais elétricos que podem ser interpretados por sistemas computacionais. Por exemplo, podem medir a intensidade da luz, detectar sons, registrar movimentos ou monitorar a temperatura de um ambiente ou de um ser vivo.

Os sensores desempenham um papel crucial em diversas aplicações tecnológicas ao fornecer dados essenciais para a tomada de decisões e controle de sistemas (Kumar, 2025). Exemplos de sensores incluem sensores de luz, utilizados para medir a intensidade da iluminação em um ambiente; sensores de temperatura, que medem a temperatura de um ambiente ou de um ser vivo e são amplamente utilizados em sistemas de controle climático, equipamentos médicos e processos industriais; sensores de som, que detectam variações na pressão sonora e são utilizados em sistemas de segurança, aparelhos auditivos e dispositivos de gravação de áudio; e sensores de movimento, que detectam movimentos em um determinado espaço e são essenciais para sistemas de segurança, dispositivos de monitoramento de atividade física e automação residencial.

Na Internet das Coisas (IoT), esses dispositivos são fundamentais para conectar objetos físicos à Internet, permitindo a coleta, análise e transmissão de dados em tempo real (Islam et al., 2022)(Yasar; Gillis, 2024). Eles são a base de diversas vertentes da IoT, tais como saúde inteligente, onde monitoram sinais vitais de pacientes e transmitem esses dados a profissionais de saúde para análise e intervenção imediata; cidades inteligentes, onde sensores ajudam a gerenciar recursos urbanos de maneira eficiente, como o controle do tráfego, iluminação pública, gestão de resíduos e monitoramento da qualidade do ar; agricultura inteligente, onde sensores monitoram condições do solo, umidade, temperatura e outros parâmetros ambientais para otimizar o cultivo, irrigação e proteção de plantas; sistemas de monitoramento de segurança e meio ambiente, onde sensores detectam intrusões, monitoram áreas de risco, detectam incêndios florestais e monitoram parâmetros ambientais para proteger a flora e fauna; e estacionamento inteligente, onde sensores identificam vagas disponíveis e direcionam os motoristas, reduzindo o tempo de

busca por estacionamento e otimizando o uso do espaço urbano (Islam et al., 2022)(Yasar; Gillis, 2024).

Os sensores na IoT são capazes de coletar vastas quantidades de dados a qualquer momento e de qualquer local. Esses dados são transmitidos através de redes IoT em tempo real, permitindo a análise e a tomada de decisões informadas. Atuadores são dispositivos responsáveis por executar ações em um sistema, frequentemente em resposta aos dados coletados pelos sensores. Eles podem ser dispositivos mecânicos, como motores e bombas hidráulicas, ou ações eletrônicas, como o envio de mensagens ou o controle de LEDs. Traduzem sinais elétricos em movimento físico ou outras formas de ação, permitindo a interação física com o ambiente(Dauer, 2021). Exemplos incluem motores e servomotores, utilizados para mover componentes em máquinas, robôs, veículos e outros sistemas automatizados; bombas hidráulicas, empregadas em sistemas que exigem movimentação de fluidos; controle de LEDs, que acendem ou apagam luzes em resposta a determinados estímulos; e envio de mensagens, que atuam enviando alertas ou notificações baseadas em dados coletados pelos sensores (Islam et al., 2022)(Yasar; Gillis, 2024).

Na IoT, esses dispositivos desempenham um papel vital ao utilizar os dados coletados para fornecer respostas inteligentes e automatizadas . São responsáveis por transformar as informações sensoriais em ações concretas, contribuindo para a automação e o controle eficiente de sistemas. Por exemplo, em um sistema de irrigação inteligente, os sensores de umidade do solo detectam níveis baixos de umidade, e os atuadores acionam bombas hidráulicas para irrigar a área(Abdelmoneim et al., 2025). Em um ambiente de saúde inteligente, os sensores detectam quedas de um paciente idoso, e os atuadores enviam alertas para os cuidadores ou acendem luzes de emergência (Reddy; Harivardhagini; Sreelakshmi, 2023). Em cidades inteligentes, a monitoração do tráfego pelos sensores permite que os atuadores ajustem os sinais de trânsito para otimizar o fluxo de veículos(Arshi; Mondal, 2023). Assim, a combinação de sensores e atuadores na IoT cria um sistema interconectado que melhora a eficiência, segurança e qualidade de vida em diversas áreas, desde o cuidado com a saúde até a gestão urbana e ambiental.

Tabela 1 – Alguns sensores utilizados na IoT.

| Sensor | Utilização |
|------------------------|---|
| Temperatura | Mede a quantidade de energia térmica. |
| Proximidade | Detecta a presença ou ausência de um objeto próximo, ou propriedades desse objeto. |
| Pressão | Detecta pressão e converte em sinais elétricos. |
| Químico | Indicam mudanças em um líquido ou mudanças químicas no ar. |
| SpO₂ | Indica a quantidade de oxigênio presente na hemoglobina do sangue, expressa como uma porcentagem. |
| IR | É usado para detectar certas características de seus arredores, emitindo ou detectando radiação infravermelha. |
| Movimento | Detecta movimento físico e converte em sinal elétrico. |
| Acelerômetro | É um transdutor que mede a aceleração física experimentada por um objeto devido às forças iniciais e converte o movimento mecânico em uma saída elétrica. |
| Giroscópio | Usado para medir a taxa angular ou velocidade angular. |
| Umidade | Detecta o vapor de água no ar ou em outros gases por meio da detecção de mudanças em correntes elétricas ou na temperatura do ar. |
| Óptico | Um sensor que mede a quantidade física de raios de luz e os converte em um sinal elétrico. |

Fonte: Elaborado pelo autor (2024).

4.1 SENSORES DE SPO₂

A saturação de oxigênio no sangue (SpO₂) é uma medida que indica a quantidade de oxigênio presente na hemoglobina do sangue, expressa como uma porcentagem. A hemoglobina pode estar na forma de hemoglobina oxigenada (HbO₂) ou desoxigenada (RHb), e a SpO₂ é a relação entre a quantidade de hemoglobina oxigenada e a quantidade total de hemoglobina no sangue (Elron et al., 2023).

Para medir a SpO₂, os sensores utilizam a técnica de oximetria de pulso, que envolve a emissão de luz por meio da pele e a detecção da luz transmitida ou refletida. A luz emitida passa através do tecido e é absorvida de forma diferente pela hemoglobina oxigenada e desoxigenada. Com base na quantidade de luz absorvida, o sensor pode determinar a proporção de hemoglobina oxigenada, calculando assim a saturação de oxigênio no sangue (Elron et al., 2023).

Sensores ópticos são projetados para capturar os sinais fotopletismográficos (PPG) gerados pela interação da luz com o tecido. Os sinais fotopletismográficos (PPG) são sinais ópticos gerados pela interação da luz com o tecido biológico, especialmente a pele. Esses sinais são utilizados em diversas aplicações médicas e de monitoramento de saúde para obter informações sobre a atividade cardiovascular e a oxigenação do sangue. Quando a luz emitida por um LED atravessa a pele e é absorvida pelos tecidos, parte dessa luz é refletida de volta para um fotodetector. A variação na quantidade de luz refletida está relacionada às mudanças no volume sanguíneo nos vasos sanguíneos próximos à superfície da pele. Consequentemente, as variações na intensidade da luz refletida capturadas pelo fotodetector geram o sinal fotopletismográfico (Park et al., 2021).

Os sinais PPG são compostos por uma componente contínua (DC) e uma componente

alternada (AC). A componente AC do sinal PPG está associada às variações no volume sanguíneo causadas pelo pulso arterial, enquanto a componente DC reflete a absorção de luz pelos tecidos circundantes. A análise desses sinais PPG pode fornecer informações valiosas sobre a frequência cardíaca, a variabilidade do pulso e a saturação de oxigênio no sangue, entre outros parâmetros fisiológicos. Esses sensores são calibrados e integrados com algoritmos para fornecer medições precisas, permitindo monitorar a oxigenação do sangue de forma não invasiva e em tempo real(Park et al., 2021).

O circuito integrado MAX32664 atua como um hub de sensores biométricos para medir vários indicadores de saúde, integrando algoritmos de medição biológica (Analog Devices, 2021). Em conjunto com os algoritmos da Maxim Integrated, ele oferece uma solução abrangente para medir a frequência cardíaca, os níveis de oxigênio no sangue, a qualidade do sono e a pressão arterial estimada em dispositivos vestíveis.

No contexto da estimativa da pressão arterial, (Kim; Cho, 2020) propõe um algoritmo onde os sinais de fotopletismografia (PPG) são analisados para obter informações relacionadas ao tempo de trânsito do pulso (PTT), formas de onda do pulso e outros parâmetros relevantes, utilizando apenas dois sensores de PPG, eliminando a necessidade de eletrocardiogramas. O uso de dois sensores de PPG permite o monitoramento contínuo das tendências da pressão sanguínea, oferecendo insights em tempo real sobre a saúde cardiovascular. A tecnologia de fotopletismografia (PPG) possibilita a medição de saturação de oxigênio, pressão arterial e débito cardíaco através de diversos mecanismos. Para a saturação de oxigênio, a PPG detecta mudanças no volume sanguíneo dentro dos leitos microvasculares dos tecidos. Quando a luz emitida por uma fonte, geralmente um LED, é absorvida pelo tecido no pulso, o fotodetector mede a quantidade de luz transmitida ou refletida. À medida que o sangue flui pelas artérias, ele absorve mais luz, diminuindo a quantidade que chega ao fotodetector. Isso gera uma onda pulsátil correspondente ao ciclo cardíaco. Ao analisar essas variações na absorção de luz, a PPG pode estimar os níveis de saturação de oxigênio no sangue (Kim; Cho, 2020).

Em relação à pressão arterial, a PPG pode estimá-la indiretamente analisando a onda pulsátil gerada pelas mudanças no volume sanguíneo durante o ciclo cardíaco. A amplitude e a forma da onda PPG fornecem informações sobre variações na pressão arterial. Embora a PPG sozinha não forneça medições diretas da pressão arterial, pode ser utilizada junto com técnicas de calibração para estimá-la de forma não invasiva (Kim et al., 2023).

A PPG também pode estimar o parâmetro que representa o volume de sangue bombeado pelo coração por minuto. Ao analisar as variações no volume sanguíneo detectadas pelo sensor PPG durante cada ciclo cardíaco, juntamente com as informações sobre a frequência cardíaca, algoritmos podem estimar o volume cardíaco. Essa estimativa oferece valiosos insights sobre a função cardiovascular e a saúde circulatória geral. No geral, a capacidade da tecnologia PPG de detectar mudanças no volume sanguíneo e analisar as ondas pulsáteis permite a estimativa de saturação de oxigênio, pressão arterial e volume cardíaco, tornando-a uma ferramenta versátil para monitoramento não invasivo de diversos parâmetros fisiológicos (Kim; Cho, 2020)(Kim et

al., 2023).

O funcionamento do algoritmo de estimativa de pressão arterial (BPT), descrito por (Kim; Cho, 2020), baseia-se em vários componentes fundamentais. Primeiramente, o Tempo de Trânsito do Pulso (PTT) é o tempo que uma onda de pulso leva para viajar entre dois pontos no sistema circulatório. Neste algoritmo, a diferença de tempo entre as ondas de pulso detectadas por dois sensores de PPG é medida, e o PTT é calculado com base nessas diferenças nos tempos de chegada das ondas de pulso nos dois sensores. Em seguida, a Pressão Arterial Média (MAP) é um parâmetro crucial na determinação da pressão sanguínea, calculada como a pressão média nas artérias durante um ciclo cardíaco. O algoritmo estima o MAP usando as informações obtidas dos sinais de PPG e a relação entre a frequência cardíaca (HR), o volume sistólico (SV) e a resistência vascular sistêmica (SVR) (Kim; Cho, 2020)(Analog Devices, 2021).

Para o cálculo do algoritmo, combinam-se os valores de PTT e MAP obtidos dos dois sensores de PPG, permitindo calcular os valores da pressão sanguínea. Consideram-se também valores específicos que representam as condições individuais dos vasos sanguíneos, como área de secção transversal, débito cardíaco e gravidade específica do sangue, para melhorar a precisão da estimativa. O algoritmo também visa reduzir erros na medição da pressão sanguínea utilizando a diferença de fase entre os sinais de MAP e PPG. Incorpora-se valores específicos individuais relacionados às condições dos vasos sanguíneos para melhorar a precisão da estimativa sem a necessidade de extensa entrada de dados do usuário (Kim; Cho, 2020)(Analog Devices, 2021).

O MAX30102 é um sensor integrado altamente utilizado em dispositivos vestíveis para monitoramento de parâmetros fisiológicos, especialmente a frequência cardíaca e os níveis de oxigênio no sangue. Este sensor é um módulo óptico que combina um fotodíodo, LEDs e circuitos analógicos para detectar as variações no fluxo sanguíneo e a absorção de luz pelas moléculas de oxigênio no sangue (Vohra; Chavan; Upadhyay, 2024).

O funcionamento do MAX30102 baseia-se na fotopletismografia (PPG), um método não invasivo que utiliza a luz para medir as mudanças no volume sanguíneo. O sensor possui dois LEDs (um vermelho e um infravermelho) que emitem luz através da pele e do tecido subjacente. O fotodíodo detecta a luz refletida ou transmitida, e as variações na intensidade da luz recebida correspondem às mudanças no fluxo sanguíneo. Como o sangue oxigenado e desoxigenado absorvem luz de forma diferente em comprimentos de onda específicos, o sensor pode calcular a oxigenação do sangue (SpO2) além da frequência cardíaca(Martínez; Pérez, 2021).

Uma das principais vantagens do MAX30102 é sua alta integração e precisão. Ele é projetado para consumo de energia ultra-baixo, o que é crucial para dispositivos vestíveis que dependem de baterias pequenas e precisam operar por longos períodos sem recarga frequente. Além disso, o sensor é capaz de compensar artefatos de movimento, melhorando a precisão das medições em condições de uso real, onde o usuário está se movimentando.

4.2 SENSOR DE TEMPERATURA

O sensor MLX90614 é um dispositivo avançado para a medição de temperatura, utilizando um termopar infravermelho altamente sensível e um processador de sinal integrado. Este sensor capta a radiação infravermelha emitida por objetos e a converte em um sinal elétrico, que é subsequentemente amplificado e transformado em um fluxo de bits por meio de um modulador Sigma-Delta. Este fluxo é processado por um processador digital de sinal (DSP), permitindo o cálculo preciso das temperaturas do objeto e do ambiente (Chai; Ghadziathanasiou; Black, 2021). A precisão do MLX90614 é destacada por sua capacidade de medir temperaturas com uma exatidão de $\pm 0,5$ °C em uma faixa típica de 0 °C a 50 °C. Esta alta precisão é alcançada graças ao uso de um amplificador de baixo ruído, um conversor analógico-digital (ADC) de 17 bits e um DSP integrado. O sensor oferece uma resolução de 0,02 °C, possibilitando medições detalhadas e precisas. Importante destacar que o sensor é calibrado de fábrica, assegurando alta precisão desde a instalação inicial, sem a necessidade de recalibração pelo usuário (Costanzo; Flores, 2020).

O MLX90614 incorpora um algoritmo interno sofisticado que realiza medições e cálculos necessários, fornecendo as temperaturas de saída por meio de interfaces PWM (modulação por largura de pulso) ou SMBus (System Management Bus). A interface PWM oferece um sinal digital facilmente interpretado por microcontroladores, enquanto a SMBus permite a comunicação eficiente com dispositivos inteligentes e sistemas de gerenciamento (Costanzo; Flores, 2020)(Chai; Ghadziathanasiou; Black, 2021). Operando em uma ampla faixa de temperatura, o MLX90614 suporta temperaturas ambientais de -40 °C a +85 °C e pode medir temperaturas de objetos de -70 °C a +380 °C. Esta ampla faixa de operação torna o sensor adequado para diversas aplicações, incluindo automação industrial, controle climático, dispositivos médicos e eletrônicos de consumo. Sua robustez é complementada pela capacidade de operar em ambientes adversos, onde medições de temperatura sem contato são essenciais(Costanzo; Flores, 2020)(Chai; Ghadziathanasiou; Black, 2021).

O sensor está disponível em várias versões, com diferentes campos de visão (FOV), permitindo adaptações para diversas necessidades de aplicação. Versões com FOV estreito são ideais para medir temperaturas de pequenos objetos ou áreas específicas, enquanto versões com FOV amplo são adequadas para monitoramento de áreas maiores. Um aspecto crítico do MLX90614 é sua capacidade de compensação de emissividade, ajustando as leituras para diferentes tipos de superfícies. A emissividade influencia a quantidade de radiação infravermelha emitida por uma superfície, e a compensação garante medições precisas independentemente da natureza da superfície medida (Sudianto et al., 2020)(Costanzo; Flores, 2020). A integração do MLX90614 a dispositivos vestíveis é facilitada por seu baixo consumo de energia e operação com baixa tensão. O sensor pode ser alimentado com tensões entre 3,6V e 5V e consome tipicamente menos de 1,5 mA, tornando-o ideal para aplicações em dispositivos alimentados por baterias e portáteis (Sudianto et al., 2020)(Costanzo; Flores, 2020).

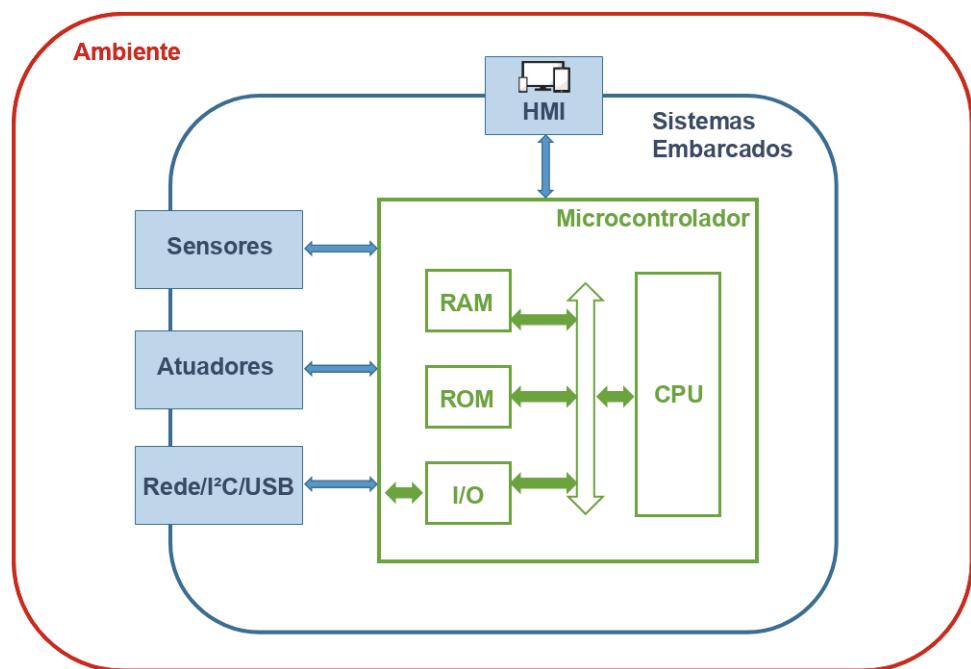
O sensor é amplamente utilizado para a medição de temperatura corporal devido à sua precisão, rapidez e capacidade de realizar medições sem contato. Este sensor infravermelho é especialmente adequado para aplicações médicas e de saúde, onde a temperatura corporal precisa ser monitorada de forma eficiente e higiênica (Costanzo; Flores, 2020).

4.3 SISTEMAS EMBARCADOS

A interação entre software e hardware está passando por significativas transformações, impulsionadas pela tendência crescente da denominação *inteligente* (ou *smart*). Para que um dispositivo seja classificado como *smart*, é essencial que ele incorpore sensores e atuadores (Kumar, 2025)(Mir; Jawad, 2025).

Os Sistemas Embarcados (Embedded Systems - EmbS) constituem uma parte fundamental dessa revolução tecnológica. Um Sistema Embarcado é essencialmente um hardware equipado com um microprocessador ou microcontrolador, projetado para executar tarefas específicas para as quais foi programado, como o controle de atuadores e a leitura de sensores, tanto local quanto remotamente. Esses sistemas são desenvolvidos para operar com capacidades limitadas de processamento, memória e armazenamento (Islam et al., 2022)(Biglari; Tang, 2023). Diferentemente de computadores e servidores tradicionais, os microcontroladores integram em um único chip RAM, ROM e interfaces de entrada/saída (I/O), necessários para interagir com sensores e atuadores e para se conectar com redes e outros dispositivos de comunicação. Essa integração é ilustrada na Figura 4 que apresenta um exemplo de arquitetura de Sistemas Embarcados (Kumar, 2025)(Mir; Jawad, 2025).

Figura 4 – Exemplo de Arquitetura de Sistemas Embarcados



Fonte: Adaptado de (Noergaard, 2018)

Um microcontrolador pode ser descrito como um microcomputador encapsulado em um único circuito integrado (CI), composto por uma unidade de processamento central (CPU) relativamente simples, combinada com dispositivos periféricos como interfaces de entrada/saída, memórias EEPROM e temporizadores. De acordo com (Noergaard, 2018), a essência de um microcontrolador reside em seu microprocessador. Microprocessadores modernos, equipados com interfaces de rede e software, permitem o acesso remoto a dados de sensores e a ativação remota de atuadores. No contexto da Internet das Coisas (IoT), as interfaces que conectam esses elementos são inspiradas e derivadas da tecnologia da informação, especialmente das tecnologias web.

Os microcontroladores mais simples operam com palavras de 8 bits e são adequados para aplicações que requerem pequenas quantidades de memória e funções lógicas simples, consumindo geralmente pouca energia. Dispositivos como nós de rede de sensores e dispositivos de vigilância, por exemplo, podem funcionar com uma pequena bateria por muitos anos (Noergaard, 2018). Ao contrário dos microprocessadores, que são projetados para processar dados complexos (como gráficos, vídeos e jogos), os microcontroladores são utilizados para tarefas mais simples, como controle de luminosidade, temperatura, temporização e leitura de sensores (Noergaard, 2018) (Domínguez-Bolaño et al., 2024) (Zormati; Lakhlef; Ouni, 2024).

Sistemas Embarcados modernos, que fazem uso extensivo de microcontroladores, são parte integrante das aplicações de IoT. Para simplificar o desenvolvimento e uso de microcontroladores e microprocessadores no contexto da IoT, diversas plataformas foram desenvolvidas, incluindo Arduino, ESP8266, ESP32 e Raspberry Pi (Domínguez-Bolaño et al., 2024) (Zormati; Lakhlef; Ouni, 2024). Essas plataformas oferecem um ambiente acessível e flexível para o desenvolvimento de aplicações IoT, promovendo a integração de hardware e software e facilitando a criação de dispositivos inteligentes.

O Arduino é uma plataforma de prototipagem eletrônica de código aberto baseada em hardware e software flexíveis e fáceis de usar. A placa Arduino contém um microcontrolador ATmega, que é essencial para a realização de diversas interações e operações com componentes externos (Barragán, 2004). Além do microcontrolador, a placa inclui diversos componentes necessários para facilitar a comunicação e a conectividade, como uma interface USB para comunicação com um computador, além de portas analógicas e digitais que permitem a conexão com uma ampla gama de dispositivos eletrônicos, incluindo motores, relés, sensores de luz, diodos laser, microfones, entre outros (Blum, 2019).

O desenvolvimento do Arduino teve como principal objetivo criar uma ferramenta acessível, funcional e de baixo custo, facilitando a programação e o uso por estudantes, hobbistas e desenvolvedores amadores. A acessibilidade do Arduino se deve não apenas ao seu custo relativamente baixo, mas também à simplicidade de seu design e à facilidade de uso de sua linguagem de programação, baseada em Wiring, uma linguagem derivada de C/C++ (Barragán, 2004) (Bluetooth SIG, 2014) (Prabowo; Irwanto, 2023).

Segundo Barragán (2004), a concepção do Arduino foi inspirada no Projeto Wiring,

desenvolvido durante sua tese de mestrado no Instituto Design de Interação de Ivrea (IDII). O Projeto Wiring foi concebido para simplificar a eletrônica envolvida na construção de circuitos complexos, abstraindo os detalhes técnicos frequentemente complicados e permitindo que os projetistas se concentrem nos objetivos do projeto como um todo. Esta abordagem inovadora teve como meta democratizar o acesso à tecnologia e possibilitar que mais pessoas, independentemente de seu nível de conhecimento técnico, possam criar e desenvolver seus próprios projetos eletrônicos.

O Arduino não apenas tornou a criação de dispositivos eletrônicos mais acessível, mas também inspirou uma geração de inovadores a explorar e experimentar com tecnologia de maneiras antes inimagináveis. A filosofia de código aberto do Arduino também incentivou a colaboração e o compartilhamento de conhecimentos, permitindo que a comunidade global contribua com melhorias e novas funcionalidades (Bluetooth SIG, 2014)(Prabowo; Irwanto, 2023).

O Arduino Nano BLE Sense Rev2 é uma plataforma de desenvolvimento que vem equipada com uma variedade de sensores integrados, possibilitando uma ampla gama de aplicações em dispositivos IoT e vestíveis (Kurniawan, 2021). Com um microcontrolador nRF52840 da Nordic Semiconductor, arquitetura ARM Cortex-M4 de 32 bits, oferece alto desempenho, baixo consumo de energia e suporte integrado para Bluetooth Low Energy (BLE). A capacidade de conectividade Bluetooth Low Energy permite a comunicação com smartphones, tablets e outros dispositivos compatíveis com BLE, facilitando a integração em aplicações vestíveis. A placa conta com o sensor BMI270, que mede a aceleração linear ao longo dos eixos X, Y e Z, com faixas selecionáveis de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$, e uma resolução de 16 bits por eixo, permitindo medições precisas de movimento e vibração (Kurniawan, 2021). Da mesma forma, o giroscópio mede a taxa de rotação nos mesmos três eixos, com faixas selecionáveis por software de $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$ e $\pm 2000^\circ/s$, e uma resolução de 16 bits por eixo, possibilitando um rastreamento preciso da orientação e dos movimentos angulares (Demrozi et al., 2020). O magnetômetro integrado detecta o campo magnético terrestre ao longo das três dimensões espaciais, com várias faixas de medição selecionáveis, incluindo $\pm 1,3$ gauss, $\pm 2,0$ gauss, $\pm 4,0$ gauss e até $\pm 8,0$ gauss, proporcionando resolução de 12 bits por eixo e uma taxa máxima de dados de 100 Hz (Demrozi et al., 2020)(Kurniawan, 2021).

Além dessas capacidades de detecção de movimento, está incluso um sensor de pressão com faixa de 300 a 1100 hPa e resolução de 0,12 hPa, permitindo a medição da pressão atmosférica, e um sensor combinado de temperatura e umidade, que pode detectar temperatura ambiente de $-40^\circ C$ a $85^\circ C$ com resolução de $0,01^\circ C$ e umidade relativa de 0 a 100% com resolução de 1,5% (Kurniawan, 2021).

Outro sensor integrado é o de proximidade e gestos, capaz de detectar a presença de objetos e reconhecer gestos simples, como aproximação, afastamento, deslize e toque, com uma faixa de detecção de até 10 cm. A plataforma também inclui um microfone capaz de capturar amostras de áudio de 16 bits e 16 kHz, fornecendo um meio para monitorar o som ambiente (Aras

et al., 2016)(Kurniawan, 2021). Estudos e projetos usando essa placa incluem trabalhos como os de (Kurniawan, 2021), que destacam a aplicação em reconhecimento de movimentos usando IA, e (Demrozi et al., 2020), que exploram o uso em dispositivos vestíveis para monitoramento de saúde e atividades físicas. Fontes adicionais de estudos que utilizaram a Arduino Nano BLE Sense Rev2 incluem a pesquisa sobre leitura eficiente de todos os sensores IMU na placa (Arduino Forum, 2023) e discussões sobre o uso da mesma para coleta e análise de dados de acelerômetro para várias atividades humanas (Arduino Forum, 2021).

Os módulos Arduino são conjuntos de componentes eletrônicos integrados em uma única placa, projetados para ampliar as capacidades do Arduino e simplificar a criação e implementação de projetos eletrônicos. Eles variam desde displays LCD e acelerômetros até drivers, amplificadores, conversores, e uma variedade de sensores e atuadores. Essa diversidade permite que os módulos sejam adaptados a diversas necessidades e aplicações. Entre os exemplos mais comuns de módulos Arduino estão o módulo RTC, que é basicamente um relógio indicando data e hora locais; módulo display, que exibe informações de tempo e data; o módulo acelerômetro, que mede movimentos e vibrações; o módulo Bluetooth, para comunicação sem fio; o módulo Wi-Fi, que permite a conexão com a internet; e o módulo relé, usado no controle de circuitos de potência. Cada um desses módulos oferece funcionalidades específicas que podem ser combinadas para criar sistemas eletrônicos complexos (Tsebesebe et al., 2025).

A aplicação de módulos Arduino em dispositivos vestíveis, conhecidos como wearables, tem transformado a forma como a tecnologia vestível é desenvolvida e utilizada. Os dispositivos eletrônicos vestíveis projetados para serem usados diretamente no corpo estão se tornando cada vez mais populares devido à sua capacidade de monitorar, analisar e interagir com o ambiente e com o usuário (Wan et al., 2018)(Tsebesebe et al., 2025). Essa tecnologia abrange desde simples pulseiras de monitoramento de fitness até avançados sistemas de vigilância da saúde .

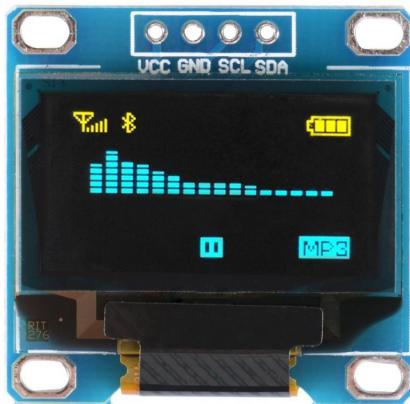
A flexibilidade do Arduino em integrar diversos sensores e módulos de comunicação permite o desenvolvimento de dispositivos que respondem em tempo real a estímulos externos (Cheng et al., 2021). No campo da saúde, por exemplo, wearables equipados com sensores como o MAX30102 podem monitorar sinais vitais, incluindo frequência cardíaca, respiração, pressão arterial e temperatura corporal. Esses dados são coletados e transmitidos para computadores ou smartphones, onde podem ser analisados, facilitando o acompanhamento contínuo e detalhado do estado de saúde do usuário (Cheng et al., 2021)(Ho; Tsai; Lin, 2023). Além do monitoramento de sinais vitais, a utilização de acelerômetros e giroscópios em wearables permite a detecção precisa de movimentos e quedas. Essa funcionalidade pode garantir a segurança de pacientes idosos ou pessoas com doenças crônicas, cujos movimentos podem ser monitorados remotamente. Em caso de uma queda, os dados coletados pelos sensores podem ativar alertas imediatos, permitindo uma resposta rápida e eficaz, que pode ser fundamental em situações de emergência. A integração com tecnologias de comunicação sem fio, como Bluetooth e Wi-Fi, permite a transmissão contínua de dados para aplicativos móveis e serviços na nuvem, facilitando o acompanhamento remoto da saúde do usuário por profissionais médicos, que podem acessar e analisar as informações sem

necessidade de uma visita física.

4.3.1 Módulo Display Oled 128x64

O módulo de display OLED (*Organic Light-Emitting Diode* - Figura 5) é um componente eletrônico avançado projetado para proporcionar uma interface visual de alta resolução e baixo consumo de energia. Esta tecnologia oferece contraste superior, tempos de resposta rápidos e maior durabilidade em comparação com os displays LCD tradicionais (Cain, 2021). Com uma resolução de 128x64 pixels, apresenta uma densidade de pixels que garante a exibição detalhada de gráficos e texto, sendo compatível com interfaces de comunicação como I2C e SPI, facilitando sua integração com microcontroladores e sistemas embarcados. Sua operação de baixo consumo de energia é particularmente vantajosa para dispositivos vestíveis alimentados por bateria, prolongando a vida útil do sistema sem comprometer o desempenho visual (Tsujimura, 2017).

Figura 5 – Módulo display Oled 128x64 nas cores de led amarelo e azul.



Fonte: (Huinfinito, 2025)

4.3.2 Módulo Mini MicroSD H8-003

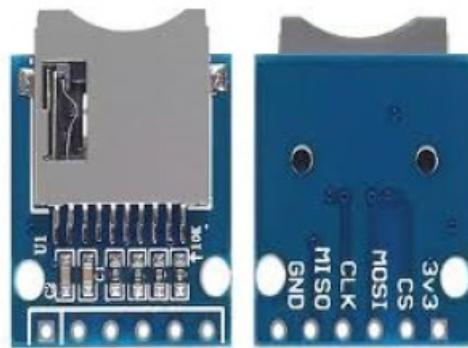
O módulo H8-003 (Figura 6) é um componente eletrônico compacto, projetado para expandir o armazenamento em sistemas embarcados e diversos dispositivos eletrônicos. É compatível com cartões MicroSD padrão, oferecendo uma solução eficiente e de baixo consumo de energia para leitura e gravação de dados. Pode ser utilizado em aplicações que exigem armazenamento externo de dados, como sistemas de monitoramento, dispositivos IoT (Internet das Coisas) como *wearables* e ainda em projetos de automação industrial. Sua interface de comunicação SPI padrão facilita a integração com microcontroladores e placas de desenvolvimento (Arduino, 2019).

O módulo suporta uma ampla faixa de temperaturas operacionais, tornando-o adequado para diversos ambientes. Sua construção compacta é ideal para sistemas com restrições de espaço

e sensíveis ao consumo de energia, garantindo o armazenamento seguro e estável de dados, essencial para aplicações críticas onde a integridade dos dados é crucial.

Exemplos de projetos nos quais o módulo pode ser empregado incluem armazenamento de dados de sensores, imagens e logs, registro de dados em tempo real, atualizações de firmware, compartilhamento de arquivos e execução de programas armazenados no cartão MicroSD. Suas vantagens incluem a fácil expansão de armazenamento, compatibilidade com diferentes tipos de cartões MicroSD e sistemas microcontroladores, além de ser acessível em termos de custo-benefício(Arduino, 2019).

Figura 6 – Vistas frontal e traseira do módulo H8-003.



Fonte: (Huinfinito, 2025)

4.3.3 Módulo RTC DS3231

O módulo de relógio de tempo real (RTC) DS3231 é um cronômetro extremamente preciso que utiliza um oscilador de cristal integrado. Ele foi projetado para fornecer informações exatas de tempo e data, incluindo segundos, minutos, horas, dia, data, mês e ano (Raza, 2018)(Pratama; Ariyanto; Abdurohman, 2017)(Alias; Mohyar, 2017)(Dutta et al., 2020).

Este dispositivo incorpora um oscilador de cristal compensado por temperatura (TCXO), o que significa que ele mantém uma alta precisão independentemente das variações de temperatura (Figura 7). O DS3231 também possui memória EEPROM e uma interface I2C, facilitando a comunicação com microcontroladores (Raza, 2018).

Outra característica do módulo RTC DS3231 é uma bateria de backup, permitindo a contagem do tempo mesmo quando a alimentação principal é interrompida. Isso o torna ideal para aplicações que exigem alta precisão e confiabilidade no controle de tempo, como sistemas de automação, dispositivos IoT e projetos de eletrônica em geral (Alias; Mohyar, 2017)(Dutta et al., 2020).

A utilidade do RTC citado foi demonstrada em diversas aplicações como sistemas de controle de ponto biométrico inteligente e seguro com capacidades de alerta GSM, onde seu papel é registrar a hora e data exatas da presença, prevenindo fraudes (Dutta et al., 2020) e em sistemas de sincronização de relógio digital usando protocolos IP e sem fio, proporcionando alta precisão temporal e baixa variação em relação ao relógio mestre (Pratama; Ariyanto; Abdurohman, 2017).

Figura 7 – Vista frontal do módulo DS3231.



Fonte: (Huinfinito, 2025)

4.3.4 Módulo Carregador de Bateria de Li-Ion TP4056

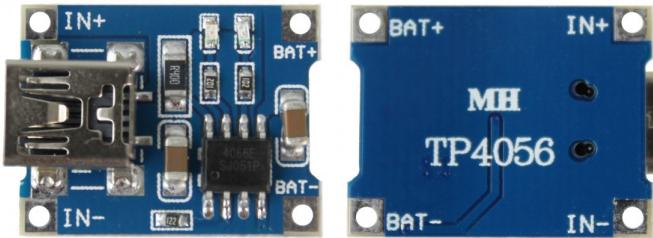
As baterias de íons de lítio tornaram-se a principal fonte de energia para dispositivos eletrônicos portáteis, desde smartphones e laptops até câmeras digitais e dispositivos IoT (Internet das Coisas) (Omariba; Zhang; Sun, 2018). Sua popularidade se deve à densidade de energia e baixas taxas de autodescarga, características essenciais para aplicações onde soluções de energia compactas e eficientes são importantes (Valsan; Emmanuel, 2017) . A reação eletroquímica entre o lítio e os íons presentes no eletrólito subjacente permite uma transferência de energia eficiente, possibilitando que essas baterias ofereçam capacidade de armazenamento superior em relação ao seu tamanho e peso (Zubi et al., 2018).

O módulo carregador de bateria de Li-Ion TP4056 (Figura 8) é projetado para facilitar o uso pelos desenvolvedores de circuitos e ocupar o menor espaço possível. Equipado com uma interface de carregamento micro-USB, o módulo permite uma conexão simples a fontes de energia comuns, como carregadores de celular e portas USB de computadores. Também inclui proteção contra sobrecarga, sobrecorrente e curto-círcuito, garantindo a segurança tanto da bateria quanto do dispositivo alimentado (Scrosati; Garche, 2010)(Croy; Abouimrane; Zhang, 2014)(Zubi et al., 2018).

Uma das principais vantagens deste módulo é sua capacidade de carregamento de maneira controlada, ajustando a corrente conforme necessário para proteger a bateria. Isso o torna adequado para uma variedade de aplicações, incluindo projetos de eletrônica DIY, dispositivos vestíveis e sistemas de energia de backup. Ainda, possui dois indicadores LED que fornecem feedback visual sobre o status do carregamento, com um LED vermelho indicando que a bateria

está sendo carregada e um LED verde que se acende quando o carregamento está completo (Scrosati; Garche, 2010).

Figura 8 – Vistas frontal e traseira do módulo TP4056.



Fonte: (Huinfinito, 2025)

4.3.5 Módulo ProtoCentral Pulse Express PC-MED-0411

O sensor ProtoCentral Pulse Express SpO₂, Frequência Cardíaca e Pressão Sanguínea Estimada (BPT) é uma placa compacta projetada para medir sinais vitais em aplicações de saúde vestíveis. Ele combina o sensor óptico MAX30102 e o hub de sensor biométrico MAX32664D, permitindo medições precisas de oximetria de pulso (SpO₂), frequência cardíaca e de pressão sanguínea estimada (Figura 9). Este dispositivo possui sensores ópticos de alta sensibilidade para medições precisas de SpO₂ e frequência cardíaca, enquanto o hub de sensor biométrico realiza cálculos avançados sem a necessidade de desenvolver algoritmos adicionais. Projetado com baixo consumo de energia, é ideal para dispositivos vestíveis e aplicações onde eficiência energética é crucial (ProtoCentral, 2024).

A placa se conecta facilmente a qualquer microcontrolador host via interface padrão I2C, facilitando a integração. Ela é adequada para uma variedade de aplicações, incluindo dispositivos como smartwatches, monitores de atividade física, pesquisa e desenvolvimento para prototipagem rápida, e pesquisas médicas para estudar condições fisiológicas. O sensor inclui uma biblioteca Arduino para facilitar a integração e documentação abrangente que cobre configuração, uso e exemplos de software (ProtoCentral, 2024). O módulo é código aberto sob a licença CERN-OHL-P v2 para hardware e Licença MIT para software (ProtoCentral, 2024).

Figura 9 – Vista frontal do módulo PC-MED-0411.



Fonte: (Huinfinito, 2025)

5 AGENTES INTELIGENTES E APRENDIZADO DE MÁQUINA

Este capítulo apresenta os fundamentos teóricos e metodológicos que sustentam o desenvolvimento do SMA proposto nesta tese, com foco em agentes inteligentes e aprendizado de máquina. São explorados os conceitos essenciais relacionados a sistemas multiagente, incluindo suas características de autonomia, modularidade e cooperação, além dos princípios das redes neurais artificiais empregados para classificação de sinais fisiológicos e geração de prognósticos clínicos. As seções seguintes abordam de forma sistemática a arquitetura de redes neurais Multi-layer Perceptron (MLP), os processos de treinamento supervisionado, otimização e regularização, além da utilização de *frameworks* como *TensorFlow*, *Keras* e *Scikit-Learn*.

O estudo sobre agentes tem sido um tópico de interesse em pesquisas recentes por apresentar uma estruturação para a construção de sistemas proativos de fácil extensão, quando comparados com as tecnologias atualmente disponíveis para o desenvolvimento de sistemas, devido às suas características de decomposição, abstração e organização (Jennings et al., 2001). Algumas áreas dessas pesquisas incluem arquiteturas orientadas a objeto, arquiteturas de objetos distribuídos, engenharia de software, sistemas de aprendizado adaptativos, inteligência artificial, sistemas especialistas, algoritmos genéticos, processamento distribuído, algoritmos distribuídos, segurança, internet das coisas, entre outros.

Do ponto de vista arquitetônico, agentes são entidades compostas por um identificador único e três componentes principais: código, dados e estado, além de possuírem um ciclo de vida associado ao seu estado de execução (Maes, 1995). Esses componentes são mantidos pelo mobile agent system na plataforma de agentes, tendo o agent middleware como software que implementa a plataforma de agentes e a administra.

No contexto da literatura, o termo *agente* não possui uma definição unanimemente aceita; em parte, devido à multiplicidade de enfoques sob os quais é estudado. Algumas das definições fornecidas são:

Agentes inteligentes são entidades-software que manipulam conjuntos de operações em benefício de um usuário ou de outro programa com certo grau de independência ou autonomia, e de alguma forma, empregam conhecimento ou representação das metas e anseios do usuário. (Grosof et al., 1995).

Agentes Inteligentes realizam continuamente três funções: percepção de condições dinâmicas no ambiente; ação para afetar condições no ambiente; e raciocínio para interpretar percepções, solucionar problemas, redigir inferências e determinar ações. (Hayes-Roth, 1995).

Partindo dessas definições, Franklin e Graesser (1996) sugerem um novo conceito do termo *agente*:

Um agente atua por concessão de um indivíduo ou de outro agente (Grosof et al., 1995), dependendo crucialmente do que se considera um ambiente e do que significam os termos *sentir* e *agir*, onde deve-se restringir, pelo menos, algumas noções referentes a esta dependência, possuindo também raciocínio durante o processo de selecionar ações (Hayes-Roth, 1995).

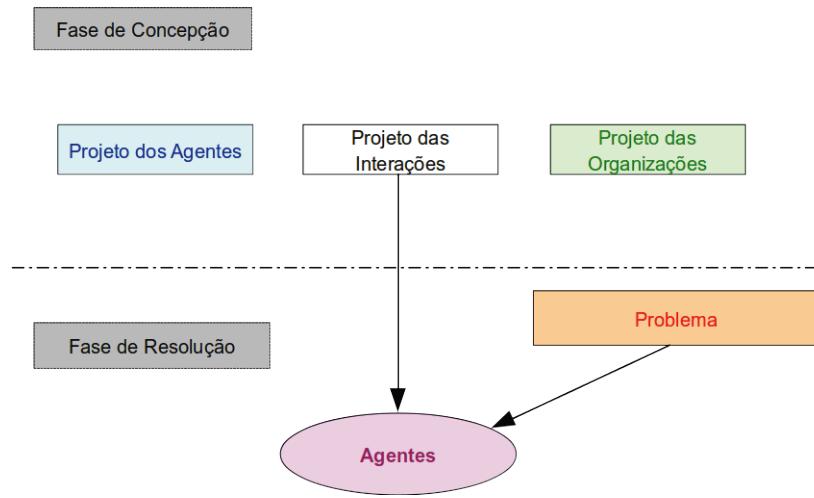
5.1 SISTEMA MULTIAGENTE

Um Sistema Multiagente (SMA) consiste em uma aplicação distribuída, composta por um conjunto de processos autônomos, heterogêneos, distribuídos e inteligentes, que cooperam entre si para a solução de um problema complexo que está além das suas capacidades individuais (Sajja, 2008).

O SMA busca definir formas de coordenação dos seus conhecimentos, objetivos, habilidades e planos, de forma que, conjuntamente, os processos possam realizar ações ou resolver problemas (Bond; Gasser, 2014). Para tanto, a missão dos agentes pode ser centralizada – na qual todos os agentes trabalham em direção a um único objetivo global – ou distribuída – na qual há vários objetivos individuais, podendo haver interação entre eles. A Figura 10 ilustra esse cenário.

Quanto à mobilidade, os agentes migram entre os hosts do SMA para utilizar os recursos destes hosts remotos de forma local. Nesse contexto, em um SMA, cada máquina é mapeada utilizando o conceito de localização, ou seja, possui uma identificação única de um lugar no espaço de endereçamento distribuído.

Figura 10 – Estrutura de um SMA.



Fonte: Elaborado pelo autor (2025).

5.2 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais constituem um modelo computacional inspirado no funcionamento dos neurônios biológicos, sendo amplamente utilizadas para resolver problemas de classificação, regressão e previsão em dados complexos e de alta dimensionalidade. Em um Sistema Multiagente para aplicações em saúde, essas redes são fundamentais para processar sinais fisiológicos capturados por sensores vestíveis e para realizar inferência diagnóstica e prognóstica em tempo real. A seguir, são apresentados os principais conceitos empregados no

desenvolvimento das redes neurais desta pesquisa, com foco na arquitetura MLP, nos métodos de treinamento e nos mecanismos de regularização adotados para garantir generalização e robustez.

5.2.1 Arquitetura MultiLayer Perceptron

A arquitetura Multilayer Perceptron (MLP) constitui um modelo clássico de rede neural *feedforward* amplamente utilizado para tarefas de classificação e regressão, caracterizando-se por sua estrutura em camadas densamente conectadas. Cada MLP inclui uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, em que os neurônios artificiais realizam transformações não lineares por meio de funções de ativação, possibilitando a modelagem de relações complexas entre variáveis de entrada e saída (Goodfellow; Bengio; Courville, 2016).

Em uma MLP, cada neurônio em uma camada está conectado a todos os neurônios da camada subsequente por pesos sinápticos treináveis. Esses pesos são ajustados durante o treinamento para minimizar uma função de perda que mede a discrepância entre as previsões da rede e os rótulos reais. O processamento em cada camada ocorre por meio de operações lineares seguidas da aplicação de funções de ativação como a ReLU (*Rectified Linear Unit*), amplamente empregada por sua capacidade de reduzir o problema do gradiente desaparecendo e acelerar a convergência (Nair; Hinton, 2010).

O processo de aprendizado de uma MLP utiliza o algoritmo de retropropagação do erro, que calcula gradientes parciais da função de perda em relação a cada peso da rede e os atualiza no sentido de minimizar esse erro global (Rumelhart; Hinton; Williams, 1986). Para otimização, esta pesquisa empregou o método Adam (*Adaptive Moment Estimation*), conhecido por combinar momentos adaptativos com técnicas de aceleração do gradiente, proporcionando convergência mais estável em cenários de dados heterogêneos (Kingma; Ba, 2015).

No contexto desta tese, arquiteturas MLP foram aplicadas para classificar parâmetros fisiológicos como frequência cardíaca, saturação periférica de oxigênio (SpO_2), temperatura corporal e pressão arterial estimada, assim como para inferência prognóstica baseada em séries temporais. Para cada tarefa, foram projetadas arquiteturas específicas, com número de camadas e neurônios ajustados empiricamente para equilibrar capacidade de modelagem e risco de sobreajuste. Configurações típicas incluíram entre duas e quatro camadas ocultas, com tamanhos progressivamente decrescentes (por exemplo, 32, 16, 8 neurônios), incentivando a extração hierárquica de características latentes (Goodfellow; Bengio; Courville, 2016). Visando mitigar problemas de instabilidade no treinamento e sobreajuste, foram incorporadas técnicas como *Batch Normalization* e *Dropout*. A normalização em *mini-batches* padroniza as ativações em cada camada, reduzindo o *internal covariate shift* e acelerando o treinamento (Ioffe; Szegedy, 2015). O *Dropout* consiste em desativar aleatoriamente frações de unidades durante o treinamento, prevenindo coadaptações excessivas e promovendo generalização (Srivastava et al., 2014).

5.2.2 Treinamento e Otimização

O treinamento de redes neurais artificiais, incluindo as arquiteturas MLP utilizadas neste trabalho, baseia-se no ajuste dos parâmetros sinápticos de forma a minimizar uma função de perda que quantifica a discrepância entre as saídas previstas pela rede e os valores-alvo. O método clássico empregado para esse ajuste é o algoritmo de *retropropagação do erro* (Rumelhart; Hinton; Williams, 1986), que aplica a regra da cadeia para calcular de maneira eficiente os gradientes parciais em relação aos pesos de cada camada. Essa técnica permite o treinamento de redes profundas ao propagar as informações de erro do nível de saída para as camadas anteriores, ajustando os pesos no sentido de reduzir o erro total.

Em cada iteração de treinamento, o modelo realiza uma *passagem direta* (*forward pass*), na qual os dados de entrada atravessam as camadas da rede até produzir uma predição, e uma *passagem reversa* (*backward pass*), em que os gradientes da função de perda são calculados e os pesos são atualizados. A atualização ocorre por meio de algoritmos de otimização baseados em gradiente, cujo objetivo é encontrar um mínimo local da superfície de erro. A escolha do otimizador influencia significativamente a velocidade de convergência e a estabilidade do treinamento.

Nesta pesquisa, empregou-se o otimizador Adam (*Adaptive Moment Estimation*) (Kingma; Ba, 2015), uma técnica que combina as vantagens do método Momentum com a adaptação de taxas de aprendizado individuais para cada parâmetro. O Adam armazena estimativas dos momentos de primeira ordem (médias dos gradientes) e de segunda ordem (médias dos quadrados dos gradientes), corrigindo o viés inicial por meio de termos de amortecimento. Essa abordagem se mostrou particularmente eficaz para o treinamento de modelos com dados biomédicos, que podem apresentar alta variabilidade e padrões heterogêneos, garantindo convergência estável mesmo em cenários com ruído ou desbalanceamento entre classes. Outro aspecto central do treinamento das MLPs foi a escolha das funções de ativação. As camadas ocultas utilizaram majoritariamente a função ReLU (*Rectified Linear Unit*), definida como demonstrado na Equação 1 por sua simplicidade computacional e pela capacidade de mitigar o problema do *vanishing gradient*, que afeta funções como Sigmoid ou Tanh em redes profundas (Nair; Hinton, 2010; Goodfellow; Bengio; Courville, 2016). A ReLU introduz esparsidade nas ativações, incentivando representações mais eficientes e facilitando o aprendizado de padrões relevantes mesmo com dados ruidosos provenientes de sensores vestíveis.

$$f(x) = \max(0, x) \tag{1}$$

Para problemas de classificação multiclasse, a camada de saída foi projetada com a função Softmax, transformando o vetor de logits em uma distribuição de probabilidade sobre as classes possíveis. A função de perda utilizada foi a *entropia cruzada categórica*, que mede a divergência entre a distribuição prevista pela rede e a distribuição alvo. Esse critério fornece gradientes bem

comportados para problemas de classificação, melhorando a capacidade do modelo de distinguir estados clínicos com alta precisão.

O treinamento foi realizado em *mini-batches*, técnica que combina as vantagens do gradiente estocástico com a eficiência do processamento vetorial. Essa estratégia reduz a variância das atualizações de gradiente, contribuindo para trajetórias de convergência mais suaves e para a capacidade de generalização do modelo. Combinando essas práticas — retropropagação eficiente, otimização adaptativa, funções de ativação não lineares e atualização em *mini-batches* — o sistema proposto alcançou desempenho competitivo em tarefas de diagnóstico e prognóstico em tempo real, mesmo em ambientes clínicos domiciliares com recursos computacionais limitados.

5.2.3 Regularização: Batch Normalization e Dropout

As redes neurais profundas estão sujeitas ao problema de sobreajuste (*overfitting*), especialmente em cenários com dados limitados ou ruidosos, como ocorre em sinais fisiológicos coletados por sensores vestíveis. Para mitigar esse risco e melhorar a capacidade de generalização dos modelos, foram incorporadas técnicas consagradas de regularização: *Batch Normalization* e *Dropout*. A *Batch Normalization* foi introduzida para reduzir o chamado *internal covariate shift*, que ocorre quando as distribuições das ativações internas mudam durante o treinamento, dificultando o aprendizado eficiente. A técnica normaliza as ativações de cada camada para que tenham média zero e variância unitária em cada mini-batch, estabilizando o fluxo de gradientes e permitindo o uso de taxas de aprendizado mais elevadas (Ioffe; Szegedy, 2015). Especificamente, dado um conjunto de ativações x em um *mini-batch*, a normalização é realizada segundo a Equação 2:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (2)$$

onde μ e σ^2 são a média e a variância do *mini-batch*, e ϵ é um termo de estabilidade numérica. Em seguida, são aplicados parâmetros aprendíveis de escala e deslocamento para recuperar a expressividade da camada, demonstrado na Equação 3:

$$y = \gamma\hat{x} + \beta \quad (3)$$

Essa técnica não apenas acelera a convergência, como também atua como uma forma de regularização adicional, reduzindo a sensibilidade a inicializações inadequadas dos pesos e a flutuações nos dados de entrada. No presente trabalho, *Batch Normalization* foi incorporada após camadas densas em arquiteturas MLP, garantindo estabilidade durante o treinamento mesmo com sinais fisiológicos ruidosos e heterogêneos.

A técnica de *Dropout* consiste em desligar aleatoriamente uma fração das unidades em cada camada durante o treinamento, prevenindo coadaptações excessivas entre neurônios

(Srivastava et al., 2014). Em cada iteração, neurônios são *desativados* com probabilidade p , forçando a rede a aprender representações redundantes e robustas. Essa estratégia se mostrou especialmente eficaz para cenários com número limitado de amostras ou alto grau de ruído, como é característico de sinais biomédicos coletados em ambiente domiciliar. A implementação de *Dropout* neste trabalho utilizou taxas ajustadas empiricamente conforme a tarefa, com valores típicos entre 0.2 e 0.5 para camadas ocultas. Durante a inferência, o *Dropout* é desativado, e os pesos são escalados para compensar a ausência de unidades durante o treinamento. Essa abordagem reduz o risco de sobreajuste sem penalizar a capacidade expressiva do modelo.

5.3 MACHINE LEARNING NO SMA

O aprendizado de máquina supervisionado com redes neurais profundas consiste em treinar modelos a partir de exemplos rotulados, de modo que aprendam a mapear características de entrada para saídas desejadas. Redes profundas, como as arquiteturas *Multilayer Perceptron* utilizadas neste trabalho, são compostas por múltiplas camadas de transformação não linear, permitindo capturar relações complexas e hierárquicas nos dados fisiológicos coletados por sensores vestíveis. No contexto do SMA proposto, o aprendizado supervisionado é empregado para classificar estados clínicos em tempo real e realizar prognósticos personalizados, com modelos treinados para reconhecer padrões sutis mesmo em sinais ruidosos ou heterogêneos.

Para viabilizar a implementação prática, foram utilizadas bibliotecas modernas de aprendizado de máquina: *TensorFlow* oferece uma infraestrutura escalável e eficiente para construção e treinamento de redes neurais profundas; *Keras* provê uma interface de alto nível que simplifica a definição de modelos, camadas e fluxos de treinamento, tornando o desenvolvimento mais acessível e reproduzível; e *Scikit-learn* foi empregado para tarefas auxiliares como pré-processamento, avaliação de métricas e experimentação de modelos baseados em aprendizado de máquina clássico. Em conjunto, essas ferramentas permitiram a construção de um pipeline completo, modular e flexível para processamento e análise automatizada de sinais vitais no ambiente domiciliar.

5.3.1 TensorFlow

O *TensorFlow* é uma biblioteca de código aberto desenvolvida pelo Google para computação numérica baseada em grafos de fluxo de dados (*dataflow graphs*), projetada para atender a aplicações de aprendizado de máquina em grande escala. Sua flexibilidade e capacidade de paralelização permitem que o treinamento de modelos ocorra de forma eficiente em CPUs, GPUs e TPUs, além de ser compatível com dispositivos móveis e ambientes distribuídos (Abrahams; Smith; Zhang, 2016). A estrutura do *TensorFlow* facilita a definição de operações matemáticas complexas por meio de grafos computacionais, nos quais os nós representam operações e as arestas representam tensores — objetos matemáticos n-dimensionais que generalizam vetores e matrizes.

Uma das características importante do *TensorFlow* no contexto supervisionado é a integração com a API `tf.keras`, que permite construir, treinar e avaliar modelos de forma concisa e modular. A API sequencial e a funcional possibilitam desde arquiteturas simples até modelos complexos, como redes profundas com múltiplas entradas e saídas. A construção de modelos supervisionados em `tf.keras` segue um ciclo padronizado:

1. Definição da arquitetura da rede (*Sequential* ou *Functional API*);
2. Compilação do modelo com especificação da função de perda, otimizador e métricas;
3. Treinamento com o método `fit()` a partir de dados rotulados;
4. Avaliação com `evaluate()` e geração de previsões com `predict()`.

No aprendizado supervisionado, a escolha da função de perda é crucial para que o modelo aprenda a minimizar o erro entre previsão e rótulo real. O *TensorFlow* fornece funções de perda otimizadas para diferentes tarefas:

- `mean_squared_error` para regressão;
- `binary_crossentropy` para classificação binária;
- `sparse_categorical_crossentropy` para classificação multiclasse com rótulos inteiros;
- `categorical_crossentropy` para rótulos *one-hot* codificados.

A biblioteca oferece também diversos otimizadores, como `SGD`, `RMSProp`, `Adam` e suas variantes, cada um com características próprias de atualização dos pesos. O `Adam`, por exemplo, é amplamente utilizado por seu bom desempenho em problemas com grande dimensionalidade (Abrahams; Smith; Zhang, 2016).

Se tratando de treinamento, o método `model.fit()` permite o treinamento supervisionado com diversas funcionalidades embutidas:

- Controle de *batch size*, número de épocas e conjunto de validação;
- *Callbacks*, como `EarlyStopping`, `ModelCheckpoint` e `TensorBoard`, que facilitam a visualização de métricas, o salvamento de modelos e a interrupção automática do treinamento quando a performance estagna;
- Suporte a treinamento distribuído por meio da API `tf.distribute`, que permite escalar o treinamento em múltiplas GPUs, TPUs ou máquinas.

Para usuários mais experientes, o *TensorFlow* também disponibiliza mecanismos para a implementação de *loops* de treinamento personalizados por meio da API `tf.GradientTape`, permitindo controle granular sobre o cálculo dos gradientes e a atualização dos pesos da rede, o que se mostra especialmente útil em experimentações avançadas e em cenários de pesquisa que demandam maior flexibilidade computacional (Géron, 2019). Uma vez finalizado o processo de treinamento, o modelo supervisionado pode ser serializado em diferentes formatos, como `.h5` ou `SavedModel`, o que garante portabilidade, reutilização e integração com outras aplicações.

A biblioteca também oferece suporte à implantação dos modelos treinados em ambientes de produção por meio de diferentes ferramentas especializadas: o *TensorFlow Serving*, voltado para o uso em servidores e sistemas de backend; o *TensorFlow Lite (TFLite)*, que possibilita a execução eficiente em dispositivos móveis e embarcados; e o *TensorFlow.js*, que permite a execução de modelos diretamente em navegadores web, viabilizando aplicações baseadas em cliente com inferência em tempo real (Chollet, 2021).

5.3.2 Keras

O *Keras* desempenha um papel fundamental no contexto do aprendizado de máquina (*Machine Learning*), especialmente no campo do *deep learning*. Trata-se de uma API de alto nível desenvolvida para simplificar a definição, treinamento, avaliação e implantação de modelos de redes neurais de forma intuitiva e modular. Atualmente, o Keras é distribuído como parte integrante do *TensorFlow*, na forma do módulo `tf.keras`.

Conforme destaca (Géron, 2019), “*Keras is a high-level Deep Learning API that makes it very simple to train and run neural networks*”. A biblioteca permite a implementação eficiente de modelos como perceptrons multicamadas (MLPs), redes neurais convolucionais (CNNs) e redes neurais recorrentes (RNNs), utilizadas com sucesso em tarefas como classificação de imagens, análise de sentimentos, tradução automática, detecção de objetos, entre outras. Diferentemente de abordagens tradicionais de aprendizado supervisionado — como regressão logística, máquinas de vetores de suporte (SVM) e árvores de decisão — que exigem forte intervenção na engenharia de características, as redes neurais implementadas com Keras extraem automaticamente representações relevantes a partir de dados brutos, como imagens, sinais e texto.

Uma das principais vantagens do *Keras* reside na sua integração nativa com o *TensorFlow* 2, possibilitando o uso de aceleração por GPU, serialização de modelos nos formatos `.h5` e `SavedModel`, monitoramento via *TensorBoard*, além de implantação em diferentes plataformas por meio do *TensorFlow Serving*, *TensorFlow Lite* ou *TensorFlow.js* (Chollet, 2021). No processo de modelagem, oferece suporte a técnicas modernas como funções de ativação (`relu`, `sigmoid`, `softmax`), funções de perda (`binary_crossentropy`, `mean_squared_error`), otimizadores (`Adam`, `RMSprop`), e mecanismos de regularização como *Dropout*, ℓ_1/ℓ_2 , *early stopping* e *callbacks* personalizados para ajuste dinâmico dos hiperparâmetros.

5.3.3 Scikit-Learn

O *Scikit-Learn* é uma das bibliotecas utilizadas para aprendizado de máquina com Python (Chollet, 2021). Desenvolvido originalmente como uma extensão da biblioteca SciPy, o *Scikit-Learn* foi projetado para fornecer ferramentas eficientes, reutilizáveis e acessíveis para análise preditiva baseada em algoritmos estatísticos clássicos (Géron, 2019). A biblioteca oferece uma gama abrangente de algoritmos supervisionados e não supervisionados, incluindo regressão linear e logística, árvores de decisão, máquinas de vetores de suporte (SVM), algoritmos de vizinhos mais próximos (*k-Nearest Neighbors*), *Random Forests*, análise de componentes principais (PCA), entre outros.

Uma das características do *Scikit-Learn* é sua API padronizada e coerente. O fluxo de trabalho típico envolve as seguintes etapas: (i) pré-processamento dos dados com transformadores como `StandardScaler` e `OneHotEncoder`, (ii) divisão em conjuntos de treino e teste com `train_test_split`, (iii) criação e ajuste de modelos com o método `fit()`, e (iv) avaliação por meio de funções como `accuracy_score`, `confusion_matrix` ou `mean_squared_error`.

A biblioteca também fornece suporte para técnicas fundamentais de avaliação e otimização de modelos, como *validação cruzada* (`cross_val_score`), busca em grade (`GridSearchCV`) e pipelines de processamento (`Pipeline`), que integram pré-processamento e modelagem em uma única estrutura reutilizável.

6 DESENVOLVIMENTO DO SISTEMA SMACARE

Este estudo foi aprovado pelo Comitê de Ética em Pesquisa com Seres Humanos da Universidade do Estado de Santa Catarina (CAAE 82997424.7.0000.0118). O Sistema Multiagente (SMA) proposto neste trabalho é concebido como um *framework* extensível para monitoramento de sinais vitais. Sua arquitetura modular, baseada em agentes especializados, permite a integração de diferentes dispositivos de aquisição de dados, bem como a personalização de modelos de inferência para diagnóstico e prognóstico. Este capítulo apresenta o detalhamento das etapas de desenvolvimento e a arquitetura completa do sistema proposto, integrando hardware vestível, infraestrutura de comunicação e modelos de inteligência artificial implementados no SMA. Descreve também a concepção, o projeto e a integração dos principais componentes, incluindo o desenvolvimento do hardware com sensores fisiológicos, a definição da arquitetura de software distribuído e a modelagem de redes neurais para classificação e prognóstico clínico.

As seções subsequentes abordam de forma sistemática as escolhas de projeto, as heurísticas adotadas para definição das arquiteturas de redes neurais artificiais e os métodos de pré-processamento dos dados fisiológicos coletados. São descritas as estratégias de regularização, otimização e balanceamento de classes no treinamento supervisionado dos modelos, bem como a integração dos resultados no SMA para inferência em tempo real e apoio à decisão clínica, destacando a coerência entre as diferentes etapas do sistema e sua aplicabilidade em cenários de saúde domiciliar.

6.1 HARDWARE

O primeiro passo para a implementação do hardware como meio de coleta de dados desta pesquisa consistiu na realização de revisão bibliográfica voltada para dispositivos vestíveis com uso de microcontroladores, com o objetivo inicial de comparar e prototipar um detector de quedas. Durante essa etapa, foram identificados os tipos mais comuns empregados, destacando-se os localizados no pulso (*smartwatches*), localizados na pelve, localizados no braço e os localizados no peito. Com base nessas informações, desenvolveu-se um protótipo de detector de quedas para uso no peito e este foi testado e validado com um autor sendo o voluntário da pesquisa.

Foram realizados testes abrangentes, incluindo atividades como andar, parar, sentar, levantar e correr, com resultados compilados ao longo de uma semana. Posteriormente, diferentes tipos de quedas foram simuladas para fins de comparação dos sinais. O protótipo, os testes e os resultados desta primeira fase da pesquisa culminaram na publicação de um artigo intitulado **Low Cost System for Fall Detection in the Elderly** (Filgueiras; Torres; Bertemes-Filho, 2020). Embora o dispositivo tenha demonstrado eficácia, a localização do detector mostrou-se desconfortável.

Observando-se essa limitação, deu-se início ao desenvolvimento de um bracelete inteligente com conexão digital (no dedo), capaz de detectar quedas e monitorar, em tempo real, sinais biológicos relevantes. O dispositivo foi projetado para fornecer um histórico abrangente de saúde

do usuário, auxiliando tanto no prognóstico clínico quanto na detecção precoce de quedas. Além da configuração no dedo, o bracelete também pode ser utilizado no braço, com sensor conectado ao lóbulo da orelha, permitindo múltiplas configurações anatômicas, adaptáveis às necessidades e preferências dos usuários.

Com base em uma revisão atualizada da literatura científica, definiu-se o monitoramento dos seguintes sinais vitais para esta investigação: batimento cardíaco, pressão arterial estimada e saturação de oxigênio no sangue (SpO_2). Adicionalmente, incorporou-se a medição da temperatura corporal, utilizando um sensor infravermelho acoplado ao dispositivo. O dispositivo vestível desenvolvido neste projeto incorpora um módulo de armazenamento baseado em cartão *microSD*, cuja finalidade é garantir a persistência dos dados adquiridos localmente. Toda vez que o dispositivo é ativado para uso, os sinais fisiológicos coletados são automaticamente registrados em um arquivo com extensão *.csv* no *microSD* inserido, viabilizando sua utilização posterior para auditoria, recuperação de dados e análise offline no contexto do Cenário 2 do sistema.

De forma padronizada e sequencial, o dispositivo executa três operações essenciais a cada ciclo de utilização:

1. Grava localmente os sinais fisiológicos em arquivo *.csv* no *microSD*;
2. Transmite os dados em tempo real, via *Bluetooth Low Energy (BLE)*, para o smartphone previamente pareado e autorizado;
3. Envia simultaneamente os mesmos dados, também via BLE, ao Sistema Multiagente (SMA), possibilitando o diagnóstico em tempo real por meio de inferência neural distribuída.

Essa estratégia assegura redundância nos registros, suporte à auditoria médica e integridade dos dados em contextos de conectividade intermitente, além de permitir o funcionamento assíncrono do Cenário 2, a ser descrito na seção 6.2.

Para a estimativa da pressão arterial, foi empregado um sensor comercial cujo *firmware* embarcado já incorpora os algoritmos necessários para o cálculo dos valores sistólico e diastólico, bem como da saturação periférica de oxigênio (SpO_2) e da frequência cardíaca. Esses parâmetros são disponibilizados diretamente pelo dispositivo, dispensando qualquer processamento adicional pelo sistema. De acordo com o fabricante, os algoritmos implementados no sensor foram previamente calibrados, assegurando a coerência dos valores retornados dentro dos limites esperados de operação.

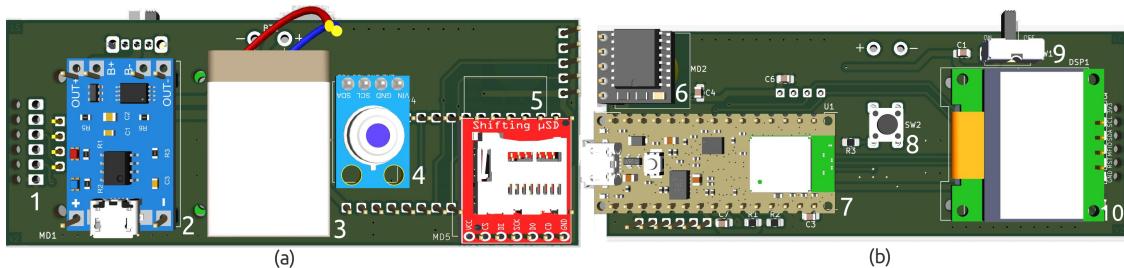
O desenvolvimento do protótipo inicial foi realizado utilizando o microcontrolador Arduino Nano BLE Sense Rev2, montado sobre uma *protoboard* para viabilizar o processo de prototipagem e a execução de testes preliminares. Para a aquisição dos sinais de pressão arterial estimada (BPT), saturação periférica de oxigênio (SpO_2) e frequência cardíaca (BPM), foi empregado o módulo Protocentral PC-MED-0411. A medição da temperatura corporal

foi realizada por meio do sensor infravermelho sem contato MLX90614, escolhido pela sua capacidade de aferição não invasiva e pela viabilidade de integração ao microcontrolador.

Após a validação funcional da montagem em *protoboard*, foi projetada e fabricada uma placa de circuito impresso (PCI) (Figura 11) com o objetivo de integrar todos os componentes de forma mais robusta e confiável, eliminando limitações inerentes à prototipagem inicial. Testes preliminares foram realizados com um dos autores atuando como voluntário, a fim de validar tecnicamente o funcionamento da placa. Em seguida, o protótipo passou por um processo de validação comparativa com dispositivos comerciais ao longo de um período de dez dias, envolvendo dez participantes. Os resultados dessa etapa serviram de base para a redação de um artigo científico intitulado **Evaluating the Accuracy of Low-Cost Wearable Sensors for Healthcare Monitoring** (Filgueiras; Bertemes-Filho; Noveletto, 2025), publicado na revista *Micromachines* (DOI: <https://doi.org/10.3390/mi16070791>).

Na sequência, foram produzidas unidades adicionais do dispositivo, com o intuito de viabilizar uma validação mais ampla envolvendo um total de trinta participantes (diagramas do dispositivo e da case disponíveis nos Apêndices C e D). Contudo, devido a imprevistos pessoais da autora e a atrasos relacionados ao processo de aprovação pelo comitê de ética, o cronograma disponível para a finalização desta pesquisa foi impactado. Em função dessas restrições, decidiu-se postergar a implementação e avaliação da funcionalidade de detecção de quedas, prevendo sua incorporação em estudos futuros.

Figura 11 – Ilustrações do dispositivo vestível fabricado. (a) Vista traseira da placa, mostrando a interface de 6 pinos do módulo MAX32664 (1), o módulo de gerenciamento de energia (2), a bateria (3), o sensor infravermelho (IR) (4) e a interface para cartão microSD (5). (b) Vista frontal da placa, incluindo o relógio em tempo real (RTC) (6), o microcontrolador nRF52840 (7), o botão para reativar o visor a partir do modo de espera (8), o interruptor liga/desliga (9) e o visor (10).



Fonte: Elaborado pelo autor (2025).

6.2 SISTEMA MULTIAGENTE

O sistema multiagente (SMA) desenvolvido nesta pesquisa foi implementado na linguagem de programação Java, utilizando a IDE *IntelliJ IDEA* e o framework *JADE* (*Java Agent DEvelopment Framework*). Como sistema gerenciador de banco de dados, adotou-se o

PostgreSQL, devido à sua robustez e compatibilidade com operações concorrentes em tempo real.

Os agentes inteligentes responsáveis pela inferência utilizam modelos de inteligência artificial previamente treinados em Python. Para assegurar a interoperabilidade entre as linguagens, cada agente emprega a classe Java `ProcessBuilder` para invocar diretamente os classificadores implementados. Essa estratégia demonstrou melhor desempenho e maior robustez, além de minimizar problemas de compatibilidade decorrentes de diferenças entre versões do *TensorFlow* disponíveis para Java e Python.

A arquitetura inicial do SMA foi concebida com dez agentes inteligentes estáticos, sendo cinco agentes principais e cinco subagentes. A cada novo dispositivo/participante conectado ao sistema, são instanciados dinamicamente um novo `BrokerAgent` e um novo `DBAgent`, ambos dedicados ao respectivo par dispositivo/participante, aumentando a quantidade de agentes do SMA. O SMACare atua em dois cenários distintos:

Cenário 1

Coleta de dados em tempo real via comunicação BLE (*Bluetooth Low Energy*) diretamente de um dispositivo vestível (Figura 12). Para este cenário, os agentes inteligentes são descritos:

MainAgent:

Responsável pela inicialização do sistema. Suas principais funcionalidades são:

- Exibir ao usuário as opções de operação disponíveis: leitura via BLE (Cenário 1) ou leitura a partir de arquivo CSV (Cenário 2);
- Criar o `DBAgent` responsável pelo cadastro de usuários e dispositivos no banco de dados;
- Receber e repassar ao `DBAgent` os dados de registro de novos usuários e dispositivos;
- Criar um `BrokerAgent` específico para cada nova associação entre dispositivo e usuário, mantendo uma lista atualizada com os endereços de todos os `BrokerAgents` ativos;
- Iniciar o `AnalysisAgent`, único no sistema, responsável por processar os dados clínicos provenientes de todos os dispositivos/usuários conectados;
- Iniciar o `NotifyAgent`.

BrokerAgent:

O agente `BrokerAgent` é criado dinamicamente para cada associação entre dispositivo e usuário, sendo responsável pelas seguintes funções:

- Receber os dados transmitidos via BLE a partir do dispositivo vestível correspondente;
- Encaminhar os dados recebidos ao DBAgent responsável pelo respectivo par dispositivo/usuário;
- Instanciar, caso ainda não exista, o DBAgent dedicado ao gerenciamento dos dados daquele dispositivo e usuário.

DBAgent:

O agente DBAgent é inicialmente criado a pedido do MainAgent para realizar o cadastro de usuários e dispositivos no banco de dados, sendo finalizado logo após a conclusão dessa tarefa. Posteriormente, novos DBAgents são instanciados dinamicamente pelos BrokerAgents, de forma dedicada a cada par dispositivo/usuário. Suas responsabilidades incluem:

- Armazenar os dados de dispositivos e usuários nas tabelas `wearable_registry` e `user_registry`;
- Registrar o mapeamento entre dispositivo e usuário na tabela `device_registry`, cuja chave primária é composta por duas tabelas: `ID_Wearable` e `ID_Participante`;
- Criar e alimentar tabelas no formato `data_dv<ID_DISPOSITIVO>_sub<ID_USUARIO>`, contendo os dados clínicos de temperatura, SpO₂, frequência cardíaca, pressão sistólica e diastólica;
- Criar *triggers* nessas tabelas para notificar o AnalysisAgent a cada nova inserção de dados;
- Armazenar os diagnósticos recebidos dos subagentes especializados (BPTAgent, BPMAgent, TemperatureAgent e SP0Agent) na tabela `history_sub<ID_USUARIO>`, com controle temporal baseado em *timestamp*;
- Registrar os resultados dos prognósticos fornecidos pelo PrognosisAgent na tabela `prognosis_data_dv<ID_DISPOSITIVO>_sub<ID_USUARIO>`.

AnalysisAgent:

Criado uma única vez pelo MainAgent, possui as seguintes funcionalidades:

- Ouvir as *triggers* das tabelas de dados para recuperar os registros recém-inseridos;
- Instanciar, na primeira leitura, os subagentes de diagnóstico: TemperatureAgent, BPTAgent, FallAgent, BPMAgent e Sp02Agent, os quais atuam de forma compartilhada para todos os dispositivos/usuários;

- Enviar para cada subagente os dados correspondentes a especialidade do mesmo;
- A cada cinco minutos, criar um novo `PrognosisAgent` para realizar prognósticos;
- Garantir a criação das tabelas de prognóstico, caso ainda não existam;
- Recebe as inferições dos quatro subagentes e roda o algoritmo para inferir diagnóstico multi-label;
- Em caso de anormalidade, notifica o `NotifyAgent`.

Subagentes de Diagnóstico (`TemperatureAgent`, `BPTAgent`, `BPMAgent`, `SPOAgent`):

Cada subagente processa a métrica correspondente utilizando redes neurais. Os modelos de inteligência artificial foram treinados em Python, e são executados pelos agentes via `ProcessBuilder` em Java, o que garante compatibilidade e desempenho. Após o processamento:

- Em caso de detecção de anomalias, envia ào `NotifyAgent`;
- Envia ao `DBAgent` e ào `AnalysisAgent` o resultado da inferência realizada.

Subagente de Prognóstico (`PrognosisAgent`):

Iniciado periodicamente (de cinco em cinco minutos) pelo `AnalysisAgent`, este subagente executa as seguintes ações:

- Lê os dados mais recentes do histórico do paciente;
- Executa o modelo de prognóstico;
- Notifica o `NotifyAgent` em caso de alterações no estado de saúde do paciente;
- Finaliza sua execução automaticamente após concluir o processamento.

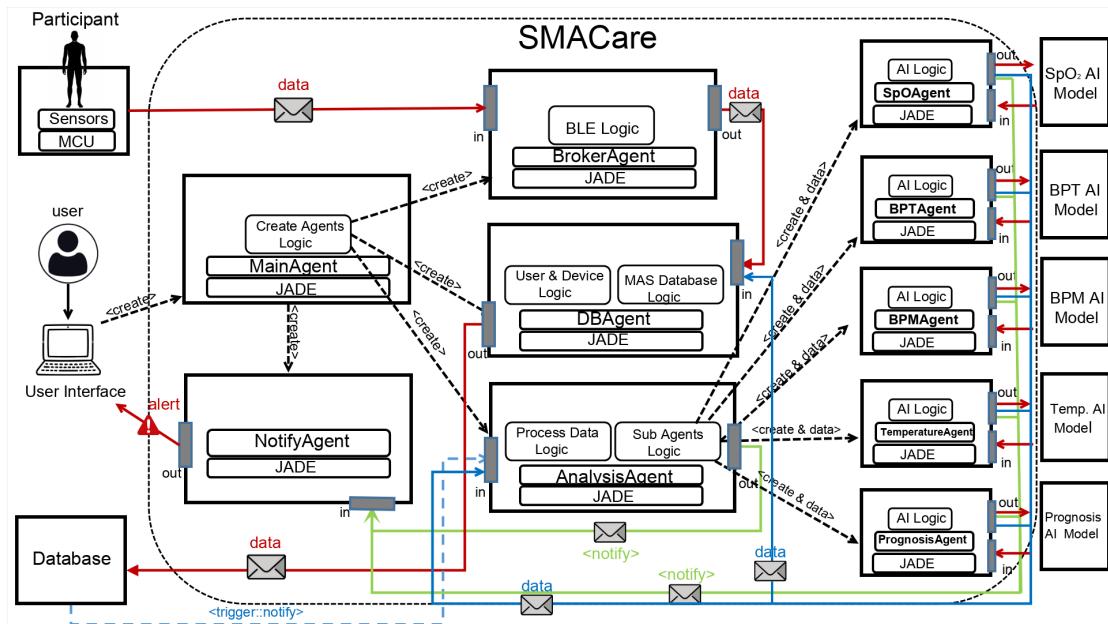
NotifyAgent:

Este agente é responsável por:

- Receber notificações de anomalias provenientes dos `AnalysisAgent` e do `PrognosisAgent`;
- Informar cuidadores, profissionais de saúde e usuários sobre situações críticas ou emergenciais;
- Exibir, na interface do sistema, a condição atual do paciente, sendo **NORMAL** o estado padrão.

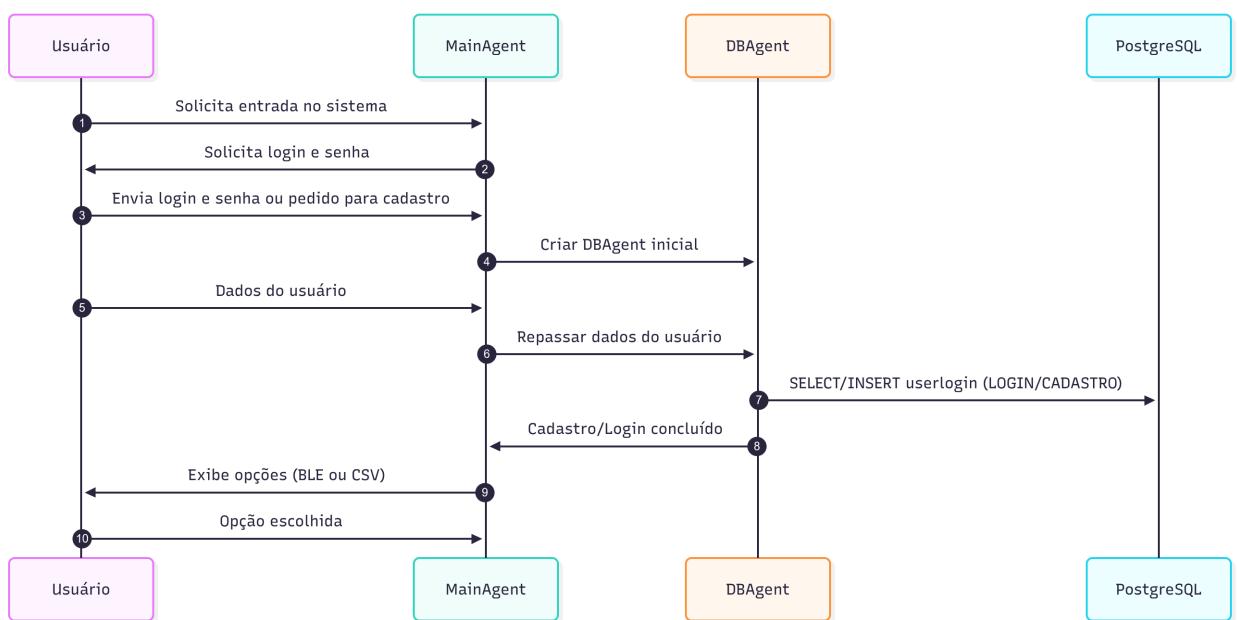
Os diagramas de sequência do funcionamento do SMACare no Cenário 1 são ilustrados nas Figuras 13 a 16.

Figura 12 – Funcionamento do SMACare - Cenário 1



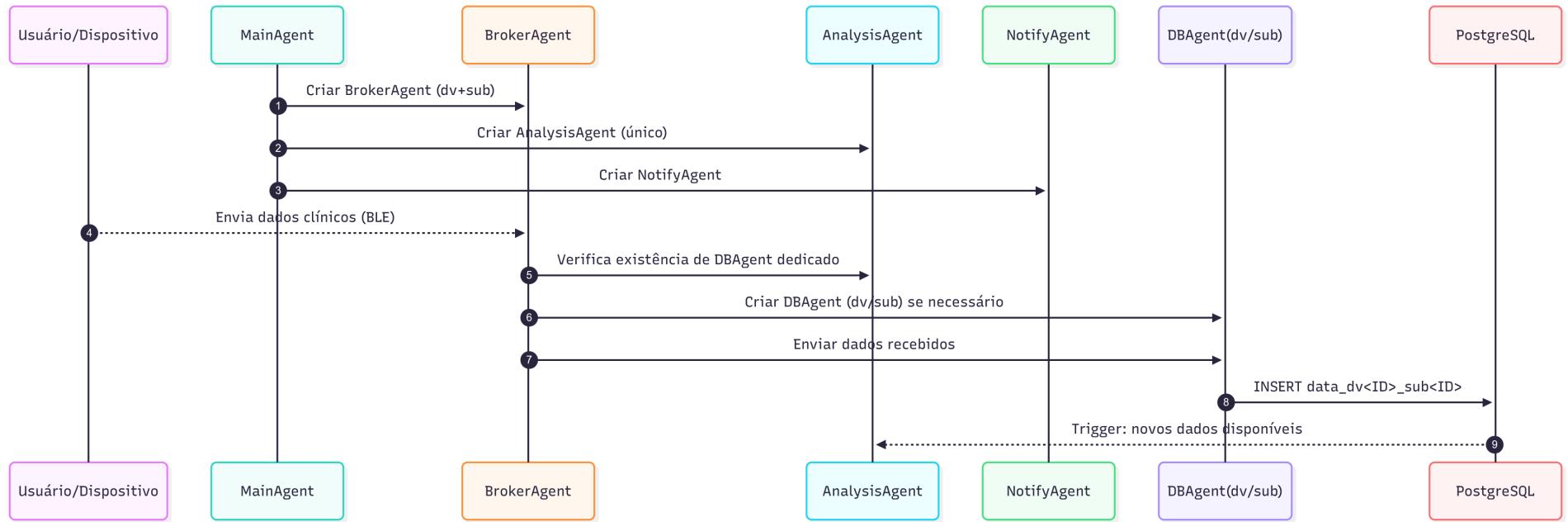
Fonte: Elaborado pela autora (2025).

Figura 13 – Diagrama de sequência das etapas de login e cadastro, válido para ambos os Cenários



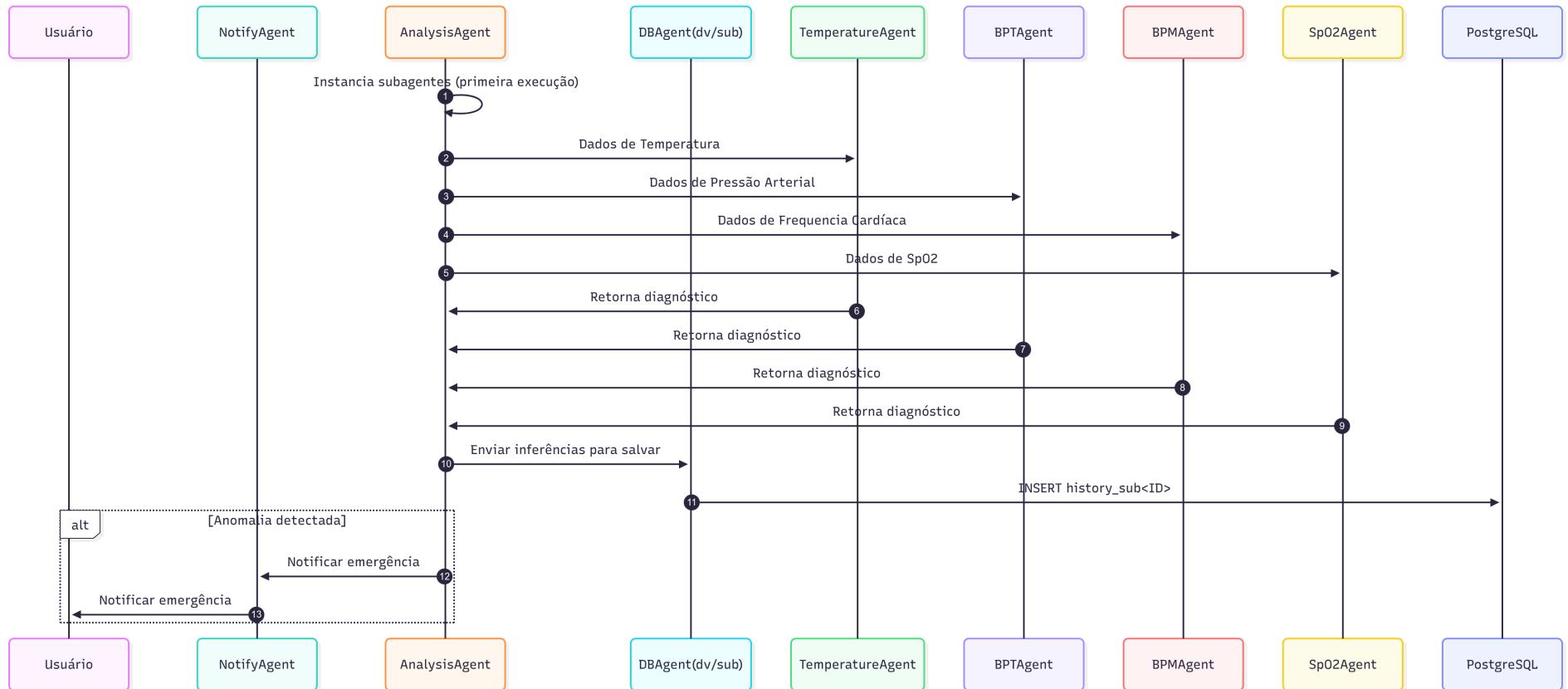
Fonte: Elaborado pela autora (2025).

Figura 14 – Diagrama de sequência das etapas pós login e preparação do sistema para as inferências



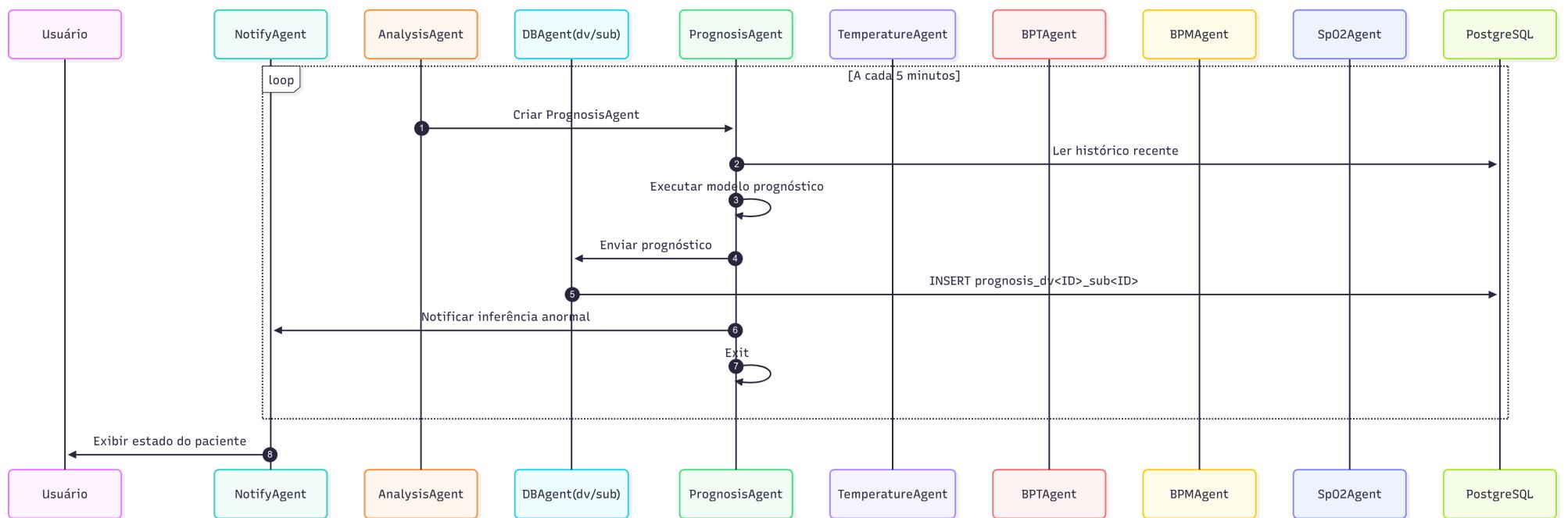
Fonte: Elaborado pela autora (2025).

Figura 15 – Diagrama de sequência das etapas de Inferência e Notificação



Fonte: Elaborado pela autora (2025).

Figura 16 – Diagrama de sequência da etapa de Prognóstico



Fonte: Elaborado pela autora (2025).

Cenário 2

Neste cenário, os dados não são coletados em tempo real, mas sim importados de um arquivo .csv contendo dados de um participante específico (Figura 17). O MainAgent é responsável por criar os seguintes agentes: BrokerAgent, NotifyAgent, AnalysisAgent e ShutdownAgent.

BrokerAgent:

- Recebe do MainAgent o arquivo .csv contendo os dados de um participante;
- Encaminha os dados ao AnalysisAgent para processamento.

AnalysisAgent:

- Lê os dados recebidos do BrokerAgent;
- Cria os subagentes de diagnóstico;
- Ao final das inferências dos subagentes, infere o diagnóstico multi-label;
- Ao final da leitura do arquivo *.csv:
 - Instancia o PrognosisAgent;
 - Informa o ShutdownAgent sobre a finalização da leitura.

Subagentes de Diagnóstico (TemperatureAgent, BPTAgent, BPMAgent, SPOAgent):

Os subagentes funcionam de forma idêntica ao Cenário 1.

Subagente de Prognóstico (PrognosisAgent):

- Realiza o prognóstico com base nos dados lidos;
- Notifica o NotifyAgent em caso de anormalidades;
- Informa o ShutdownAgent sobre a conclusão da tarefa;
- Finaliza sua execução automaticamente.

NotifyAgent:

Este agente mantém o mesmo comportamento descrito no Cenário 1.

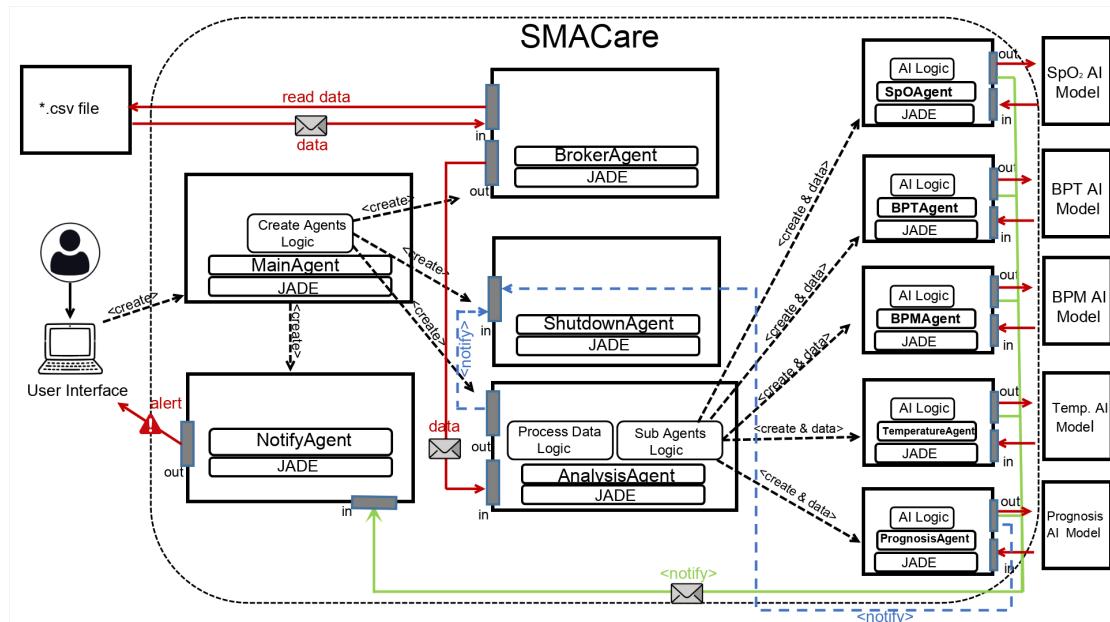
ShutdownAgent

Agente existente apenas no Cenário 2. Suas principais funcionalidades são:

- Aguarda as confirmações de término enviadas pelo AnalysisAgent e pelo PrognosisAgent;
- Encerra o Sistema Multiagente 1 minuto após receber as notificações do AnalysisAgent e do PrognosisAgent.

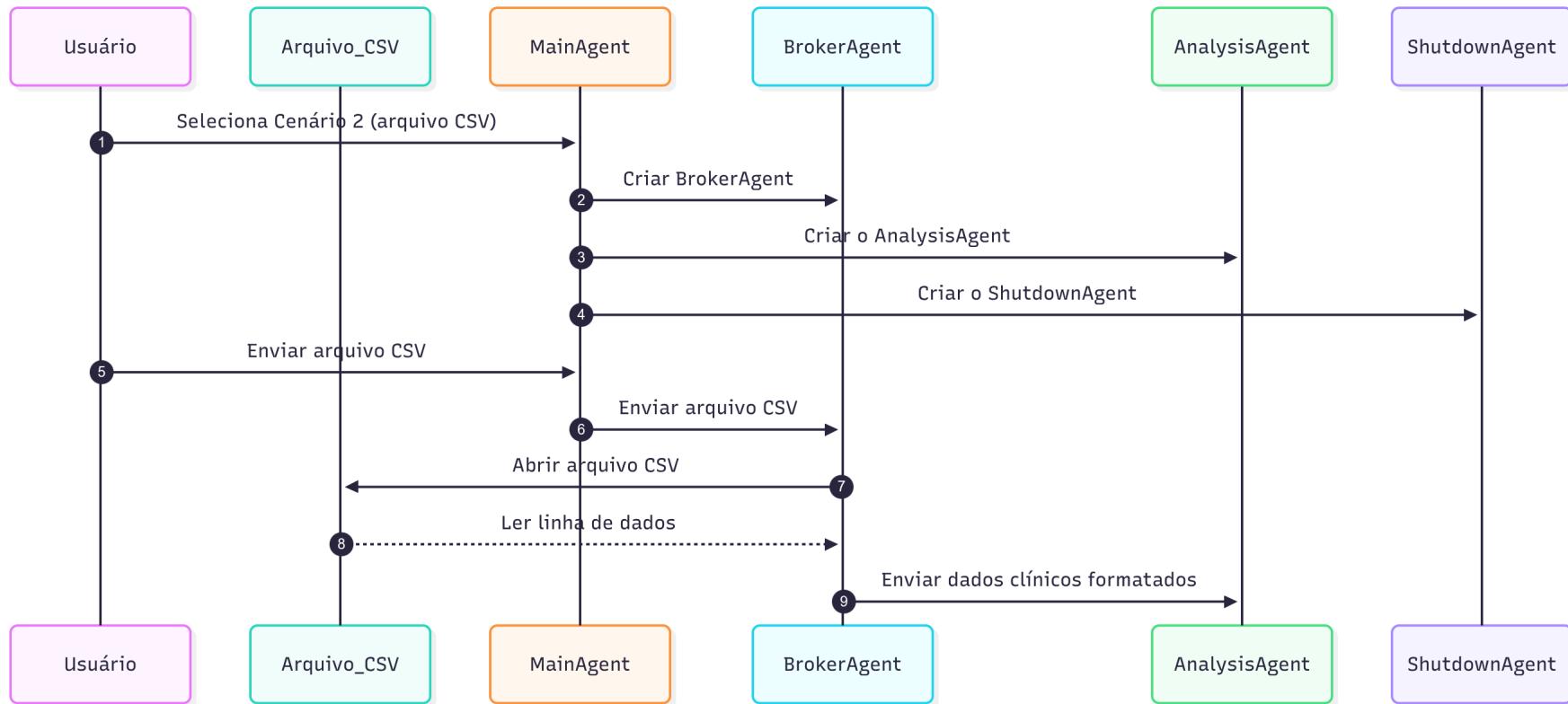
A Figura 17 ilustra o funcionamento do SMACare nesse cenário. Os diagrams de sequência ilustrando o funcionamento do SMACare no Cenário 2 encontram-se nas Figuras 18 a 21.

Figura 17 – Funcionamento do SMACare - Cenário 2



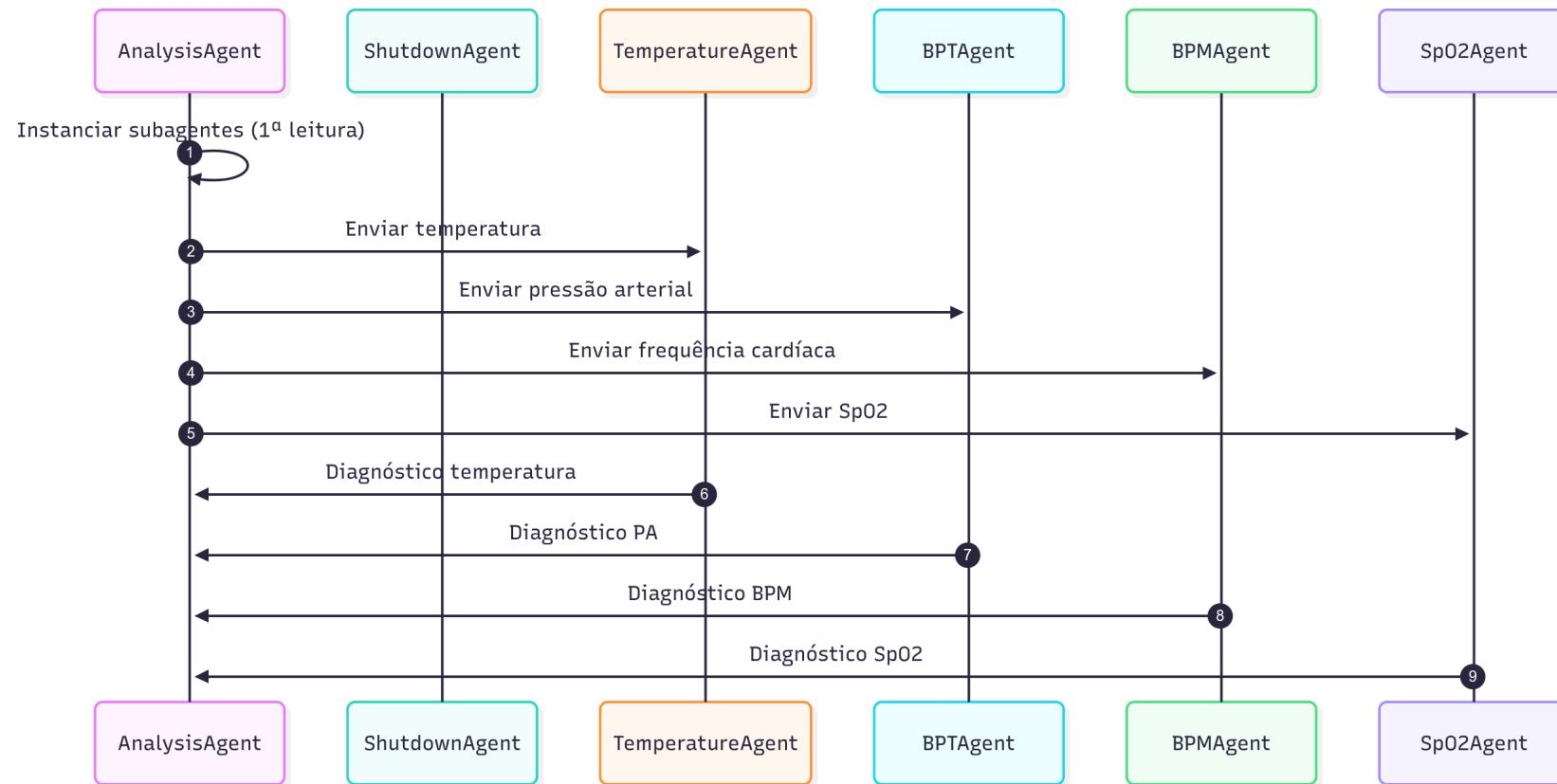
Fonte: Elaborado pela autora (2025).

Figura 18 – Diagrama de sequência das etapas pós login e preparação do sistema para as inferências



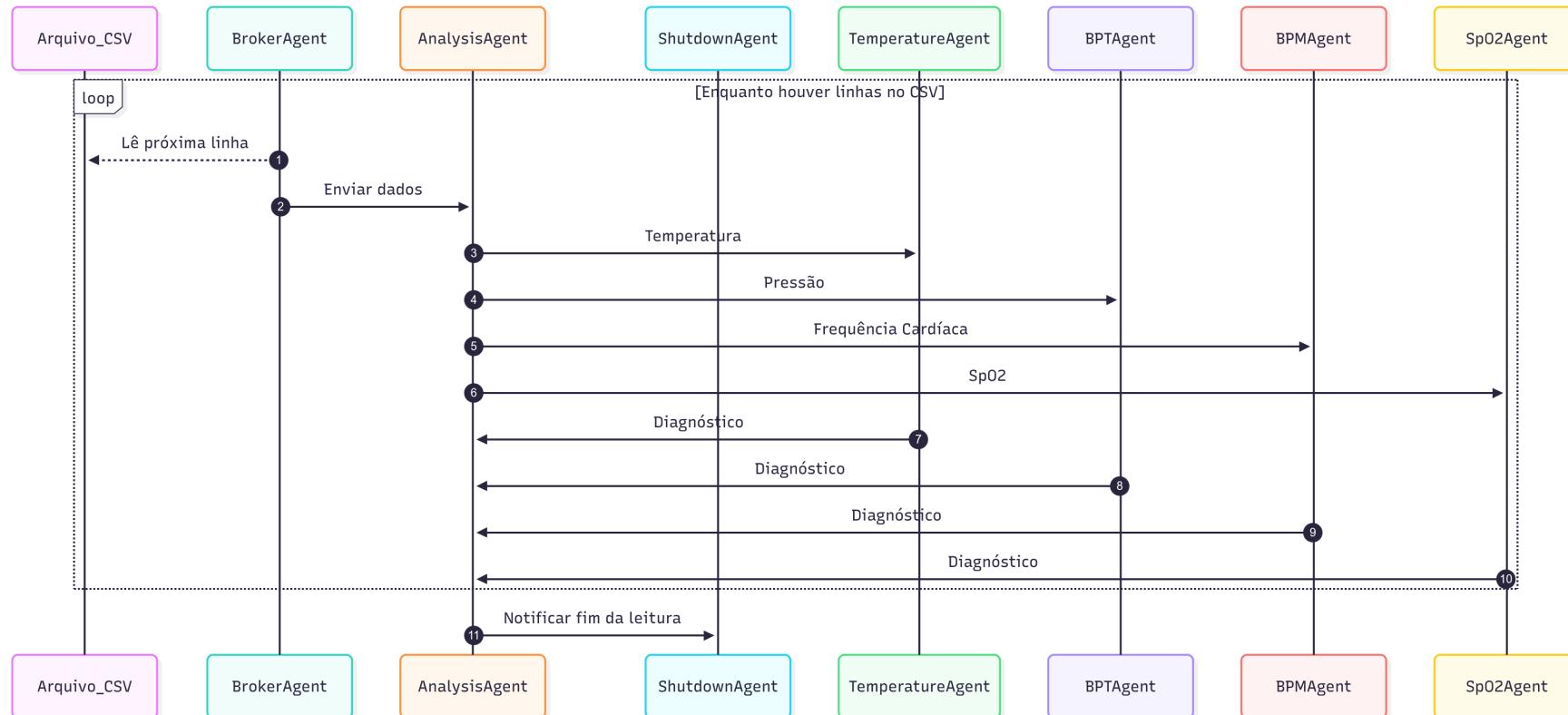
Fonte: Elaborado pela autora (2025).

Figura 19 – Diagrama de sequência das etapas de Inferência



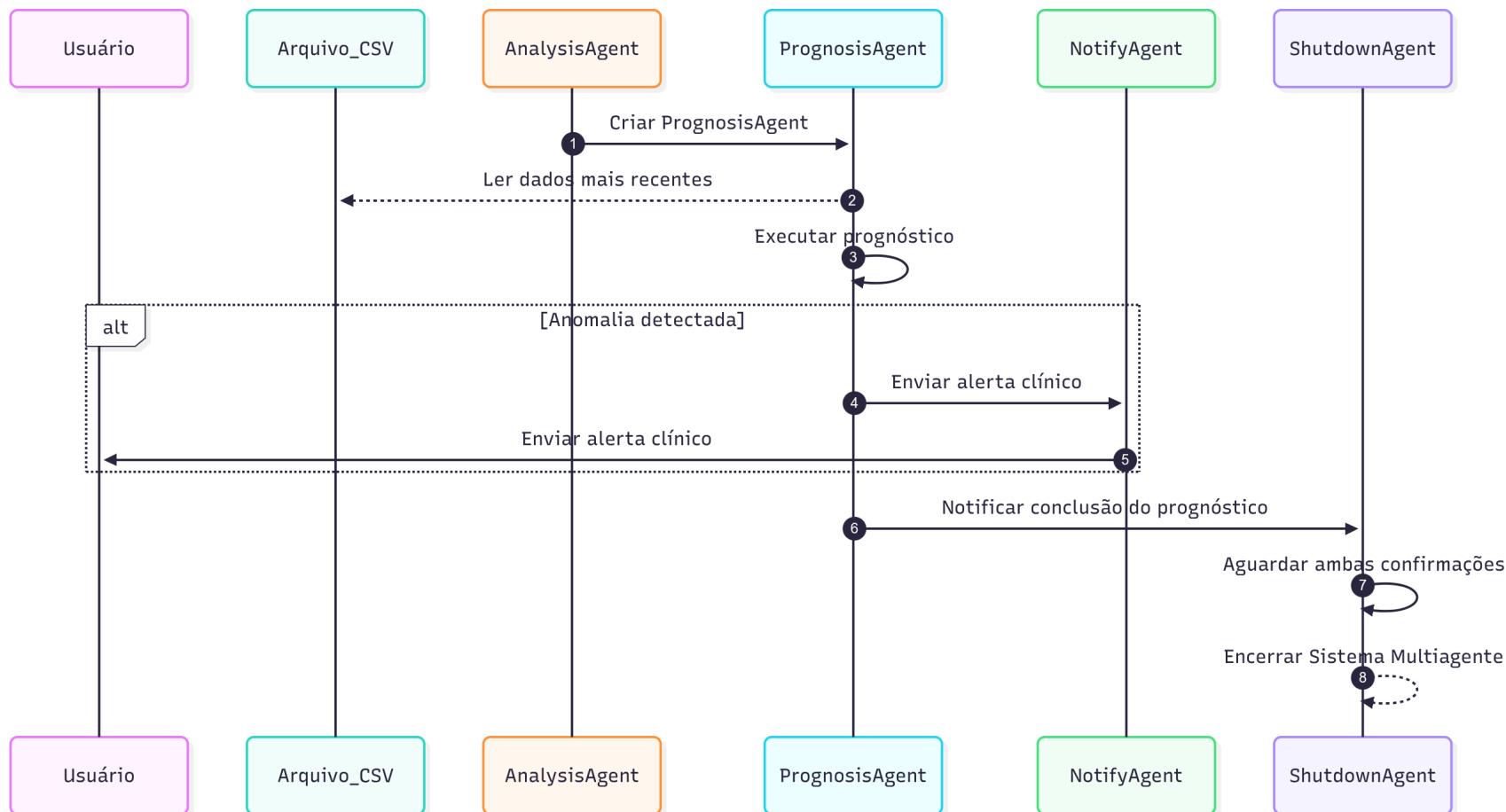
Fonte: Elaborado pela autora (2025).

Figura 20 – Diagrama de sequência das etapas de inferência por linha de arquivo



Fonte: Elaborado pela autora (2025).

Figura 21 – Diagrama de sequência da etapa de Prognóstico



Fonte: Elaborado pela autora (2025).

6.2.1 Estrutura do Banco de Dados e Integração com o Sistema Multiagente

Para a persistência e gerenciamento dos dados gerados pelo sistema, foi adotado o Sistema Gerenciador de Banco de Dados Relacional PostgreSQL. A escolha por esta tecnologia deve-se à sua robustez, à conformidade com os padrões ACID e, sobretudo, à sua capacidade nativa de suporte a *triggers*, recurso essencial para a arquitetura proposta. Cada vez que um novo dado é inserido no banco de dados, uma *trigger* é acionada automaticamente, notificando o AnalysisAgent para a realização das tarefas de análise e inferência em tempo real.

O processo de autenticação e identificação dos usuários é realizado por meio de login ou registro, cujos dados são armazenados na tabela `userlogin`. Esta tabela é composta pelos seguintes campos: `id`, `username`, `pwd` (senha), `access` (registro do último acesso ao sistema) e `loginname`. O campo `loginname` é padronizado como `sub<id>`, de forma a garantir a associação explícita entre o identificador do usuário e o sujeito clínico monitorado.

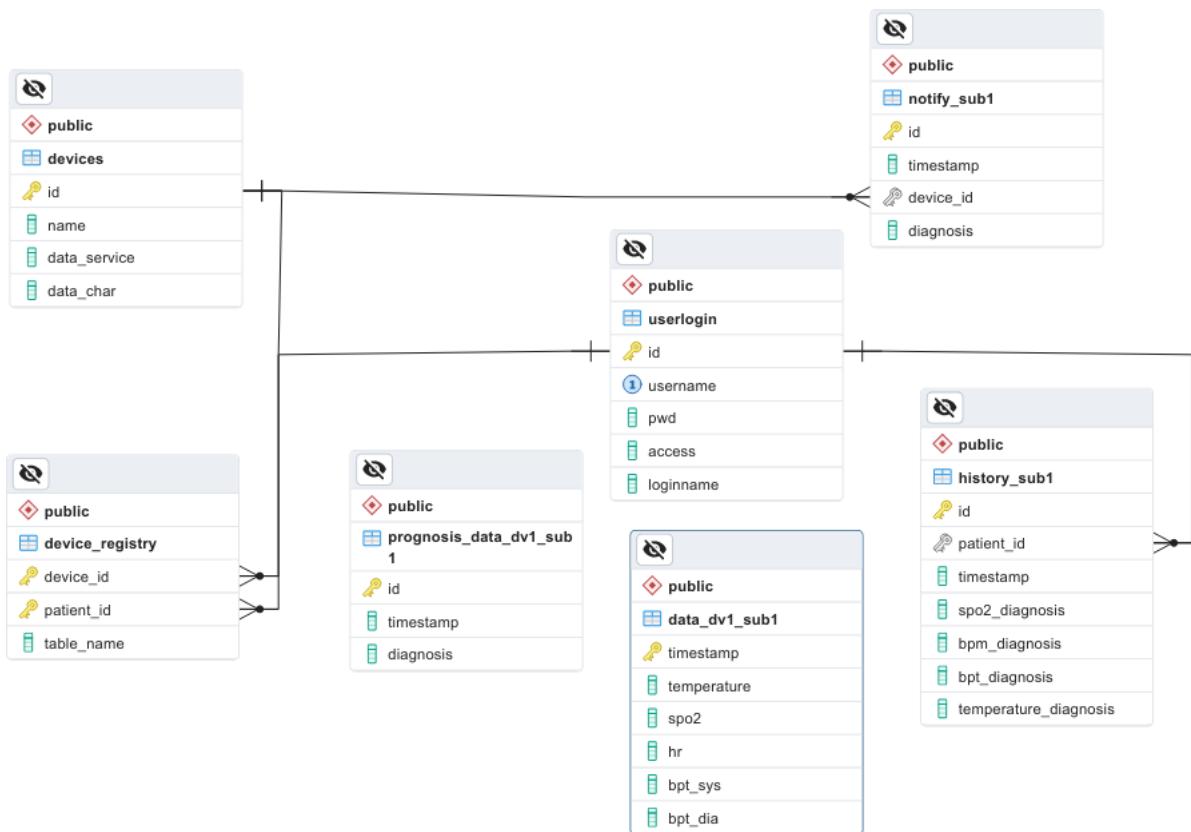
O cadastro dos dispositivos vestíveis disponíveis no sistema é mantido na tabela `devices`, a qual armazena os campos `id`, `name`, `DATA_SERVICE` e `DATA_CHAR`. Os dois últimos campos referem-se aos identificadores dos serviços e características utilizados para a conexão via Bluetooth Low Energy (BLE). A inserção de novos dispositivos é de responsabilidade exclusiva do administrador do sistema.

Sempre que uma nova instância de pareamento entre um usuário e um dispositivo é estabelecida, os dados correspondentes são registrados na tabela `device_registry`. Essa tabela contém os campos `device_id` (chave estrangeira para `devices`), `patient_id` (chave estrangeira para `userlogin`) e `table_name`, que indica o nome da tabela na qual os dados recebidos do dispositivo serão armazenados. Caso a tabela de dados `data_dv<id do dispositivo>_sub<id do paciente>` ainda não exista, ela é automaticamente criada. Se já existir, o sistema continua o processo de inserção a partir do último registro. Esta tabela contém os seguintes campos: `timestamp` (chave primária), `temperature`, `spo` (saturação de oxigênio), `hr` (frequência cardíaca), `bpt_sys` (pressão arterial sistólica) e `bpt_dia` (pressão arterial diastólica), todos provenientes diretamente dos sensores vestíveis embarcados.

No momento em que os dados são processados pelos subagentes do sistema multiagente, o DBAgent é responsável por criar, caso ainda não exista, a tabela `history_sub<id>`. Essa tabela armazena os diagnósticos inferidos pelos agentes especializados em sinais fisiológicos. Seus campos incluem: `id`, `patient_id` (chave estrangeira para `userlogin`), `timestamp` (momento da inferência), `spo2_diagnosis` (resultado do agente de SpO_2), `bpm_diagnosis` (resultado do agente de frequência cardíaca), `bpt_diagnosis` (resultado do agente de pressão arterial) e `temperature_diagnosis` (resultado do agente de temperatura). Após o processamento pelos subagentes, os dados podem ser encaminhados ao PrognosisAgent, responsável pela inferência de prognósticos clínicos baseados no histórico do paciente. Os resultados são armazenados na tabela `prognosis_data_dv<id do dispositivo>_sub<id do paciente>`, composta pelos campos `id`, `timestamp` e `diagnosis`, este último representando o prognóstico gerado.

Em situações que requerem comunicação ativa com o usuário ou profissional de saúde, o sistema aciona o *NotifyAgent*. Este agente interage com a tabela *notify_sub<id>*, criada sob demanda. Essa tabela contém os campos *id*, *timestamp*, *device_id* (chave estrangeira para *devices*) e *diagnosis*, onde são armazenadas as notificações geradas em tempo real para o respectivo sujeito clínico. O modelo entidade-relacionamento do banco de dados descrito é ilustrado na Figura 22.

Figura 22 – Modelo Entidade-Relacionamento (MER) do banco de dados do sistema SMACare, ilustrando as entidades estáticas, as tabelas geradas dinamicamente para cada par dispositivo-paciente e a integração com os módulos responsáveis por armazenamento, diagnóstico, prognóstico e notificações.



Fonte: Elaborado pelo autor (2025).

6.3 REDES NEURAIS ARTIFICIAIS

Como mencionado anteriormente, a presente tese propõe integrar a autonomia proporcionada por SMA com redes neurais artificiais, visando aprimorar a capacidade de inferência de diagnósticos e prognósticos clínicos de forma precisa e adaptativa. Diante desse objetivo,

uma das primeiras questões de pesquisa foi definir qual modelo de redes neurais seria mais adequado para o problema em estudo. Para responder a essa questão, inicialmente foi realizada uma revisão bibliográfica sistemática, com o intuito de mapear e analisar arquiteturas de redes neurais utilizadas em tarefas de classificação e previsão em saúde. A partir dessa análise, foram selecionados cinco modelos representativos, considerando tanto sua capacidade de modelar padrões complexos em dados clínicos quanto sua viabilidade de implementação no contexto distribuído do SMA.

Os modelos escolhidos para testes foram:

- **Multi-Layer Perceptron (MLP)**: uma arquitetura clássica e versátil de rede totalmente conectada;
- **Long Short-Term Memory (LSTM)**: modelo recorrente, capaz de capturar dependências temporais em séries temporais fisiológicas;
- **Wide & Deep (W&D)**: combinação de componentes lineares e não lineares para equilibrar memorizações explícitas e generalização;
- **Residual Networks (ResNet)**: redes profundas com conexões residuais que facilitam o treinamento e permitem maior profundidade;
- **Deep & Cross Network (DCN)**: arquitetura projetada para capturar interações explícitas entre variáveis, útil em dados tabulares heterogêneos.

Para o desenvolvimento e treinamento de todos os modelos, foram utilizados o *framework* TensorFlow e a biblioteca Keras juntamente com a linguagem de programação Python, escolhidas por sua ampla adoção na comunidade científica e, especialmente, pela possibilidade de futuramente permitir a implementação embarcada em dispositivos de baixo consumo, como o Arduino Nano BLE Sense Rev2, por meio do TensorFlow Lite. Essa escolha visa viabilizar, em etapas posteriores do projeto, a execução embarcada de redes neurais para detecção de quedas e outras inferências diretamente no dispositivo, garantindo maior autonomia e redução na dependência de processamento externo.

6.3.1 Análise Comparativa dos Modelos para Integração ao SMA

Com o objetivo de comparar e identificar o modelo mais adequado para utilização no SMA, foram realizados dois experimentos. O primeiro consistiu na seleção de conjuntos de dados provenientes de bases públicas (PhysioNet, 2025) (NHANES, 2025) (Medicine, 2025) e no treinamento de cada modelo escolhido usando as mesmas, com o objetivo de comparar suas respectivas taxas de acurácia de treinamento. As bases de dados públicas utilizadas neste trabalho apresentavam previamente os dados já categorizados em classes, sendo necessário apenas proceder com a tradução dos rótulos para a língua portuguesa a fim de padronizar os conjuntos

para as análises subsequentes. Para cada sinal fisiológico — frequência cardíaca (BPM), pressão arterial estimada (BPT), saturação periférica de oxigênio (SpO_2) e temperatura corporal (Temp) — foi realizado o treinamento com o conjunto completo de modelos selecionados. Também foram conduzidos experimentos específicos para as tarefas de Prognóstico e Diagnóstico Multi-label (diagnóstico integrado, que considera todos os sinais fisiológicos simultaneamente, excetuando-se o Prognóstico). Os resultados obtidos para cada sinal e tarefa estão apresentados na Tabela 2. A Tabela 3 apresenta os resultados de treinamento para prognóstico e diagnóstico multi-label.

Observando a Tabela 2 para o sinal de BPM, observa-se um desempenho elevado e estável em todos os modelos. Os valores médios de acurácia estão consistentemente acima de 0,97, com desvios padrão baixos (o máximo foi 0,0667 para DCN). Todos os modelos atingiram 100% de acurácia no modelo final. Isso sugere que a tarefa de classificação para BPM, com apenas 1000 linhas de dados, é relativamente simples e apresenta alta separabilidade entre classes.

Para o sinal BPT (dataset maior, com 7517 amostras), os resultados também são bons, com médias de acurácia variando entre 0,9774 (ResNet) e 0,9886 (W&D). Apesar de os valores finais de acurácia ficarem ligeiramente abaixo de 100% (entre 98,3% e 99,5%), os desvios padrão foram baixos (todos abaixo de 0,021), refletindo robustez e generalização. W&D e DCN tiveram as maiores médias e os menores desvios, sugerindo que modelos com camadas de cruzamento (*cross-feature*) conseguem modelar bem relações não lineares e interações entre atributos fisiológicos, algo especialmente relevante para pressão arterial, um sinal fisiológico mais complexo.

Os resultados para SpO_2 , embora os modelos finais tenham atingido 100% de acurácia (provavelmente pela adaptação ao conjunto de validação ou *overfitting*), os valores médios durante o treinamento foram mais baixos — por exemplo, MLP teve 0,6736 ($\sigma=0,2887$) e DCN 0,788 ($\sigma=0,23$).

Para o sinal de Temperatura, os valores médios foram os mais baixos do conjunto: MLP com apenas 0,564 e LSTM com 0,636, embora o ResNet e o DCN tenham alcançado médias acima de 0,82. Os desvios padrão foram elevados em todos os casos (entre 0,21 e 0,26), indicando instabilidade no treinamento. A acurácia final de 100% em alguns modelos contrasta com as médias baixas, sugerindo *overfitting* ao conjunto de validação. Isso pode ser explicado por vários fatores: baixa variabilidade real do sinal (temperatura corporal é menos dinâmica), possível falta de características discriminantes suficientes para permitir classificação robusta, e tamanho de amostra limitado. Os valores maiores em DCN e ResNet indicam que modelos com maior capacidade de modelar interações e capturar padrões não triviais conseguem extrair alguma estrutura útil, mas os desvios elevados sugerem necessidade de regularização adicional ou dados mais amplos e balanceados.

Analizando a Tabela 3 pode-se notar que os resultados sugerem que a tarefa de diagnóstico multi-label é desafiadora, mas solucionável. O desvio padrão relativamente elevado em algumas arquiteturas (especialmente W&D) pode refletir dificuldades na modelagem de dependências inter-label ou sensibilidade a hiperparâmetros. Notavelmente, mesmo modelos relativamente

simples (MLP, DCN) apresentaram médias acima de 0,86, com desvios mais altos que ResNet, indicando que mesmo essas abordagens conseguem aprender parte importante da estrutura dos dados.

Nos modelos para prognóstico, o resultado excepcionalmente ruim do LSTM sugere algum problema estrutural ou conceitual no uso dessa arquitetura para a tarefa específica. Isso pode indicar inadequação da estrutura sequencial para os dados fornecidos, caso as entradas não contenham dependências temporais relevantes ou tenham sido mal preparadas para séries temporais, além de possíveis problemas de hiperparâmetros ou de convergência ao longo das 200 épocas, levando a um *underfitting* severo. Também não se pode descartar um potencial erro de implementação ou de formatação dos dados para o modelo. Por outro lado, o excelente desempenho dos demais modelos sugere que a tarefa de prognóstico é bem capturada por arquiteturas densas, com *cross-feature* (como W&D e DCN), ou por redes convolucionais profundas (ResNet). Os desvios padrão baixos, em torno de 0,02, indicam alta estabilidade entre execuções e sugerem que se trata de um problema de classificação mais bem definido do que o diagnóstico multi-label.

Tabela 2 – Resumo dos valores médios de acurácia e dos desvios padrão observados no treinamento dos modelos de classificação para os sinais fisiológicos.

| Sinal | Dataset (linhas) | Modelo | Épocas | Acurácia Média | Desvio Padrão | Acurácia Final (%) |
|------------------------|------------------|----------------|--------|----------------|---------------|--------------------|
| BPM | 1000 | MLP | 50 | 0,9759 | 0,0343 | 100,0% |
| | | LSTM | 50 | 0,9711 | 0,0607 | 100,0% |
| | | W&D | 50 | 0,9737 | 0,0651 | 100,0% |
| | | ResNet | 50 | 0,9879 | 0,0106 | 100,0% |
| | | DCN | 50 | 0,9723 | 0,0667 | 100,0% |
| BPT | 7517 | MLP | 50 | 0,9793 | 0,0157 | 99,5% |
| | | LSTM | 50 | 0,9783 | 0,0208 | 98,3% |
| | | W&D | 50 | 0,9886 | 0,0146 | 99,1% |
| | | ResNet | 50 | 0,9774 | 0,0159 | 99,5% |
| | | DCN | 50 | 0,9875 | 0,0137 | 99,4% |
| SpO₂ | 1000 | MLP | 80 | 0,6736 | 0,2887 | 100,0% |
| | | LSTM | 80 | 0,7471 | 0,2583 | 100,0% |
| | | W&D | 80 | 0,8201 | 0,2468 | 100,0% |
| | | ResNet | 80 | 0,9401 | 0,1248 | 100,0% |
| | | DCN | 80 | 0,7880 | 0,2300 | 100,0% |
| Temp | 1000 | MLP | 50 | 0,564 | 0,261 | 97,5% |
| | | LSTM | 50 | 0,636 | 0,239 | 95,0% |
| | | W&D | 50 | 0,695 | 0,254 | 100,0% |
| | | ResNet | 50 | 0,820 | 0,210 | 100,0% |
| | | DCN | 50 | 0,840 | 0,230 | 100,0% |

Fonte: Elaborado pela autora (2025).

O segundo experimento também fez uso dos datasets de bases públicas e teve como objetivo comparar o desempenho dos modelos em termos de taxa de acertos e avaliar o tempo médio de inferência e de resposta ao SMA. Essa etapa buscou analisar não apenas a precisão

Tabela 3 – Resumo dos valores médios de acurácia e dos desvios padrão observados no treinamento dos modelos de classificação para diagnóstico multi-label e prognóstico.

| Tipo | Modelo | Épocas | Acurácia Média | Desvio Padrão | Acurácia Final (%) |
|-------------------------|----------------|--------|----------------|---------------|--------------------|
| Diagnóstico Multi-Label | MLP | 100 | 0,861 | 0,060 | 91,62% |
| | LSTM | 100 | 0,830 | 0,140 | 92,15% |
| | W&D | 100 | 0,750 | 0,250 | 91,62% |
| | ResNet | 100 | 0,860 | 0,070 | 94,24% |
| | DCN | 100 | 0,860 | 0,140 | 93,46% |
| Prognóstico | MLP | 200 | 0,903 | 0,021 | 92,50% |
| | LSTM | 200 | 0,0178 | 0,0116 | 1,89% |
| | W&D | 200 | 0,902 | 0,021 | 92,50% |
| | ResNet | 200 | 0,902 | 0,022 | 92,50% |
| | DCN | 200 | 0,884 | 0,038 | 91,25% |

Fonte: Elaborado pela autora (2025).

das previsões, mas também a eficiência computacional, fator essencial para a integração em um ambiente distribuído e potencialmente embarcado.

Para a etapa de avaliação, foram realizadas 100 iterações utilizando um conjunto específico de dados de um único sujeito. Os resultados obtidos estão apresentados na Tabela 4, a qual detalha o tempo médio de inferência (em segundos) ao longo das cem iterações, o desvio padrão dos tempos registrados e a taxa de acertos no diagnóstico. Nas Figuras 23 a 28, são apresentados os gráficos que ilustram o tempo médio de inferência por iteração e a acurácia obtida por modelo de inteligência artificial. Nesses gráficos, as barras azuis representam o tempo médio de inferência (em segundos) para cada modelo, enquanto as barras pretas, posicionadas acima e abaixo de cada barra azul, indicam o desvio padrão ou o intervalo de confiança da média. Assim, quanto maior a extensão da barra de erro, maior é a variabilidade observada nos tempos de execução entre as iterações. As barras vermelhas representam a acurácia média (em %) da inferência modelo.

Os dados evidenciam que o modelo MLP apresentou o melhor desempenho em todas as variáveis fisiológicas analisadas, destacando-se tanto pela acurácia quanto pela eficiência no tempo de inferência. Complementarmente, a Tabela 5 apresenta a mesma análise comparativa dos modelos para as tarefas de diagnóstico multi-label e prognóstico. Observando a Tabela 4 nota-se acurácia muito elevada para quase todos os sinais e modelos, frequentemente atingindo 100%, acompanhada por tempos médios de inferência relativamente baixos (em torno de 0,09–0,13 segundos). Essa combinação indica que os modelos são eficazes e viáveis para aplicação prática em cenários de inferência em tempo real ou quase real, como monitoramento fisiológico embarcado.

Para BPM (Figura 23), todos os modelos atingiram acurácia de 100%, com exceção do W&D, que teve 99,5%. O tempo médio de inferência foi muito consistente entre modelos (aprox. 0,0969–0,1118 s), com desvios padrão baixos (0,01–0,02 s). Estes resultados sugerem

que a tarefa de classificação para BPM é relativamente simples e bem resolvida por todas as arquiteturas, mesmo aquelas menos complexas (MLP). A pequena diferença no tempo de inferência entre LSTM (um pouco mais alto) e os outros modelos reflete o custo adicional do processamento sequencial, ainda que pequeno.

Analizando os resultados de BPT (Figura 24), todos os modelos atingiram 100% de acurácia. Os tempos médios de inferência foram ligeiramente mais altos (0,1084–0,1251 s), com desvios padrão igualmente baixos (0,01–0,03 s). Isso mostra que mesmo com maior volume de dados, os modelos mantiveram alta precisão e consistência computacional, demonstrando boa escalabilidade para inferência. Modelos como W&D e ResNet apresentaram tempos ligeiramente maiores, compatíveis com suas arquiteturas mais profundas e capacidade de capturar interações complexas.

Em SpO₂ (Figura 25), todos os modelos atingiram 100% de acurácia, exceto W&D, que ficou em 99,7%. Tempos de inferência variaram de 0,0960 a 0,1257 s, com o LSTM novamente apresentando o maior tempo médio (0,1257 s) e o maior desvio padrão (0,0457 s). O desempenho consistente em acurácia sugere que a tarefa foi bem modelada, porém o aumento no desvio padrão para LSTM pode refletir variação na estrutura sequencial durante a inferência, algo a se considerar para aplicações em hardware limitado.

Para Temperatura (Figura 26), a acurácia caiu ligeiramente em comparação com outros sinais: MLP e LSTM atingiram 100%, mas W&D e ResNet ficaram em 98,1%, e DCN em 99,6%. O tempo de inferência manteve-se baixo (0,099–0,106 s), com desvios padrão baixos (0,011–0,029 s). Essa leve queda de acurácia para alguns modelos sugere que a classificação de temperatura pode ser um problema um pouco mais difícil ou menos separável, mesmo que os tempos de inferência permaneçam adequados para aplicações práticas.

Analizando a Tabela 5 observa-se que para a tarefa de diagnóstico multi-label modelos como MLP, ResNet e DCN conseguiram prever corretamente o diagnóstico *Hipotermia* com 100% de acerto, apresentando tempos médios baixos (em torno de 0,067 segundos) e desvios padrão igualmente reduzidos (0,0048 – 0,0053 s). Esses resultados indicam altíssima eficiência e estabilidade desses modelos para a tarefa, tornando-os adequados para aplicações em tempo real, como sistemas de triagem automática ou monitoramento contínuo em ambientes clínicos.

Por outro lado, LSTM e W&D falharam completamente, com 0% de acerto, classificando incorretamente os casos como *Normal* em todos os testes. O LSTM apresentou um tempo médio de inferência muito mais alto (0,538 s), sem variabilidade (provavelmente resultado de uma inferência determinística ou mal configurada). Esse tempo muito superior, aliado à falta de acerto, indica problemas de adaptação desse modelo à tarefa, possivelmente relacionados a uma inadequação entre os dados de entrada e a estrutura sequencial do LSTM, ou ainda dificuldades no treinamento e convergência. A falha do W&D, mesmo com tempo de inferência baixo (0,0715 s), sugere incapacidade de capturar corretamente as interações de atributos relevantes para essa tarefa específica, possivelmente por limitação na modelagem das relações entre os sinais fisiológicos que caracterizam a classe-alvo *Hipotermia*. Para diagnóstico multi-label, a

Tabela 4 – Resumo das métricas de acurácia e do tempo médio de inferência por modelo e sinal fisiológico.

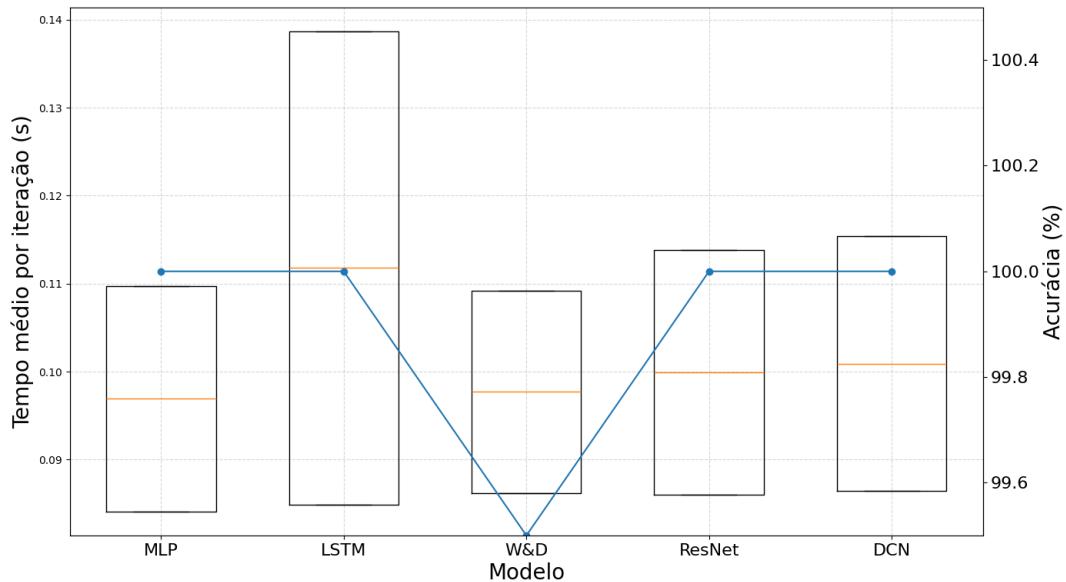
| Sinal | Dataset (linhas) | Modelo | Tempo Médio (s) | Desvio Padrão (s) | Acertos (%) |
|------------------|------------------|--------|-----------------|-------------------|-------------|
| BPM | 1000 | MLP | 0,0969 | 0,0128 | 100 |
| | | LSTM | 0,1118 | 0,0269 | 100 |
| | | W&D | 0,0977 | 0,0115 | 99,5 |
| | | ResNet | 0,0999 | 0,0139 | 100 |
| | | DCN | 0,1009 | 0,0145 | 100 |
| BPT | 1000 | MLP | 0,1084 | 0,0121 | 100 |
| | | LSTM | 0,1135 | 0,0135 | 100 |
| | | W&D | 0,1251 | 0,0341 | 100 |
| | | ResNet | 0,1227 | 0,0291 | 100 |
| | | DCN | 0,1132 | 0,0233 | 100 |
| SpO ₂ | 1000 | MLP | 0,0960 | 0,0095 | 100 |
| | | LSTM | 0,1257 | 0,0457 | 100 |
| | | W&D | 0,1032 | 0,0105 | 99,7 |
| | | ResNet | 0,0979 | 0,0130 | 100 |
| | | DCN | 0,0995 | 0,0158 | 100 |
| Temp | 1000 | MLP | 0,099 | 0,011 | 100 |
| | | LSTM | 0,106 | 0,029 | 100 |
| | | W&D | 0,100 | 0,015 | 98,1 |
| | | ResNet | 0,100 | 0,012 | 98,1 |
| | | DCN | 0,103 | 0,019 | 99,6 |

Fonte: Elaborado pela autora (2025).

tabela evidencia vantagem clara de modelos mais *tradicionais* e bem interpretáveis (MLP) e de arquiteturas profundas com conexões residuais (ResNet, DCN). Esses modelos não apenas alcançam máxima acurácia, mas o fazem de forma rápida e consistente.

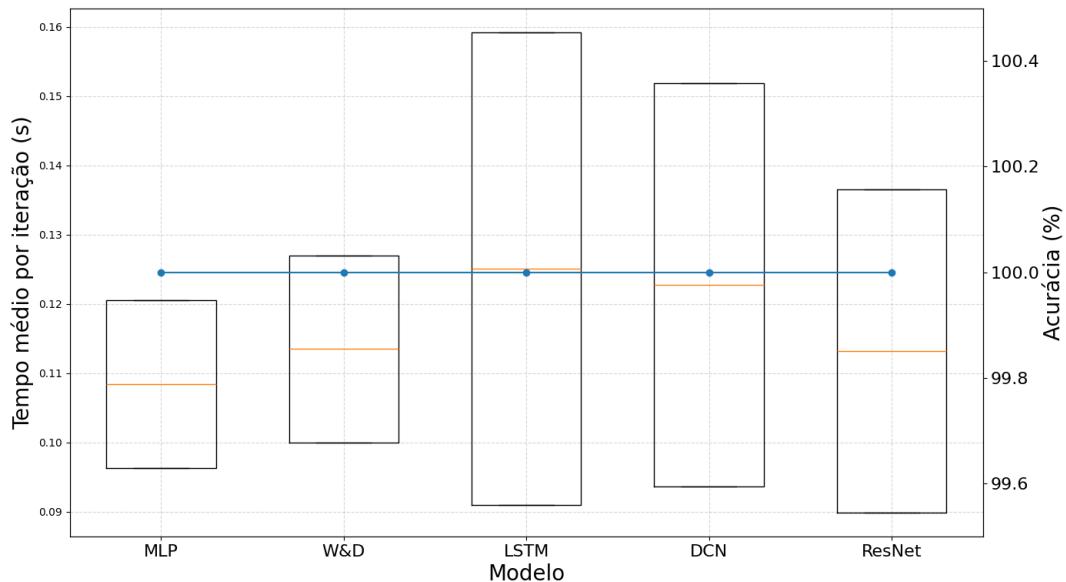
Considerando a tarefa de prognóstico, o padrão geral é semelhante em termos de divisão entre modelos bem-sucedidos e malsucedidos, embora as classes-alvo mudem. MLP, ResNet e DCN classificaram corretamente *Hipotermia/Desnutrição Severa* com 100% de acerto, com tempos médios de inferência modestos (0,081 – 0,085 s) e desvios padrão relativamente baixos (0,011 – 0,023 s). A leve elevação nos tempos, em comparação ao diagnóstico, é natural dado o aumento potencial da complexidade da tarefa ou do processamento associado a múltiplas classes prognósticas. Novamente, os resultados confirmam a robustez e consistência desses modelos, destacando-os como opções confiáveis para aplicações em sistemas de apoio à decisão clínica. Em contraste, o W&D falhou completamente, prevendo incorretamente *Colapso Metabólico* em todos os casos, resultando em 0% de acerto, apesar de manter um tempo de inferência baixo (0,0812 s). O modelo LSTM, por ter tido uma acurácia baixíssima já no treinamento, não foi sequer testado.

Figura 23 – Tempo médio de inferência e acurácia dos modelos para classificação de frequência cardíaca.



Fonte: Elaborado pela autora (2025).

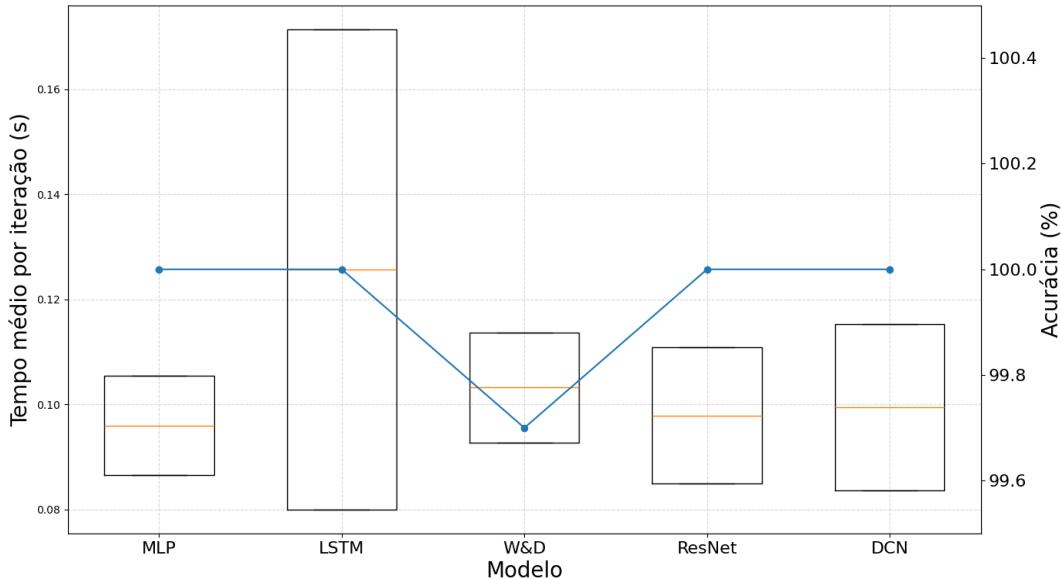
Figura 24 – Tempo médio de inferência e acurácia dos modelos para classificação de pressão arterial estimada.



Fonte: Elaborado pela autora (2025).

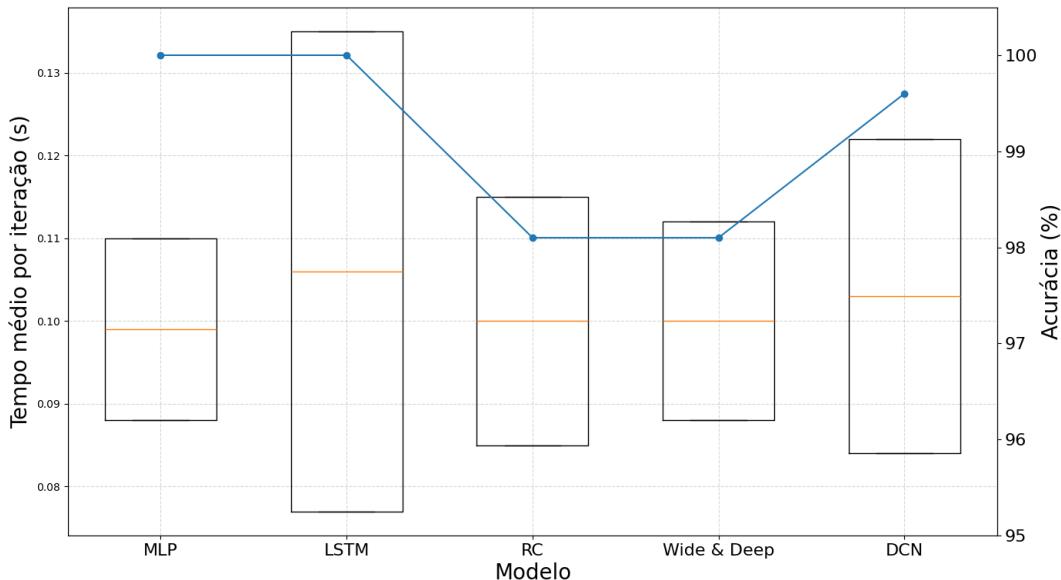
Os resultados apresentados nas tabelas e suas discussões demonstram de forma consistente que o MLP é uma escolha eficaz e pragmática para a tarefa de classificação de sinais fisiológicos em sistemas de monitoramento automático. Em primeiro lugar, o MLP apresentou acurárias elevadas em quase todos os sinais, frequentemente alcançando ou se aproximando de 100% de acerto, mesmo em datasets de diferentes tamanhos e complexidades, como BPM (1000 amostras) e BPT (7517 amostras). Essa performance indica que a arquitetura MLP é plenamente

Figura 25 – Tempo médio de inferência e acurácia dos modelos para classificação de SpO2.



Fonte: Elaborado pela autora (2025).

Figura 26 – Tempo médio de inferência e acurácia dos modelos para classificação de temperatura.



Fonte: Elaborado pela autora (2025).

capaz de capturar os padrões relevantes nos dados, mesmo sem mecanismos mais sofisticados como convoluções ou conexões residuais.

Outro ponto de destaque para o MLP é sua consistência no tempo de inferência. Em todas as tarefas analisadas, o tempo médio por inferência do MLP permaneceu baixo (em torno de 0,096 – 0,108 s), com desvios padrão pequenos. Essa característica o torna particularmente vantajoso para aplicações embarcadas e em tempo real, como dispositivos vestíveis ou sistemas móveis, em que recursos computacionais e consumo de energia são limitados. A relativa simplicidade

arquitetural do MLP garante menor carga computacional, maior previsibilidade do desempenho em diferentes plataformas e facilidade de implementação em hardware menos potente.

Outro argumento importante reside na estabilidade do MLP entre execuções. Apesar de não ter sempre a menor variabilidade, o MLP demonstrou desvios padrão aceitáveis em todas as métricas avaliadas, indicando boa capacidade de generalização mesmo sem ajustes excessivamente finos de hiperparâmetros. Para aplicações práticas em saúde, essa característica é fundamental: modelos que exigem calibração complexa ou que apresentam instabilidade entre execuções podem gerar incertezas clínicas inaceitáveis.

Em tarefas mais desafiadoras — como diagnóstico multi-label (Figura 28) e prognóstico (Figura 27) — o MLP continuou competitivo. No diagnóstico multi-label, atingiu média de acurácia de 0,861 com desvio padrão moderado (0,06), e no prognóstico foi um dos melhores modelos, com média de 0,9 e desvio padrão igualmente baixo (0,021). Esses resultados mostram que, mesmo em cenários mais complexos, o MLP consegue manter um equilíbrio excelente entre desempenho preditivo e estabilidade computacional. Além do desempenho empírico, há vantagens adicionais em transparência e interpretabilidade.

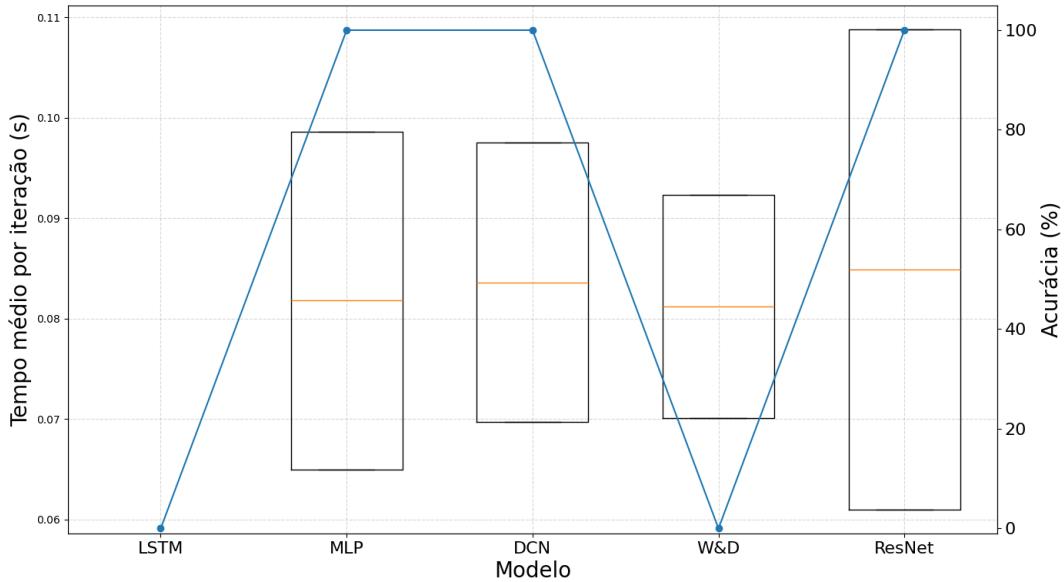
Do ponto de vista de desenvolvimento e manutenção de sistemas, o MLP oferece baixo custo de treinamento e ajuste, menor susceptibilidade a *overfitting* (em comparação com arquiteturas mais profundas) e facilidade de reutilização em diferentes tarefas com ajustes mínimos. Dessa forma, o modelo escolhido para as redes neurais do SMACare foi o modelo MLP, para todos os sinais fisiológicos e para prognóstico e diagnóstico multi-label.

Tabela 5 – Resultados comparativos de tempo de inferência, variabilidade e acurácia dos modelos de diagnóstico e prognóstico.

| Tipo | Modelo | Resultado | Tempo Médio (s) | Desvio Padrão (s) | Acertos (%) |
|--------------------------------|----------------|--------------------------------|------------------------|--------------------------|--------------------|
| Diagnóstico Multi-Label | MLP | Hipotermia | 0,067210 | 0,004810 | 100 |
| | LSTM | Normal | 0,538564 | 0,000000 | 0 |
| | W&D | Normal | 0,071538 | 0,006799 | 0 |
| | ResNet | Hipotermia | 0,067610 | 0,005327 | 100 |
| | DCN | Hipotermia | 0,067451 | 0,004994 | 100 |
| Prognóstico | MLP | Hipotermia/-Desnutrição Severa | 0,0818 | 0,0168 | 100 |
| | LSTM | Não testado | – | – | – |
| | W&D | Colapso Metabólico | 0,0812 | 0,0111 | 0 |
| | ResNet | Hipotermia/-Desnutrição Severa | 0,0849 | 0,0239 | 100 |
| | DCN | Hipotermia/-Desnutrição Severa | 0,0836 | 0,0139 | 100 |

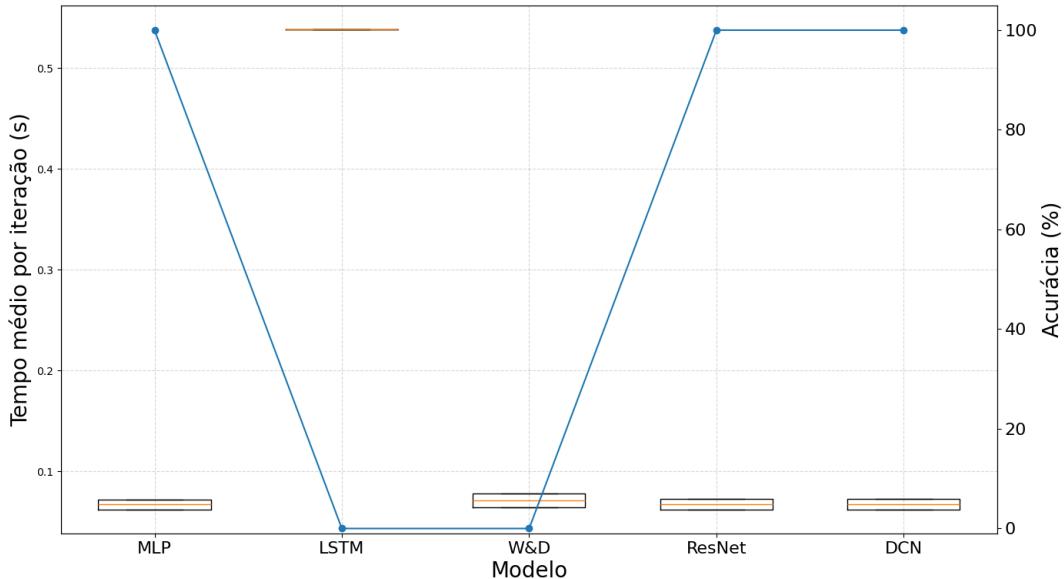
Fonte: Elaborado pela autora (2025).

Figura 27 – Tempo médio de inferência e acurácia dos modelos para classificação de prognóstico.



Fonte: Elaborado pela autora (2025).

Figura 28 – Tempo médio de inferência e acurácia dos modelos para classificação de diagnóstico.



Fonte: Elaborado pela autora (2025).

6.4 ARQUITETURA DO SMACARE

Segundo HAYKIN (2009), redes densas multicamadas constituem o alicerce do MLP, cuja capacidade de aproximação universal é garantida para funções contínuas, desde que haja um número suficiente de unidades ocultas e uma função de ativação não linear adequada. Outro aspecto como descrito por (Mehlig, 2021), é o aumento da profundidade. Este permite que a

rede aprenda representações hierárquicas dos dados, favorecendo a decomposição progressiva de padrões complexos em níveis crescentes de abstração. Em consonância com esse princípio, camadas de normalização de lote (*batch normalization*) e técnicas de regularização como *dropout* foram incorporadas para mitigar problemas clássicos de redes profundas, como o *vanishing gradient* e o sobreajuste (*overfitting*), promovendo maior estabilidade e capacidade de generalização. A escolha de redes densas e, quando apropriado, profundas, proporcionou ao SMA a flexibilidade necessária para modelar variações fisiológicas complexas, alinhando-se às melhores práticas descritas na literatura contemporânea de aprendizado de máquina supervisionado. Os modelos utilizados neste trabalho consistem em três componentes principais:

6.4.1 Camadas e Transformações

- **Camada de Entrada:** Recebe n variáveis médicas (e.g., PAS, PAD, pulso).
- **Camadas Ocultas:** Aplicam transformações não lineares via Equação 4:

$$\mathbf{a}^{(l)} = \sigma \left(\mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \right) \quad (4)$$

onde σ é a função de ativação ReLU ($\sigma(z) = \max(0, z)$).

- **Camada de Saída:** Gera probabilidades usando Softmax (Equação 5):

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}, \quad i = 1, \dots, K \quad (5)$$

6.4.2 Pré-Processamento de Dados

Os dados são normalizados para média zero e variância unitária antes da inferência (Equação 6):

$$\mathbf{x}_{\text{scaled}} = \frac{\mathbf{x} - \mu}{\sigma} \quad (6)$$

onde μ e σ são parâmetros armazenados em arquivos `scaler_*.pkl`, como ilustrado na Figura 6.1.

Figura 6.1 – Trecho de código fonte em python para classificação com base em modelo pré treinado de diagnóstico de SpO2

```

1 # Dados de entrada
2 input_data = np.array([[spo2_value]])
3 # Aplica a normalização
4 input_scaled = scaler.transform(input_data)

```

6.4.3 Treinamento do MLP

Os modelos foram treinados sob o paradigma de aprendizado supervisionado, isto é, ajustando pesos a partir de exemplos de entrada-saída conhecidos. Todos eles utilizaram a função de perda entropia cruzada categórica esparsa (*sparse categorical crossentropy*) – adequada à saída softmax – e o otimizador Adam (Kingma; Ba, 2015) para a atualização dos pesos. O uso de entropia cruzada configura o treinamento como um problema de maximização de verossimilhança das classes corretas, alinhado às práticas estatísticas clássicas de aprendizado de máquinas. O Adam, por ser um método de descida de gradiente estocástico adaptativo, acelera a convergência e é robusto quanto à escolha da taxa de aprendizado.

Durante o treinamento, a preocupação com *overfitting* (sobreajuste) e *underfitting* (subajuste) se fez presente. O *overfitting* ocorre quando a rede memoriza padrões do conjunto de treino, mas perde a capacidade de generalização, enquanto *underfitting* indica que a rede não possui complexidade suficiente para capturar a relação subjacente nos dados. Nos modelos utilizados nesse trabalho, o risco de sobreajuste foi endereçado no modelo de prognóstico, que inclui camadas *dropout* com taxa de 35%. O *dropout* é uma técnica de regularização que desativa aleatoriamente uma fração dos neurônios durante cada iteração de treinamento, forçando a rede a aprender representações mais robustas e reduzindo a dependência excessiva de determinados caminhos de ativação. Isso equivale, aproximadamente, a treinar uma grande coleção de sub-redes e fazer uma média de seus comportamentos, o que tende a melhorar a capacidade de generalização do modelo.

Os demais modelos (sem *dropout*) possuem arquiteturas mais simples e com menos parâmetros, o que por si só atenua o risco de sobreajuste – embora também limite a capacidade de representação. Para evitar *underfitting*, as configurações escolhidas (por exemplo, duas camadas ocultas em vez de apenas uma) conferem às redes flexibilidade suficiente para modelar relações não lineares presentes nos dados de entrada. Em suma, o processo de treinamento empregado buscou otimizar o desempenho na predição das classes corretas ao mesmo tempo em que incorporou medidas para equilibrar a complexidade do modelo e sua capacidade de generalização.

Para o treinamento dos modelos utilizados para inferência, as seguintes etapas foram seguidas:

Função de Perda e Otimização:

- **Entropia Cruzada (Equação 7):**

$$\mathcal{L} = - \sum_{i=1}^N \sum_{j=1}^K y_{ij} \log(\hat{y}_{ij}) \quad (7)$$

- **Atualização de Pesos:** Via gradiente descendente com **Adam** (Equações 8 e 23):

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(t)}} \quad (8)$$

$$\mathbf{b}^{(t+1)} = \mathbf{b}^{(t)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(t)}} \quad (9)$$

Inferência nos Modelos:

A predição segue três passos:

1. Carregamento do Modelo (Figura 6.2):

```
1 model = keras.models.load_model("blood_pressure_model.h5",
    compile=False)
```

Figura 6.2 – Trecho de código fonte em python para carregamento do modelo pré-treinado para classificação de pressão arterial.

2. Propagação Direta (Figura 6.3):

```
1 prediction = model.predict(input_scaled, verbose=0)
```

Figura 6.3 – Trecho de código fonte em python para propagação direta.

3. Decodificação (Equação 10):

$$\text{Classe} = \text{argmax}(\hat{\mathbf{y}}) \quad (10)$$

Classificação de Pressão Arterial:

- **Entrada**, dada pela Equação 11:

$$\mathbf{x} = [\text{PAS}, \text{PAD}, \text{BPM}] \quad (11)$$

Sendo: PAS: pressão arterial sistólica; PAD: pressão arterial diastólica; BPM: frequência cardíaca.

- **Saída**, dada pela Equação 12:

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \quad (12)$$

Prognóstico com Dados Temporais:

- **Agregação Temporal**, dada pela Equação 13:

$$\text{Diagnóstico Final} = \text{mode}(\{\hat{y}_1, \dots, \hat{y}_{1000}\}) \quad (13)$$

Classificação de SpO₂:

- **Entrada Unidimensional**, dada pela Equação 14:

$$x = \text{SpO}_2 \quad (14)$$

- **Normalização**, dada pela Equação 15:

$$x_{\text{scaled}} = \frac{x - \mu_{\text{SpO}_2}}{\sigma_{\text{SpO}_2}} \quad (15)$$

As classificações de temperatura e frequência cardíaca seguem o mesmo molde da classificação de SpO₂. O pseudocódigo base de treinamento das redes neurais e geração de modelo pode ser encontrado no Apêndice A.1.

6.5 ARQUITETURA DE REDES NEURAIS PARA PRESSÃO ARTERIAL ESTIMADA

Para o diagnóstico de pressão arterial estimada foi utilizada a arquitetura de Rede Neural *Feedforward* (também conhecida como *Multilayer Perceptron* — MLP) sendo composta por três camadas densamente conectadas: duas camadas ocultas e uma camada de saída. Cada uma das camadas Dense implementa conexões totalmente conectadas com seus neurônios adjacentes, e, portanto, a quantidade total de parâmetros treináveis pode ser calculada a partir da expressão dada na Equação 16:

$$P = (n \cdot h) + b \quad (16)$$

Onde: P = número total de parâmetros, n = número de entradas, h = número de neurônios, b = número de termos de bias.

A definição da arquitetura da rede neural utilizada para classificação de pressão arterial estimada foi fundamentada em princípios teóricos, heurísticas clássicas de projeto de redes e restrições computacionais de um futuro ambiente embarcado. A tarefa consiste em classificar pacientes nas três categorias clínicas. As entradas consistem de três sinais fisiológicos (Equação 17):

$$\mathbf{x} = [\text{bpt_sys}, \text{bpt_dia}, \text{bpt_bpm}] \in \mathbb{R}^3 \quad (17)$$

A escolha por uma rede rasa, com apenas duas camadas ocultas, está relacionada à complexidade moderada do mapeamento não linear entre os sinais fisiológicos e as classes clínicas. Embora o Teorema Universal da Aproximação afirme que uma única camada oculta pode aproximar qualquer função contínua definida em um subconjunto compacto de \mathbb{R}^n , essa abordagem apresenta limitações práticas:

- Exige número elevado de neurônios em uma única camada;
- Dificulta a extração hierárquica de características;
- Aumenta o risco de sobreajuste e instabilidade.

Optou-se, assim, por uma rede com duas camadas ocultas, permitindo o balanceamento entre capacidade de aprendizado, simplicidade estrutural e viabilidade computacional. A definição do número de neurônios por camada foi baseada em heurísticas clássicas e estratégias de compressão progressiva. Utilizou-se a heurística representada pelas Equações 18 e 19 para definição de quantidade de neurônios da primeira camada.

$$n_1 = \alpha \cdot (n_{\text{entrada}} + n_{\text{sada}}) \quad (18)$$

Com $n_{\text{entrada}} = 3$, $n_{\text{sada}} = 3$ e $\alpha = 2$:

$$n_1 = 2 \cdot (3 + 3) = 12 \quad (19)$$

Adotou-se o valor arredondado para cima (Equação 20):

$$n_1 = 16 \quad (20)$$

Para a segunda camada oculta, seguindo uma estratégia de compressão progressiva (Equação 21):

$$n_2 = \frac{n_1}{2} = \frac{16}{2} = 8 \quad (21)$$

Essa compressão força a rede a condensar a informação, eliminando redundâncias e promovendo uma representação mais discriminativa. Portanto, a estrutura da rede ficou a seguinte: A primeira camada oculta da arquitetura implementada é composta por 16 neurônios, cada um utilizando a função de ativação ReLU (*Rectified Linear Unit*). Diferentemente das funções tradicionais como a sigmoide ou a tangente hiperbólica, a ReLU não sofre saturação na região positiva do domínio, o que contribui para a mitigação do problema de gradientes desvanecentes (*vanishing gradient*) durante o treinamento. Apesar de sua simplicidade computacional, a ReLU é altamente eficaz para introduzir não linearidade ao modelo, permitindo que a rede aprenda relações complexas entre os sinais biomédicos.

Esta primeira camada recebe como entrada um vetor composto por três variáveis fisiológicas previamente normalizadas: pressão sistólica (bpt_sys), pressão diastólica (bpt_dia) e frequência cardíaca (bpt_bpm). Cada um dos 16 neurônios conecta-se às três entradas, além de possuir um viés (*bias*) individual. Assim, tem-se o número total de 64 parâmetros treináveis nesta camada. Estes parâmetros são ajustados durante o processo de treinamento da rede, permitindo que a camada aprenda representações úteis para as etapas subsequentes do modelo.

A segunda camada oculta da rede é composta por 8 neurônios densamente conectados, cuja função é refinar as representações extraídas pela primeira camada organizando os dados de

forma mais clara e eficiente, preparando-os para que a camada de saída (com Softmax) possa fazer uma classificação precisa. Cada um desses neurônios recebe como entrada os 16 valores ativados provenientes da camada anterior, aplicando novamente a função de ativação ReLU para garantir a continuidade da não linearidade ao longo da rede. Além das conexões sinápticas, cada neurônio desta camada também possui um viés associado, resultando em 136 parâmetros treináveis.

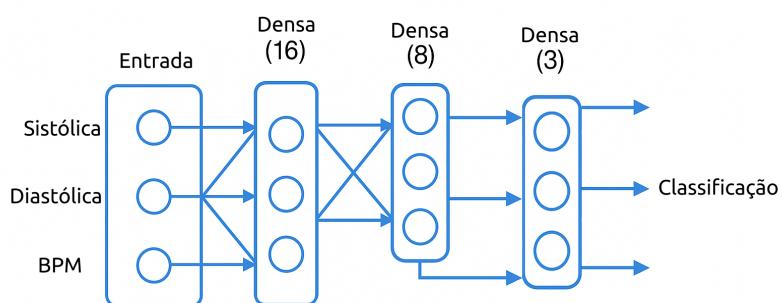
A camada de saída do modelo é composta por 3 neurônios, os quais correspondem às classes-alvo do problema de classificação:

- Hipotensão;
- Normal;
- Hipertensão.

Cada neurônio recebe como entrada as 8 ativações produzidas pela segunda camada oculta, além de possuir um viés associado. Dessa forma, o número total de parâmetros treináveis nesta camada é de 27 parâmetros. Somando os parâmetros de todas as camadas densas da rede, obtém-se 227 parâmetros treináveis. Esse número relativamente baixo de parâmetros reflete a leveza computacional do modelo; essa arquitetura compacta possibilita a execução em tempo real com baixa latência, mantendo, ao mesmo tempo, uma capacidade discriminativa adequada para contextos clínicos básicos.

A função de ativação utilizada na camada de saída é a Softmax, apropriada para tarefas de classificação multiclasse. Essa função transforma os valores de ativação da camada final em uma distribuição de probabilidade, de modo que a soma das saídas seja igual a 1. Isso permite interpretar a saída do modelo como a confiança atribuída a cada uma das três classes clínicas, sendo a classe com maior probabilidade selecionada como predição final. A Figura 29 ilustra a arquitetura completa da rede neural implementada, composta por três camadas densas:

Figura 29 – Diagrama da Rede Neural Feedforward (MLP) para o sinal de pressão arterial estimada (BPT), com três camadas densas: 16, 8 e 3 neurônios, respectivamente.



Fonte: Elaborado pela autora (2025)

- **Entrada:** vetor com os sinais fisiológicos adquiridos pelo sensor BPT: pressão sistólica (bpt_sys), pressão diastólica (bpt_dia) e frequência cardíaca (bpt_bpm);
- **Camada 1 — Dense(16):** aplica a função de ativação ReLU sobre as três variáveis de entrada;
- **Camada 2 — Dense(8):** reduz a dimensionalidade da representação interna, também utilizando ReLU;
- **Camada 3 — Dense(3):** realiza a classificação em uma das três categorias clínicas por meio da função Softmax.

6.6 ARQUITETURA DE REDES NEURAIS PARA SATURAÇÃO PERIFÉRICA DE OXIGÊNIO

O modelo de classificação de saturação periférica de oxigênio (SpO_2) foi implementado como uma rede neural MLP, composta por duas camadas ocultas densas com 8 e 4 neurônios, respectivamente, e uma camada de saída com 6 neurônios, que representam categorias clínicas distintas (Figura 30). A definição desta arquitetura foi orientada por heurísticas clássicas, características da entrada univariada e pela necessidade de execução eficiente em ambiente embarcado. A entrada do modelo consiste em um único valor contínuo, dado pela Equação 22:

$$x = \text{SpO}_2 \in \mathbb{R} \quad (22)$$

A tarefa de classificação clínica define um mapeamento da forma dada pela Equação 23:

$$f : \mathbb{R} \rightarrow \mathbb{R}^6 \quad (23)$$

Trata-se de um problema de classificação multiclasse com baixa dimensionalidade de entrada, porém com separação não linear entre as classes. Embora o Teorema Universal da Aproximação afirme que uma única camada oculta é suficiente para aproximar qualquer função contínua, estudos como os de (Hecht-Nielsen, 1989) e (Goodfellow; Bengio; Courville, 2016) demonstram que redes com múltiplas camadas ocultas podem alcançar melhor desempenho e generalização, com menos neurônios por camada e menor risco de sobreajuste.

Optou-se por uma arquitetura com duas camadas ocultas, o que permite:

- Representação hierárquica de características;
- Compressão progressiva da informação;
- Maior estabilidade no treinamento;

- Compatibilidade com hardware embarcado.

Para a definição da quantidade de neurônios a ser utilizada por camada, utilizou-se a heurística clássica (Equação 24):

$$n_1 = \alpha \cdot (n_{\text{entrada}} + n_{\text{saída}}) \quad (24)$$

Com $n_{\text{entrada}} = 1$, $n_{\text{saída}} = 6$ e $\alpha = 1.0$ (Equação 25):

$$n_1 = 1 \cdot (1 + 6) = 7 \text{ (Camada 1)} \quad (25)$$

Arredondou-se para $n_1 = 8$ (potência de 2, melhor alinhamento de memória). Para a camada 2, seguiu-se uma estratégia de compressão progressiva dada pela Equação 26:

$$n_2 = \frac{n_1}{2} = \frac{8}{2} = 4 \quad (26)$$

Essa camada refina as representações extraídas da camada anterior, facilitando a separação das classes clínicas.

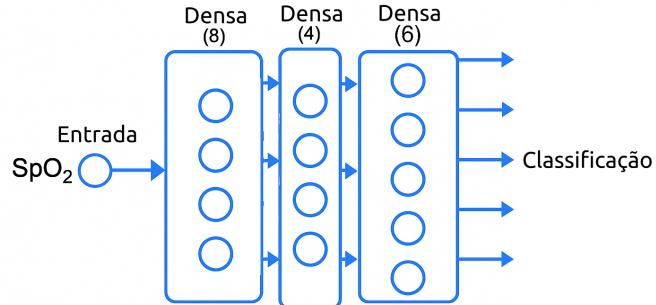
A saída contém 6 neurônios, correspondentes às categorias clínicas de oxigenação:

- Baixo nível de saturação
- Hipoxemia
- Hipoxemia grave
- Nível anormal de saturação
- Nível médio de saturação
- Nível bom de saturação

Descrevendo portanto a arquitetura completa: A primeira camada recebe como entrada uma única variável contínua normalizada: o valor de SpO₂. Essa entrada é processada por 8 neurônios com função de ativação ReLU, resultando em 16 parâmetros treináveis. Essa camada introduz não linearidade ao modelo, permitindo a extração de características mesmo em dados de baixa dimensionalidade.

A segunda camada oculta refina as ativações anteriores, reduzindo a dimensionalidade para 4 neurônios. Cada um desses neurônios conecta-se às 8 saídas da camada anterior, resultando em 36 parâmetros treináveis. Essa projeção intermediária contribui para a separabilidade das classes no espaço de decisão da rede.

Figura 30 – Diagrama da Rede Neural Feedforward (MLP) para classificação de SpO₂ com camadas Dense(8) e Dense(4).



Fonte: Elaborado pela autora (2025)

A camada de saída possui 6 neurônios, cada um representando uma categoria clínica possível de saturação. Com ativação Softmax, esta camada transforma as ativações em uma distribuição de probabilidade com 30 parâmetros treináveis. A função Softmax garante que a saída do modelo represente a probabilidade atribuída a cada classe, permitindo interpretação clínica direta e precisa. Somando-se todos os parâmetros das três camadas densas, obtém-se um total de 82 parâmetros treináveis. Esse valor reduzido de parâmetros reforça a viabilidade do modelo em sistemas de inferência embarcados, onde recursos computacionais são limitados, sem comprometer a performance na classificação multiclasse de SpO₂.

6.7 ARQUITETURA DA REDE NEURAL PARA CLASSIFICAÇÃO DA TEMPERATURA CORPORAL

O modelo de classificação da temperatura corporal foi desenvolvido utilizando uma arquitetura do tipo *Multilayer Perceptron* (MLP), composta por três camadas ocultas densas com 16, 16 e 8 neurônios, respectivamente, e uma camada de saída com número variável de neurônios, de acordo com a quantidade de categorias clínicas definidas (Figura 31). A escolha desta arquitetura foi fundamentada em heurísticas clássicas de projeto, na natureza da entrada univariada e na necessidade de representação de fronteiras de decisão não lineares entre faixas clínicas de temperatura.

A entrada do modelo consiste em uma única variável contínua (Equação 27):

$$x = \text{Temp} \in \mathbb{R} \quad (27)$$

O objetivo é realizar a classificação clínica com base nesse valor, categorizando o paciente em estados como hipotermia, normotermia, febre leve, moderada ou alta. Assim, define-se a

função de mapeamento como (Equação 28):

$$f : \mathbb{R} \rightarrow \mathbb{R}^C \quad (28)$$

onde C representa o número de classes clínicas possíveis. Apesar da simplicidade da entrada, optou-se por uma rede com três camadas ocultas ($L = 3$) para garantir a extração progressiva de representações não lineares e promover maior estabilidade e generalização durante o treinamento. Essa decisão é respaldada por estudos como os de (Bengio, 2009) e (Goodfellow; Bengio; Courville, 2016), que destacam os benefícios de redes com profundidade moderada, mesmo em problemas aparentemente simples. Para determinar o número de neurônios em cada camada, utilizou-se a heurística dada na Equação 29:

$$n = \alpha \cdot (n_{\text{entrada}} + n_{\text{saída}}) \quad (29)$$

Assumindo $n_{\text{entrada}} = 1$, $n_{\text{saída}} = C$ e $\alpha = 2$, por exemplo, com $C = 8$ (Equação 30):

$$n \approx 2 \cdot (1 + 8) = 18 \Rightarrow \text{arredondado para } 16 \quad (30)$$

Dessa forma, as duas primeiras camadas ocultas foram definidas com 16 neurônios cada. A terceira camada oculta aplicou a estratégia de compressão progressiva (Equação 31):

$$n_3 = \frac{16}{2} = 8 \quad (31)$$

Essa redução de dimensionalidade prepara as ativações intermediárias para a camada de saída, tornando o modelo mais eficiente sem perda de capacidade representacional. A camada de saída contém 8 neurônios, correspondentes às categorias clínicas de temperatura:

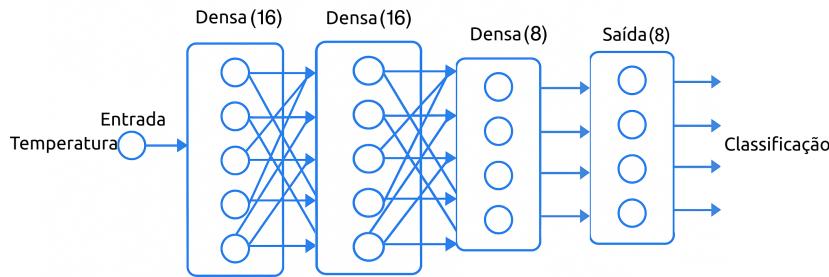
- Hipertirexia (febre muito alta);
- Febre alta;
- Febre baixa;
- Estado febril ou febrícula;
- Normal;
- Hipotermia Leve;
- Hipotermia Moderada;
- Hipotermia Grave;

A função de ativação utilizada na saída é a Softmax (Equação 32), que transforma os valores da última camada em uma distribuição de probabilidade entre as classes:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^C e^{z_j}} \quad (32)$$

A arquitetura completa pode ser descrita da seguinte forma: a primeira camada recebe como entrada o valor normalizado da temperatura corporal e aplica a função ReLU em 16 neurônios, resultando em 32 parâmetros.

Figura 31 – Diagrama da Rede Neural Feedforward (MLP) para classificação de temperatura corporal com camadas Dense(16), Dense(16) e Dense(8).



Fonte: Elaborado pela autora (2025)

A segunda camada, também com 16 neurônios, refina a representação anterior, com 272 parâmetros. A terceira camada oculta contém 8 neurônios, sendo responsável pela redução da dimensionalidade antes da saída, resultando em 136 parâmetros. Por fim, o total de parâmetros treináveis do modelo é de 440 parâmetros treináveis.

6.8 ARQUITETURA DA REDE NEURAL PARA CLASSIFICAÇÃO DA FREQUÊNCIA CARDÍACA

O modelo de classificação da frequência cardíaca (BPM) foi desenvolvido utilizando uma arquitetura do tipo *Multilayer Perceptron* (MLP), composta por três camadas ocultas densas com 16, 16 e 8 neurônios, respectivamente, e uma camada de saída com número variável de neurônios, de acordo com a quantidade de categorias clínicas definidas. A escolha desta arquitetura foi fundamentada em heurísticas clássicas de projeto, considerando a natureza univariada da entrada e a necessidade de representar fronteiras de decisão não lineares entre diferentes faixas clínicas de frequência cardíaca. A entrada do modelo consiste em uma única variável contínua dada pela Equação 33:

$$x = \text{HeartRate} \in \mathbb{R} \quad (33)$$

Para garantir melhor desempenho durante o treinamento e a inferência, os valores de BPM foram previamente normalizados utilizando um *StandardScaler*, ajustado aos dados coletados durante a fase de treinamento. Esse pré-processamento visou reduzir a influência de escalas distintas entre participantes e melhorar a estabilidade numérica do processo de otimização. As camadas ocultas aplicam a função de ativação ReLU, escolhida por sua eficiência computacional e por facilitar o aprendizado de relações não lineares mesmo em redes rasas. O modelo incorpora camadas de *Dropout* como técnica de regularização, prevenindo sobreajuste ao desligar aleatoriamente uma fração dos neurônios durante o treinamento. Essa estratégia fortalece a capacidade de generalização do classificador para novos participantes, cujos sinais fisiológicos podem apresentar variabilidade natural.

A camada de saída utiliza a função de ativação *Softmax*, permitindo a inferência probabilística sobre as diferentes classes clínicas de frequência cardíaca previamente definidas. A função de perda empregada no treinamento foi a *categorical cross-entropy*, apropriada para problemas de classificação multiclasse. O otimizador escolhido foi o *Adam*, com taxa de aprendizado ajustada empiricamente, favorecendo a convergência eficiente mesmo em bases de dados com variabilidade interindividual. Essa configuração arquitetural foi projetada para oferecer um bom equilíbrio entre capacidade preditiva e baixo custo computacional, tornando-se adequada para execução em tempo real no dispositivo vestível ou em servidores remotos integrados ao SMA.

Para determinar o número de neurônios em cada camada, utilizou-se a heurística dada pela Equação 34:

$$n = \alpha \cdot (n_{\text{entrada}} + n_{\text{saída}}) \quad (34)$$

Assumindo $n_{\text{entrada}} = 1$, $n_{\text{saída}} = C$ e $\alpha = 2$, com $C = 3$ (Equação 35):

$$n \approx 2 \cdot (1 + 3) = 8 \quad (35)$$

Dessa forma, as duas primeiras camadas ocultas foram definidas com 8 neurônios cada. A terceira camada oculta aplicou a estratégia de compressão progressiva (Equação 36):

$$n_3 = \frac{8}{2} = 4 \quad (36)$$

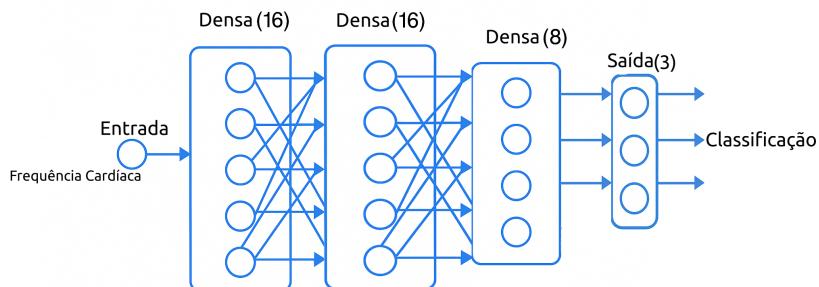
Essa redução de dimensionalidade prepara as ativações intermediárias para a camada de saída, tornando o modelo mais eficiente sem perda de capacidade representacional. A camada de saída contém 3 neurônios, correspondentes às categorias clínicas de frequência cardíaca:

- Bradicardia;
- Normal;

- Taquicardia.

A função de ativação utilizada na saída é a Softmax, que transforma os valores da última camada em uma distribuição de probabilidade entre as classes (Equação 32). A arquitetura completa pode ser descrita da seguinte forma: a primeira camada recebe como entrada o valor normalizado da frequência cardíaca e aplica a função ReLU em 8 neurônios, resultando em 16 parâmetros.

Figura 32 – Diagrama da Rede Neural Feedforward (MLP) para classificação de frequência cardíaca com camadas Dense(16), Dense(16) e Dense(8).



Fonte: Elaborado pela autora (2025)

A segunda camada, também com 8 neurônios, refina a representação anterior, com 72 parâmetros. A terceira camada oculta contém 4 neurônios, sendo responsável pela redução da dimensionalidade antes da saída e resultando em 36 parâmetros. Por fim, o total de parâmetros treináveis do modelo é de 124. Esse número evidencia a leveza computacional da rede, sem comprometer a acurácia na classificação clínica da frequência cardíaca.

6.9 ARQUITETURA DA REDE NEURAL PARA PROGNÓSTICO CLÍNICO

O modelo preditivo utilizado neste estudo foi implementado como uma rede neural multi-camadas MLP com arquitetura sequencial. Seu objetivo é realizar a classificação de instâncias em múltiplas categorias clínicas mutuamente exclusivas, com saída Softmax. A entrada do modelo consiste em um vetor com cinco variáveis fisiológicas normalizadas (Equação 37):

$$\mathbf{x} = [\text{SpO}_2, \text{HR}, \text{Temperatura}, \text{PAS}, \text{PAD}] \quad (37)$$

Onde:

- HR = BPM = Frequencia Cardíaca;
- PAS = Pressão Arterial Sistólica;

- PAD = Pressão Arterial Diastólica.

A arquitetura segue um padrão de expansão inicial com subsequente compressão progressiva (formato de “funil”), prática consolidada em projetos de redes neurais para facilitar a extração de representações latentes e melhorar a capacidade de generalização. O desenho final inclui:

- **Camada 1:** densa com 128 neurônios, ativação ReLU, seguida de *Batch Normalization* e *Dropout* com taxa de 35%. Essa camada inicial mais larga captura interações não lineares complexas entre os sinais fisiológicos e estabiliza o aprendizado ao normalizar as ativações;
- **Camada 2:** densa com 64 neurônios, também com ativação ReLU, *Batch Normalization* e *Dropout* (35%). Essa redução progressiva ajuda a remover redundâncias e força a rede a aprender representações mais compactas;
- **Camada 3:** densa com 32 neurônios, ativação ReLU e *Dropout* (35%). Essa camada final do “funil” contribui para a compressão da informação antes da classificação;
- **Camada de saída:** densa com *num_classes* neurônios (um por categoria clínica possível), usando a função de ativação *Softmax*, que transforma as ativações em uma distribuição de probabilidade sobre as classes. A classe com maior probabilidade é selecionada como a predição final do sistema.

A função de perda utilizada no treinamento foi a *categorical crossentropy*, apropriada para tarefas de classificação multi-classe. O otimizador empregado foi o Adam, com hiperparâmetros ajustados para garantir convergência eficiente. A estratégia de normalização em lote (*Batch Normalization*) e a aplicação consistente de *Dropout* foram fundamentais para mitigar o sobreajuste (*overfitting*) e garantir estabilidade no treinamento em dados potencialmente ruidosos ou heterogêneos.

O modelo foi treinado com um conjunto de dados (dataset) público contendo diversas combinações de sinais fisiológicos, representando cenários clínicos distintos retirados da literatura (Riley et al., 2019; Steyerberg, 2019; Ferrand; Poleksic; Racine, 2022). Os hiperparâmetros (número de neurônios por camada, taxa de *dropout* e número de épocas) foram definidos com base em heurísticas clássicas e validação empírica no conjunto de dados gerado. Para definir a quantidade de neurônios mínima para a primeira camada oculta, utiliza-se a seguinte heurística, dada na Equação 38:

$$n_1 \approx \alpha \cdot (n_{\text{entrada}} + n_{\text{saídas}}) \quad (38)$$

onde:

$$n_{\text{entrada}} = 5$$

$$n_{\text{saída}} = \text{num_classes}$$

$$\alpha \in [2, 6]$$

Assumindo $\alpha = 4$, obtem-se (Equação 39):

$$n_1 = 4 \cdot (5 + n_{\text{saída}}) \quad (39)$$

Por exemplo, se o conjunto de dados possuir $n_{\text{saída}} = 27$ classes (Equação 40):

$$n_1 = 4 \cdot (5 + 27) = 128 \quad (40)$$

Esse valor é então arredondado ou ajustado empiricamente para potências de dois ou valores próximos (como 128) para facilitar a implementação e otimização computacional. A escolha de α e o ajuste final são baseados em heurísticas clássicas de projeto de redes neurais e resultados empíricos de validação no conjunto de dados. Essa camada expandida permite à rede capturar combinações não lineares complexas entre os sinais fisiológicos de entrada. As camadas ocultas seguintes seguem uma estrutura de compressão progressiva, reduzindo aproximadamente pela metade o número de neurônios da camada anterior (Equações 41 e 42):

$$n_2 = \frac{128}{2} = 64 \quad (\text{camada 2}) \quad (41)$$

$$n_3 = \frac{64}{2} = 32 \quad (\text{camada 3}) \quad (42)$$

Esse padrão de redução em formato de *funil* é uma prática consolidada em redes neurais profundas, pois favorece a eliminação de redundâncias, melhora a capacidade de generalização e facilita a separabilidade das classes no espaço latente. Além disso, o modelo foi projetado respeitando o critério de que o número de parâmetros treináveis (P) deve ser significativamente inferior ao número de amostras da base de dados ($P \ll N$), mitigando o risco de sobreajuste.

A primeira camada densa contém 128 neurônios com função de ativação não linear (ReLU). Em seguida, aplica-se uma camada de *Batch Normalization*, que atua sobre as ativações normalizando suas distribuições para acelerar o treinamento e reduzir a sensibilidade à inicialização dos pesos (Ioffe; Szegedy, 2015). Posteriormente, utiliza-se uma camada de *Dropout* com taxa de 35%, que zera aleatoriamente parte das unidades durante o treinamento para mitigar o sobreajuste (Srivastava et al., 2014). Esta camada apresenta 768 parâmetros treináveis.

A segunda camada densa possui 64 neurônios e segue a mesma estrutura de bloco: ativação ReLU, *Batch Normalization* e *Dropout* (35%). Essa configuração ajuda a estabilizar o aprendizado mesmo com o aumento da profundidade. Ela contém 8,256 parâmetros treináveis. A terceira camada densa conta com 32 neurônios e utiliza ativação ReLU, seguida apenas por uma

camada de *Dropout* (35%), sem normalização adicional. Essa escolha simplifica a compactação progressiva da representação interna e fortalece a generalização do modelo. Essa camada possui 2,080 parâmetros treináveis.

A camada de saída possui 27 neurônios, onde 27 corresponde ao número de categorias clínicas distintas definidas pelo *one-hot encoding* aplicado ao conjunto de dados. Essa camada emprega a função de ativação Softmax, responsável por transformar as ativações em uma distribuição de probabilidade sobre as classes disponíveis. A classe com maior probabilidade é selecionada como a predição final do sistema de 891 parâmetros treináveis.

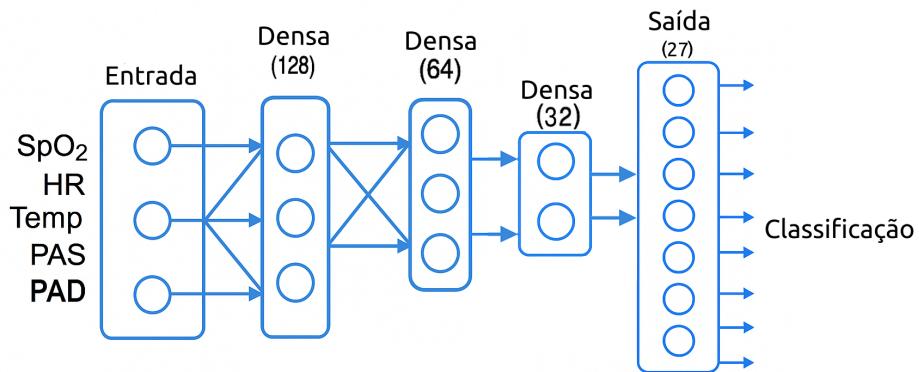
Essa configuração geral foi definida com base em heurísticas de projeto de redes neurais e resultados empíricos obtidos durante os experimentos de validação, garantindo capacidade de modelagem adequada para o problema clínico e controle do sobreajuste. A ativação Softmax transforma os valores da saída em uma distribuição de probabilidade sobre as seguintes classes:

1. **Insuficiência Respiratória:** Hipoxemia grave ($\text{SpO}_2 < 90$) associada a taquicardia ($\text{HR} > 100 \text{ bpm}$);
2. **Choque Circulatório:** Hipoxemia grave ($\text{SpO}_2 < 85$) combinada com hipotensão ($\text{PAS} < 90 \text{ mmHg}$);
3. **Choque Hipovolêmico:** Hipotensão significativa ($\text{PAS} < 90 \text{ mmHg}$, $\text{PAD} < 60 \text{ mmHg}$) acompanhada de taquicardia intensa ($\text{HR} > 120 \text{ bpm}$);
4. **Insuficiência Respiratória Iminente:** Hipoxemia moderada (SpO_2 entre 85–90) com taquicardia acentuada ($\text{HR} > 110 \text{ bpm}$);
5. **Infecção Grave/Choque Séptico:** Temperatura elevada ($> 39^\circ\text{C}$) acompanhada de taquicardia ($\text{HR} > 110 \text{ bpm}$);
6. **Colapso Metabólico:** Hipotermia grave (Temperatura $< 35^\circ\text{C}$) combinada com bradicardia ($\text{HR} < 50 \text{ bpm}$);
7. **Insuficiência Respiratória Progressiva:** Hipotermia leve ($\text{Temp} < 36^\circ\text{C}$) associada a SpO_2 entre 90–94%;
8. **Emergência Hipertensiva:** PAS muito elevada ($> 180 \text{ mmHg}$) com HR muito alto ($> 120 \text{ bpm}$) ou muito baixo ($< 50 \text{ bpm}$);
9. **Choque Compensado:** Hipotensão moderada (PAS entre 90–100 mmHg) com taquicardia ($\text{HR} > 110 \text{ bpm}$);
10. **Deterioração Clínica Grave:** Hipotensão ($\text{PAS} < 90 \text{ mmHg}$) acompanhada de hipoxemia ($\text{SpO}_2 < 90\%$);
11. **Insuficiência Respiratória Crônica:** Hipoxemia grave ($\text{SpO}_2 < 85$) com bradicardia ($\text{HR} < 50 \text{ bpm}$);

12. **Hipotermia/Desnutrição Severa:** Temperatura subnormal ($< 35^{\circ}\text{C}$) com hipotensão crônica (PAS entre 80–100 mmHg);
13. **Síncope por Hipotensão Postural:** Episódios de PAS baixa (< 110 mmHg) ou HR baixo (< 50 bpm), simulando quedas ou síncope;
14. **Crise de Ansiedade:** Saturação normal ($> 98\%$), HR elevado (100–120 bpm) e leve aumento de PAS (140–150 mmHg), sem febre significativa;
15. **Quadro Hipertensivo:** Hipertensão leve (PAS entre 140–159 mmHg, PAD entre 90–99 mmHg), com sinais vitais em faixa normal;
16. **Alto Risco de Deterioração Clínica:** Combinação de hipertensão frequente (PAS > 150 mmHg), possíveis saturações anormais ($< 90\%$) e febre moderada ($> 38^{\circ}\text{C}$) em parte dos casos;
17. **Hipertensão Urgência:** Crise hipertensiva assintomática com PAS > 180 mmHg e PAD > 110 mmHg, sem sintomas graves associados;
18. **Taquicardia Sinusal Isolada:** HR elevado (100–130 bpm) com demais sinais vitais dentro de faixas normais;
19. **Bradicardia Sinusal:** HR baixo (40–59 bpm) com PAS e demais sinais em níveis normais;
20. **Reflexo de Cushing:** Hipertensão significativa (PAS > 160 mmHg) associada a bradicardia (HR < 50 bpm);
21. **Choque Séptico Compensado:** Temperatura elevada (38.5 – 39.5°C) com PAS limítrofe (90–100 mmHg) e taquicardia (HR 100–130 bpm);
22. **Hipotensão Isolada:** PAS < 90 mmHg com HR e outros sinais dentro de faixas normais;
23. **Febre Moderada Isolada:** Temperatura entre 38 – 39°C sem alterações relevantes nos demais sinais;
24. **Hipertensão Sistólica Isolada:** PAS elevada (160–190 mmHg) com PAD normal ou baixo (60–85 mmHg), simulando padrão de envelhecimento arterial;
25. **Pressão de Pulso Ampla:** Diferença sistólico-diastólica ampliada (pressão de pulso elevada), com demais sinais normais;
26. **Síndrome Hipermetabólica Leve:** Temperatura discretamente elevada (37.5 – 38.0°C) com HR moderadamente alto (90–110 bpm);
27. **Sem Anormalidades Graves Futuras Detectadas:** Sinais vitais dentro de faixas normais, simulando indivíduos sem alterações graves previstas.

O pseudocódigo referente ao treinamento e à geração do modelo de prognóstico pode ser consultado no Apêndice A.2, e os principais termos técnicos encontram-se definidos no Apêndice B, Tabela 14.

Figura 33 – Arquitetura da rede neural para prognóstico clínico com entrada multivariada, camadas densas com normalização e saída com classificação em categorias clínicas.



Fonte: Elaborado pela autora (2025)

6.10 ARQUITETURA DA REDE NEURAL PARA DIAGNÓSTICO MULTI-LABEL

A arquitetura do modelo de diagnóstico clínico multi-label implementada neste trabalho consiste em uma rede neural MLP com quatro camadas densas organizadas em blocos estruturados, incorporando mecanismos de normalização e regularização. Seu objetivo é inferir simultaneamente múltiplos diagnósticos clínicos potenciais para cada paciente, a partir dos cinco sinais fisiológicos: saturação periférica de oxigênio (SpO_2), frequência cardíaca (HR), temperatura corporal (Temp), pressão arterial sistólica (PAS) e pressão arterial diastólica (PAD).

Diferentemente de uma abordagem de classificação *multi-class*, em que cada amostra pertence exatamente a uma entre várias categorias mutuamente exclusivas, esta tarefa foi formulada como uma classificação multi-label, permitindo que uma mesma instância possua mais de um diagnóstico clínico associado. Para esse fim, a camada de saída do modelo utiliza uma função de ativação *sigmoid*, gerando probabilidades independentes para cada diagnóstico, com função de perda *binary_crossentropy*. A entrada do modelo é um vetor com cinco variáveis fisiológicas normalizadas (Equação 43):

$$\mathbf{x} = [SpO_2, HR, Temp, PAS, PAD] \Rightarrow n_{\text{entrada}} = 5 \quad (43)$$

A saída possui n neurônios, onde n corresponde ao número de diagnósticos possíveis identificados no conjunto de dados após o *one-hot encoding*. Esse valor pode variar conforme o

banco de dados utilizado, o que confere flexibilidade ao modelo para diferentes contextos clínicos. Cada saída \hat{y}_i é interpretada como a probabilidade independente de presença do diagnóstico i (Equação 44):

$$\hat{y}_i = \sigma(z_i) \in [0, 1] \quad \forall i \in \{1, \dots, n\} \quad (44)$$

A arquitetura adota um padrão de expansão inicial seguido de compressão progressiva em formato de “funil”, estratégia consagrada que visa melhorar a capacidade de generalização e reduzir redundâncias. A estrutura detalhada é:

- **Camada 1:** 128 neurônios com ativação ReLU, seguida de *Batch Normalization* e *Dropout* (taxa de 35%);
- **Camada 2:** 64 neurônios com ativação ReLU, *Batch Normalization* e *Dropout* (35%);
- **Camada 3:** 32 neurônios com ativação ReLU e *Dropout* (35%);
- **Camada de saída:** 27 neurônios com ativação *sigmoid*.

Esse desenho modular visa capturar combinações não lineares complexas entre os sinais fisiológicos de entrada, estabilizar o treinamento por meio de normalização e reduzir o risco de sobreajuste via regularização estocástica (*dropout*).

O modelo foi treinado com a função de perda *binary_crossentropy*, utilizando o otimizador Adam, por 160 épocas e com 20% dos dados destinados à validação. Além do modelo em si, foram salvos os artefatos auxiliares, como o scaler de normalização e os nomes das classes detectadas no *one-hot encoding*, para permitir reaplicação eficiente em ambientes de produção e integração ao sistema multiagente distribuído proposto neste estudo. O conjunto de dados utilizado para o treinamento do modelo de diagnóstico multi-label abrange as mesmas 27 classes diagnósticas distintas adotadas no modelo de prognóstico. Entretanto, enquanto o modelo de prognóstico infere uma única classe com base no histórico completo de sinais fisiológicos do participante, o modelo de diagnóstico multi-label possui a capacidade de inferir múltiplas condições simultaneamente e em tempo real, a partir de combinações específicas das inferências dos subagentes de diagnóstico, que incluem SpO₂, frequência cardíaca, temperatura corporal, e pressões arteriais sistólica e diastólica.

6.11 VALIDAÇÃO DA ACURÁCIA DO SMACARE EM SINAIS INDIVIDUAIS E SAÚDE AUTORREFERIDA

Este estudo foi aprovado pelo Comitê de Ética em Pesquisa com Seres Humanos da Universidade do Estado de Santa Catarina (CAAE 82997424.7.0000.0118). Os experimentos seguintes foram conduzidos com a participação de trinta participantes (Tabela 6), ao longo de um

período de dez dias. As sessões de coleta de dados não foram realizadas em dias consecutivos, sendo organizadas de acordo com a disponibilidade individual dos participantes. Durante os testes, o dispositivo vestível foi posicionado de forma fixa no braço, enquanto o sensor de fotopletismografia permaneceu acoplado ao lóbulo da orelha por meio de uma extensão anatômica adaptada (Figura 34). Essa configuração possibilitou a aquisição contínua dos sinais vitais, garantindo estabilidade e conforto para o usuário. Os sinais coletados foram transmitidos via protocolo BLE tanto para o computador que executava o SMA quanto para um smartphone. Este último, por sua vez, encaminhou os dados para a nuvem utilizando o protocolo MQTT. A coluna *Saúde Autorelatada* contém a informação sobre o estado de saúde do sujeito que foi obtida por meio de autorrelato do participante, com base em diagnóstico previamente estabelecido por profissional médico.

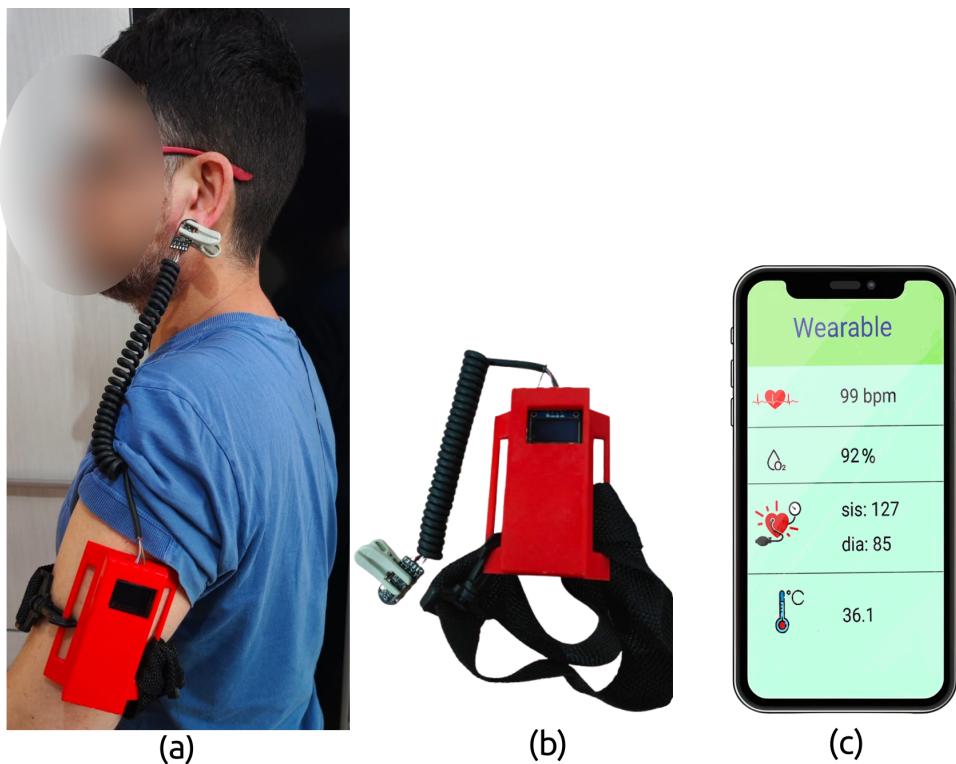
Tabela 6 – Características demográficas, antropométricas e clínicas dos trinta participantes do estudo.

| Subject ID | Idade | Altura (cm) | IMC ^a (kg/m ²) | Sexo | Saúde Autorelatada |
|------------|-------|-------------|---------------------------------------|-----------|--------------------|
| 1 | 40 | 171 | 39,30 | Feminino | Ansiedade |
| 2 | 68 | 184 | 26,60 | Masculino | Hipertenso |
| 3 | 74 | 156 | 27,20 | Feminino | Asma |
| 4 | 75 | 170 | 17,30 | Feminino | Normal |
| 5 | 60 | 185 | 25,27 | Masculino | Normal |
| 6 | 79 | 165 | 25,71 | Feminino | Normal |
| 7 | 87 | 155 | 17,78 | Feminino | Hipertenso |
| 8 | 37 | 175 | 22,86 | Feminino | Ansiedade |
| 9 | 48 | 166 | 29,03 | Feminino | Ansiedade |
| 10 | 42 | 170 | 25,26 | Masculino | Normal |
| 11 | 51 | 152 | 23,40 | Feminino | Normal |
| 12 | 48 | 173 | 31,40 | Masculino | Hipertenso |
| 13 | 26 | 169 | 21,00 | Masculino | Normal |
| 14 | 30 | 183 | 31,40 | Masculino | Hipoxemia |
| 15 | 45 | 160 | 27,30 | Feminino | Normal |
| 16 | 49 | 177 | 25,50 | Masculino | Normal |
| 17 | 88 | 165 | 25,70 | Feminino | Febril |
| 18 | 27 | 154 | 25,30 | Feminino | Febril |
| 19 | 50 | 177 | 35,10 | Feminino | Ansiedade |
| 20 | 79 | 170 | 31,80 | Feminino | Câncer Colorretal |
| 21 | 78 | 177 | 25,50 | Masculino | Alzheimer |
| 22 | 60 | 157 | 21,90 | Feminino | Hipertenso |
| 23 | 53 | 165 | 40,80 | Feminino | Ansiedade |
| 24 | 57 | 180 | 28,10 | Masculino | Ansiedade |
| 25 | 60 | 160 | 23,00 | Feminino | Normal |
| 26 | 76 | 160 | 23,40 | Feminino | Taquicardia |
| 27 | 57 | 160 | 32,00 | Feminino | Hipertenso |
| 28 | 75 | 160 | 27,00 | Feminino | Asma |
| 29 | 63 | 156 | 27,50 | Feminino | Asma |
| 30 | 61 | 182 | 29,60 | Masculino | Normal |

^a Índice de Massa Corporal (IMC), definido como peso (kg) dividido pelo quadrado da altura (m²).

Fonte: Elaborado pela autora (2025).

Figura 34 – Protótipo do dispositivo vestível desenvolvido neste estudo: (a) o sistema vestível em uso; (b) o protótipo vestível e a cinta de fixação; e (c) a interface gráfica do aplicativo móvel personalizado.



Fonte: Elaborado pela autora (2025).

Os testes foram realizados com o objetivo de avaliar a acurácia do Sistema Multiagente tanto com base nos sinais fisiológicos coletados em tempo real quanto no diagnóstico derivado das informações de saúde autoreferidas pelos participantes. Para isso, o SMA foi executado no Cenário 1, sendo registrada a taxa de acerto em função dos sinais fisiológicos. A Tabela 7 apresenta um retrato sobre a concordância entre o diagnóstico automático (baseado nos sinais fisiológicos enviados pelo dispositivo) e o relato subjetivo de saúde feito pelos próprios participantes.

Nota-se que para a maioria dos sujeitos, tanto o diagnóstico multi-label como a comparação com o autorelato resultaram em 100% de acerto, mostrando forte concordância entre o sistema e a percepção do próprio participante. Esse resultado positivo sugere que o modelo consegue, em geral, identificar corretamente as condições fisiológicas e se alinha bem com o estado de saúde percebido. Isso é fundamental para reforçar a confiança clínica em sistemas de triagem baseados em IA. Contudo, a tabela também evidencia casos de discrepância, especialmente nos registros anotados como “Saúde tida como normal (100%)” em que o sistema apontou um resultado diferente, ou onde a anotação de saúde referida (como ansiedade, hipertensão ou asma) não foi detectada, resultando em 0% de acerto. Quando o participante se declara saudável, mas o sistema detecta um sinal anômalo, há algumas interpretações possíveis:

Sinais fisiológicos subclínicos ou não percebidos: é plausível que o modelo detecte

alterações sutis que o indivíduo não nota ou não considera um problema. Por exemplo, variações discretas de frequência cardíaca ou pressão arterial podem não gerar sintomas perceptíveis, mas ainda sinalizarem risco;

Sensibilidade do modelo: o algoritmo pode estar calibrado para detectar qualquer desvio dos padrões normais, mesmo que esse desvio não seja clinicamente relevante ou que não se manifeste subjetivamente. Isso pode resultar em falsos positivos — úteis em triagem preventiva, mas potencialmente geradores de ansiedade ou necessidade de refinamento.

Por outro lado, há casos em que o participante relatou condições como ansiedade ou hipertensão, mas o sistema não detectou essas condições, resultando em 0% de acerto. Possíveis explicações para isso incluem limitações nos dados de treinamento, já que, se as amostras de indivíduos com essas condições eram insuficientes ou não representavam adequadamente a variabilidade desses estados, o modelo pode ter dificuldades para aprender seus padrões. A natureza episódica dessas condições, especialmente em transtornos como a ansiedade, faz com que os sinais fisiológicos variem ao longo do tempo, de modo que uma coleta pontual pode não capturar o estado patológico ativo e levar a classificações como “Normal”. A heterogeneidade individual, com variações fisiológicas normais entre diferentes pessoas, pode mascarar sinais diagnósticos e dificultar uma classificação binária clara. Vale destacar que os participantes com essas condições estavam sob acompanhamento médico para o tratamento de suas enfermidades.

Nota-se porém, que os sujeitos 4, 13, 16 e 30 ilustram bem o dilema clínico: embora tenham autorelatado saúde normal, o sistema indicou outra condição. Isso destaca um ponto crítico em aplicações biomédicas: o diagnóstico automatizado deve ser interpretado como uma ferramenta de apoio, e não substitutiva, especialmente em populações saudáveis ou sem sintomas aparentes.

Tabela 7 – Percentual de acertos por participante nos diagnósticos baseados em sinais fisiológicos e na saúde autorelatada.

| Subject ID | Diagnóstico por Sinal (%) | Diagnóstico Multi-label (%) | Diagnóstico Multi-label (saúde autorelatada) (%) |
|------------|---------------------------|-----------------------------|--|
| 1 | 100 | 100 | Não detectou ansiedade (0%) |
| 2 | 100 | 100 | 100 |
| 3 | 100 | 100 | 100 |
| 4 | 100 | 90 | Saúde tida como normal (100%) |
| 5 | 100 | 100 | 100 |
| 6 | 100 | 100 | 100 |
| 7 | 100 | 100 | 100 |
| 8 | 100 | 100 | Não detectou ansiedade (0%) |
| 9 | 100 | 100 | Não detectou ansiedade (0%) |
| 10 | 100 | 100 | 100 |
| 11 | 100 | 100 | 100 |
| 12 | 100 | 100 | 100 |
| 13 | 100 | 90 | Saúde tida como normal (100%) |
| 14 | 100 | 100 | 100 |
| 15 | 100 | 100 | 100 |
| 16 | 100 | 100 | Saúde tida como normal (100%) |
| 17 | 100 | 100 | 100 |
| 18 | 100 | 100 | 100 |
| 19 | 100 | 100 | Não detectou ansiedade (0%) |
| 20 | 100 | 100 | 100 |
| 21 | 100 | 100 | 100 |
| 22 | 100 | 100 | 100 |
| 23 | 100 | 100 | Não detectou ansiedade (0%) |
| 24 | 100 | 100 | 100 |
| 25 | 100 | 100 | 100 |
| 26 | 100 | 100 | 100 |
| 27 | 100 | 100 | Não detectou hipertensão (0%) |
| 28 | 100 | 100 | 100 |
| 29 | 100 | 100 | Não detectou asma (0%) |
| 30 | 100 | 100 | Saúde tida como normal (100%) |

Fonte: Elaborado pela autora (2025).

6.12 TESTE DE LATÊNCIA NA DETECÇÃO E NOTIFICAÇÃO DE ANOMALIAS

Outro teste realizado consistiu na avaliação do tempo médio que o Sistema Multiagente leva para notificar o usuário para cada sinal fisiológico, para o prognóstico e para o diagnóstico multi-label, utilizando o componente *NotifyAgent*. Durante 10 dias, o participante utilizou o *wearable*, enquanto o SMA realizava inferências em tempo real para diagnósticos específicos por sinal fisiológico, além de gerar o diagnóstico multi-label após o processamento dos quatro sinais (Tabela 8). Para a tarefa de Prognóstico, o tempo de latência foi medido considerando o intervalo entre o processamento realizado pelo *PrognosisAgent* e o envio da notificação pelo *NotifyAgent*. É importante destacar que os valores indicados como *N/A* significam que não

houve necessidade de notificação para aquele sinal ou tarefa, ou seja, o sistema não identificou qualquer anormalidade que justificasse um alerta por parte do NotifyAgent. A Tabela 9 apresenta os desvios padrão correspondentes aos valores de tempo médio de latência exibidos na Tabela 8.

De forma geral, os resultados mostram que quando ocorre notificação, o tempo médio para gerar o aviso ao usuário é relativamente baixo, geralmente entre aproximadamente 0,063 e 0,069 segundos, com desvios padrão igualmente reduzidos (tipicamente em torno de 0,005 – 0,01 s). Essa característica indica alta responsividade e estabilidade temporal do SMA, algo essencial para aplicações em tempo real como sistemas de monitoramento domiciliar ou dispositivos vestíveis.

A presença dos valores *N/A* é, na verdade, um aspecto positivo, pois demonstra que o sistema é seletivo: ele não notifica desnecessariamente em situações normais, evitando alarmes falsos e contribuindo para a confiabilidade percebida pelo usuário. Em outras palavras, a lógica de notificação é sensível a desvios relevantes nos sinais fisiológicos e silenciosa quando os parâmetros estão dentro da normalidade. Isso é um requisito importante para aplicações práticas, pois sistemas que notificam em excesso podem gerar fadiga de alerta ou serem desconsiderados pelos usuários.

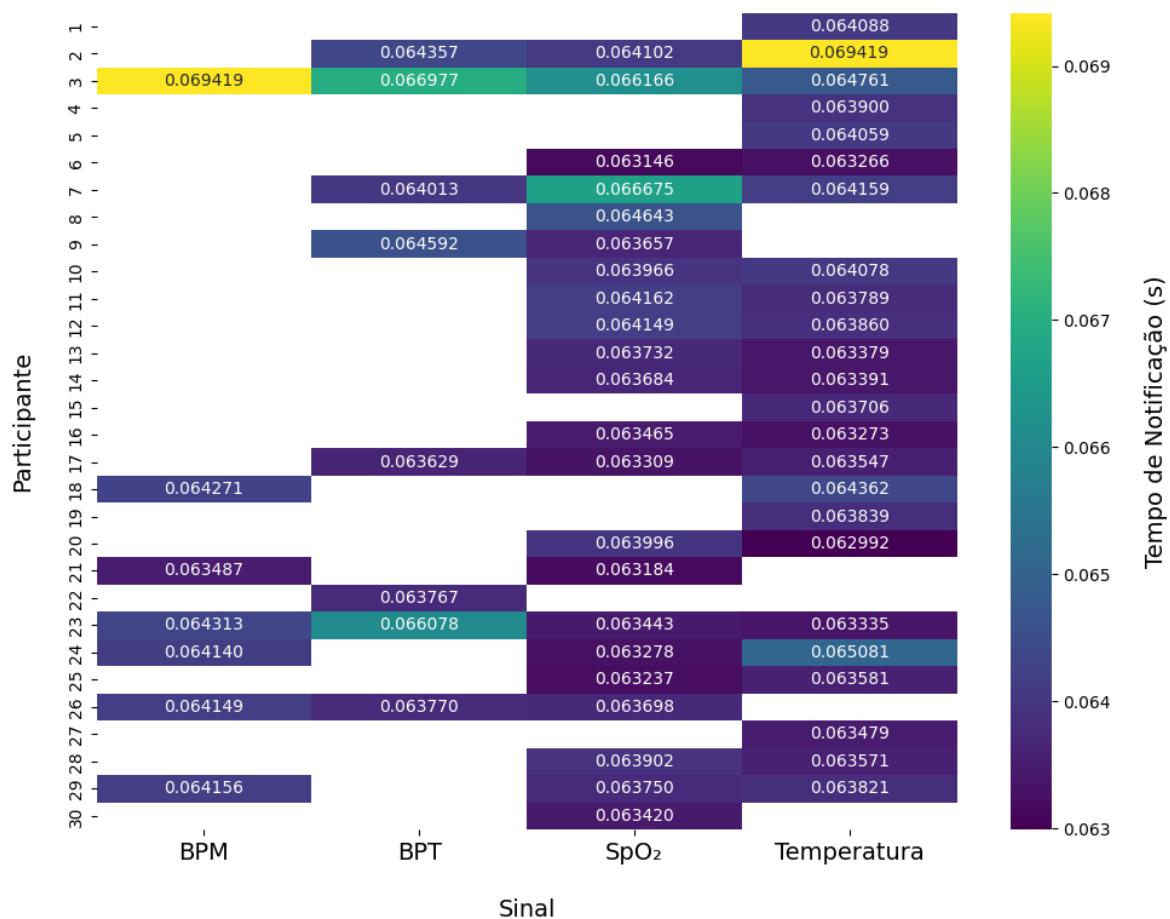
Observa-se também boa consistência entre os participantes, já que mesmo quando ocorrem notificações para sinais diferentes, os tempos médios de inferência e notificação permanecem semelhantes entre os indivíduos, com baixa variabilidade. Essa estabilidade indica que o NotifyAgent consegue fornecer alertas com rapidez previsível, independentemente das variações individuais ou do tipo de sinal processado. Além disso, sinais como SpO₂ e Temperatura aparecem mais preenchidos na tabela, o que sugere que esses parâmetros geraram notificações mais recorrentes, refletindo tanto sua relevância clínica quanto a confiabilidade dos sensores utilizados. Em contraste, sinais como BPM e BPT mostram mais ocorrências de *N/A*, o que indica menor frequência de anomalias detectadas nesses dados. Vale ressaltar ainda que o período de coleta ocorreu em uma época de alta incidência de doenças respiratórias, o que pode ter influenciado diretamente no número de diagnósticos ou alertas relacionados ao SpO₂ e a Temperatura.

A Figura 35 ilustra um heatmap dos tempos médios de notificação pelo NotifyAgent (em segundos) para cada sujeito e sinal fisiológico monitorado no SMACare. Cada célula apresenta a média do tempo de notificação para um sujeito específico (eixo Y) em relação ao sinal fisiológico monitorado (BPM, BPT, SpO₂, Temperatura no eixo X). A escala de cores indica a variação dos tempos médios, permitindo identificar padrões interindividuais e potenciais diferenças entre sinais. Os resultados evidenciam a estabilidade geral dos tempos de notificação, com variações pontuais entre sujeitos e sinais fisiológicos. A Figura 36 ilustra o heatmap do desvio padrão dos tempos de notificação (em segundos) para cada sujeito e sinal fisiológico monitorado no Sistema Multiagente (SMA). Cada célula representa a variabilidade temporal na entrega de notificações para um sujeito específico (eixo Y) em relação ao sinal fisiológico (BPM,

BPT, SpO₂, Temperatura no eixo X). A escala de cores quantifica a dispersão dos tempos médios, permitindo identificar a consistência do desempenho do sistema entre indivíduos e parâmetros monitorados.

Em termos de diagnóstico multi-label e prognóstico, os tempos médios de notificação também permanecem baixos e consistentes, reforçando que o sistema consegue integrar múltiplos sinais e entregar diagnósticos compostos ou previsões de risco sem comprometer a velocidade de resposta. Esses resultados positivos demonstram a viabilidade prática do SMA para aplicações de saúde digital: o sistema oferece notificações rápidas e estáveis apenas quando necessário, contribuindo para monitoramento eficiente, redução de alarmes falsos e melhor aceitação por usuários e profissionais de saúde. Ao mesmo tempo, os dados ressaltam a importância de continuar investindo em qualidade e cobertura sensorial, para manter essa performance em diferentes populações e contextos de uso.

Figura 35 – *Heatmap* dos tempos médios de notificação (s) para BPM (Frequência Cardíaca), BPT (Pressão Arterial Estimada), SpO₂ e Temperatura em diferentes sujeitos monitorados pelo SMA.



Fonte: Elaborado pela autora (2025).

As Figuras 37 e 38 apresentam, respectivamente, heatmaps dos tempos médios de notificação (em segundos) e dos respectivos desvios padrão para as tarefas de Diagnóstico Multi-label

Tabela 8 – Valores médios do tempo de notificação (em segundos) por participante, considerando sinais fisiológicos (BPM, BPT, SpO₂, Temperatura) e os tempos de resposta de Diagnóstico Multi-label e Prognóstico.

| Subject | BPM | BPT | SpO ₂ | Temperatura | Diagnóstico Multi-label | Prognóstico |
|-----------|----------|----------|------------------|-------------|-------------------------|-------------|
| 1 | N/A | N/A | N/A | 0,064088 | N/A | N/A |
| 2 | N/A | 0,064357 | 0,064102 | 0,069419 | 0,065269 | 0,065106 |
| 3 | 0,069419 | 0,066977 | 0,066166 | 0,064761 | 0,064244 | 0,064486 |
| 4 | N/A | N/A | N/A | 0,063900 | 0,064971 | N/A |
| 5 | N/A | N/A | N/A | 0,064059 | N/A | N/A |
| 6 | N/A | N/A | 0,063146 | 0,063266 | 0,064047 | N/A |
| 7 | N/A | 0,064013 | 0,066675 | 0,064159 | 0,064924 | 0,065223 |
| 8 | N/A | N/A | 0,064643 | N/A | 0,065150 | N/A |
| 9 | N/A | 0,064592 | 0,063657 | N/A | N/A | N/A |
| 10 | N/A | N/A | 0,063966 | 0,064078 | N/A | N/A |
| 11 | N/A | N/A | 0,064162 | 0,063789 | N/A | N/A |
| 12 | N/A | N/A | 0,064149 | 0,063860 | 0,064603 | 0,064942 |
| 13 | N/A | N/A | 0,063732 | 0,063379 | 0,065212 | N/A |
| 14 | N/A | N/A | 0,063684 | 0,063391 | 0,067070 | N/A |
| 15 | N/A | N/A | N/A | 0,063706 | N/A | N/A |
| 16 | N/A | N/A | 0,063465 | 0,063273 | 0,066318 | N/A |
| 17 | N/A | 0,063629 | 0,063309 | 0,063547 | 0,066584 | N/A |
| 18 | 0,064271 | N/A | N/A | 0,064362 | 0,065114 | N/A |
| 19 | N/A | N/A | N/A | 0,063839 | N/A | N/A |
| 20 | N/A | N/A | 0,063996 | 0,062992 | 0,065876 | N/A |
| 21 | 0,063487 | N/A | 0,063184 | N/A | 0,066012 | N/A |
| 22 | N/A | 0,063767 | N/A | N/A | 0,066502 | N/A |
| 23 | 0,064313 | 0,066078 | 0,063443 | 0,063335 | 0,066535 | N/A |
| 24 | 0,064140 | N/A | 0,063278 | 0,065081 | 0,064347 | 0,064524 |
| 25 | N/A | N/A | 0,063237 | 0,063581 | N/A | N/A |
| 26 | 0,064149 | 0,063770 | 0,063698 | N/A | 0,067118 | N/A |
| 27 | N/A | N/A | N/A | 0,063479 | N/A | N/A |
| 28 | N/A | N/A | 0,063902 | 0,063571 | 0,065752 | 0,064308 |
| 29 | 0,064156 | N/A | 0,063750 | 0,063821 | 0,069048 | 0,065110 |
| 30 | N/A | N/A | 0,063420 | N/A | 0,066927 | 0,064875 |

Fonte: Elaborado pela autora (2025).

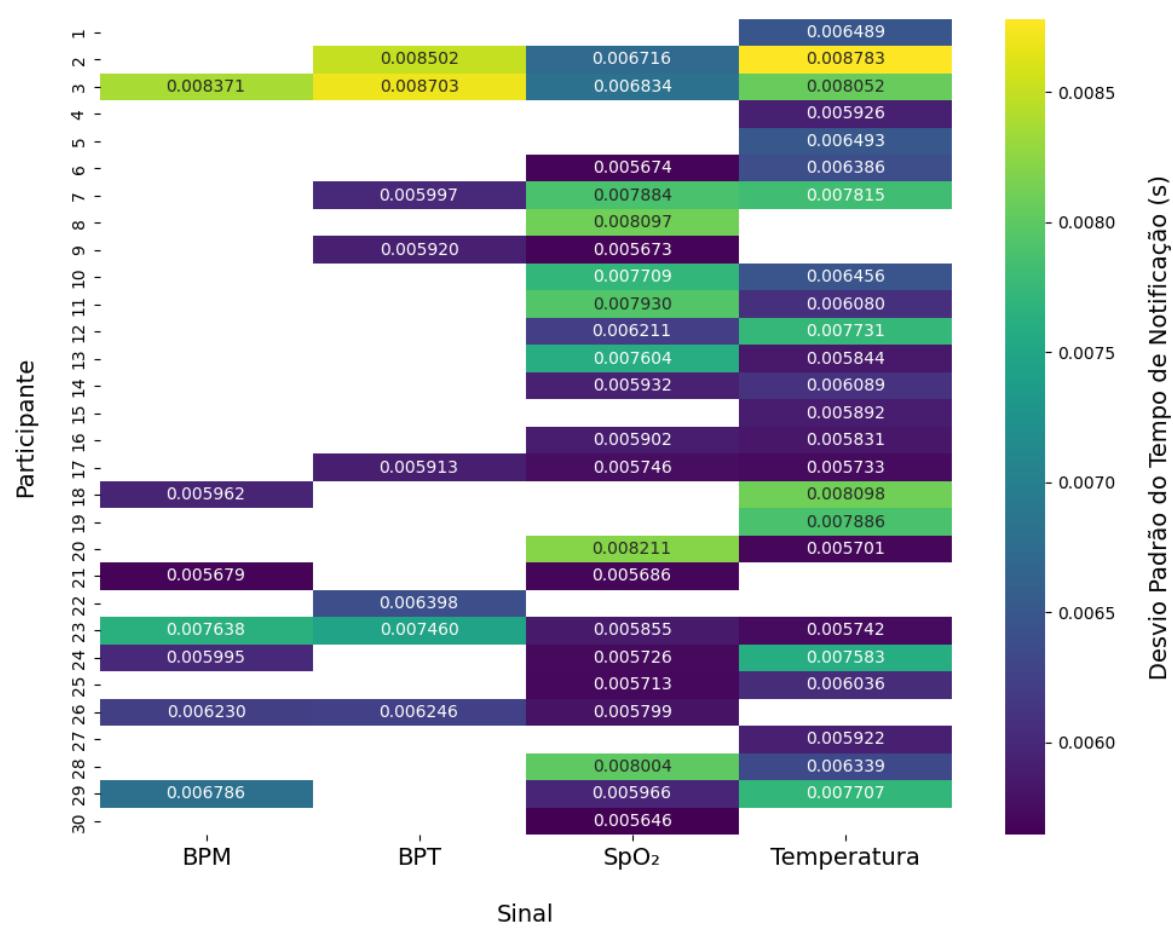
e Prognóstico, por participante. Cada célula indica o tempo médio de resposta do Sistema Multi-agente (SMA) para um determinado indivíduo (eixo Y) em cada tarefa de inferência (eixo X). A escala de cores evidencia variações interindividuais, permitindo analisar a consistência, a estabilidade temporal e o desempenho comparativo entre os módulos de diagnóstico e prognóstico do sistema.

Tabela 9 – Resultados dos desvios padrão do tempo de notificação (em segundos) por participante para cada sinal fisiológico e para as tarefas de Diagnóstico Multi-label e Prognóstico no sistema proposto.

| Subject | BPM | BPT | SpO₂ | Temperatura | Diagnóstico Multi-label | Prognóstico |
|----------------|------------|------------|------------------------|--------------------|--------------------------------|--------------------|
| 1 | N/A | N/A | N/A | 0,006489 | N/A | N/A |
| 2 | N/A | 0,008502 | 0,006716 | 0,008783 | 0,008914 | 0,008668 |
| 3 | 0,008371 | 0,008703 | 0,006834 | 0,008052 | 0,009117 | 0,008402 |
| 4 | N/A | N/A | N/A | 0,005926 | 0,010538 | N/A |
| 5 | N/A | N/A | N/A | 0,006493 | N/A | N/A |
| 6 | N/A | N/A | 0,005674 | 0,006386 | 0,008662 | N/A |
| 7 | N/A | 0,005997 | 0,007884 | 0,007815 | 0,011114 | 0,009259 |
| 8 | N/A | N/A | 0,008097 | N/A | 0,010659 | N/A |
| 9 | N/A | 0,005920 | 0,005673 | N/A | N/A | N/A |
| 10 | N/A | N/A | 0,007709 | 0,006456 | N/A | N/A |
| 11 | N/A | N/A | 0,007930 | 0,006080 | N/A | N/A |
| 12 | N/A | N/A | 0,006211 | 0,007731 | 0,009471 | 0,008661 |
| 13 | N/A | N/A | 0,007604 | 0,005844 | 0,008830 | N/A |
| 14 | N/A | N/A | 0,005932 | 0,006089 | 0,012261 | N/A |
| 15 | N/A | N/A | N/A | 0,005892 | 0,009886 | N/A |
| 16 | N/A | N/A | 0,005902 | 0,005831 | 0,009068 | N/A |
| 17 | N/A | 0,005913 | 0,005746 | 0,005733 | 0,009091 | N/A |
| 18 | 0,005962 | N/A | N/A | 0,008098 | 0,008909 | N/A |
| 19 | N/A | N/A | N/A | 0,007886 | N/A | N/A |
| 20 | N/A | N/A | 0,008211 | 0,005701 | 0,008943 | N/A |
| 21 | 0,005679 | N/A | 0,005686 | N/A | 0,011734 | N/A |
| 22 | N/A | 0,006398 | N/A | N/A | 0,009316 | N/A |
| 23 | 0,007638 | 0,007460 | 0,005855 | 0,005742 | 0,009157 | N/A |
| 24 | 0,005995 | N/A | 0,005726 | 0,007583 | 0,008566 | 0,009350 |
| 25 | N/A | N/A | 0,005713 | 0,006036 | N/A | N/A |
| 26 | 0,006230 | 0,006246 | 0,005799 | N/A | 0,011465 | N/A |
| 27 | N/A | N/A | N/A | 0,005922 | N/A | N/A |
| 28 | N/A | N/A | 0,008004 | 0,006339 | 0,008610 | 0,008491 |
| 29 | 0,006786 | N/A | 0,005966 | 0,007707 | 0,010013 | 0,008669 |
| 30 | N/A | N/A | 0,005646 | N/A | 0,008436 | 0,008458 |

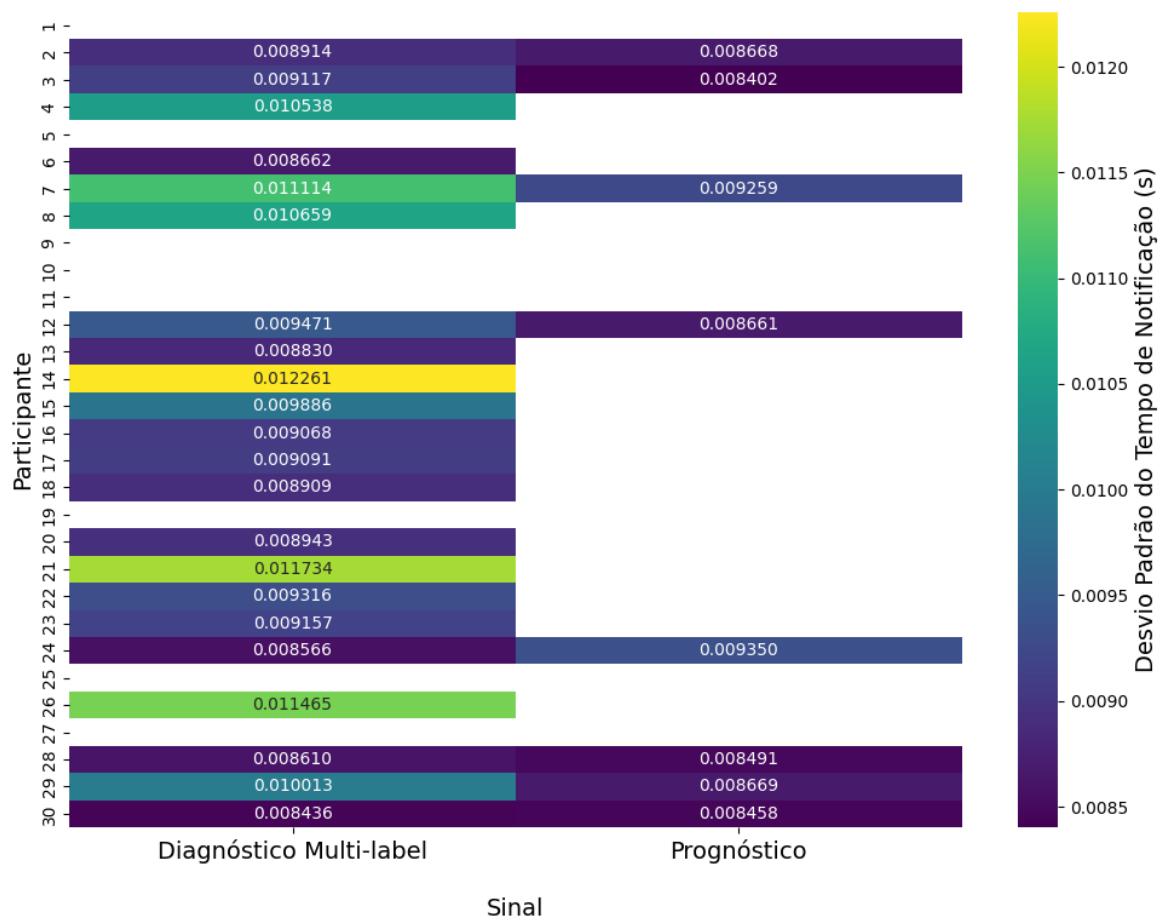
Fonte: Elaborado pela autora (2025).

Figura 36 – *Heatmap* do desvio padrão dos tempos de notificação (s) para BPM (Frequência Cardíaca), BPT (Pressão Arterial Estimada), SpO₂ e Temperatura em diferentes sujeitos monitorados no Sistema Multiagente (SMA).



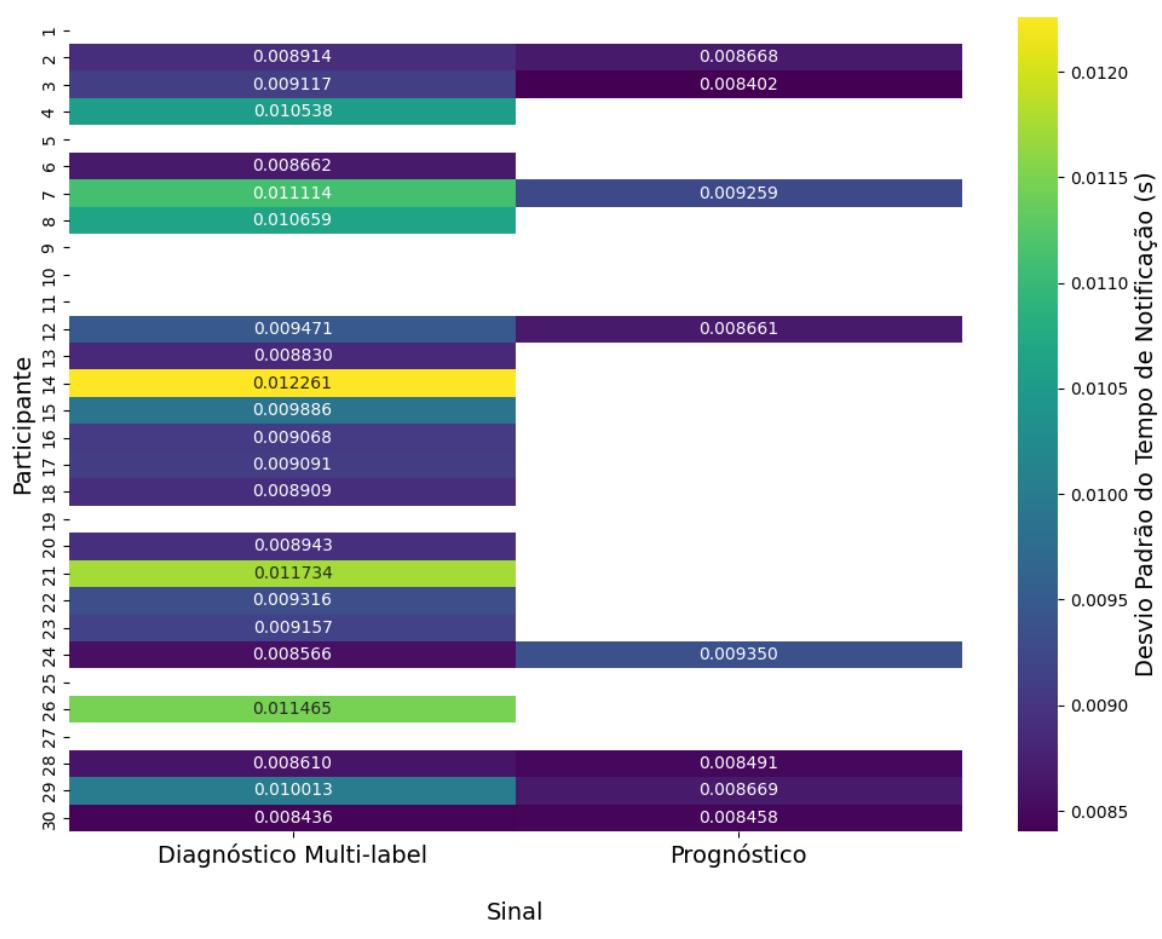
Fonte: Elaborado pela autora (2025).

Figura 37 – *Heatmap* dos tempos de notificação (s) para Prognóstico e Diagnóstico Multi-label em diferentes sujeitos monitorados no Sistema Multiagente (SMA).



Fonte: Elaborado pela autora (2025).

Figura 38 – *Heatmap* do desvio padrão dos tempos de notificação(s) para Diagnóstico Multi-label e Prognóstico, em diferentes sujeitos monitorados no Sistema Multiagente (SMA).



Fonte: Elaborado pela autora (2025).

6.13 VALIDAÇÃO EXPERIMENTAL E ANÁLISE POR MATRIZ DE CONFUSÃO

Enquanto os testes anteriores consideravam a média dos valores aferidos ao longo de dez dias para cada sujeito individualmente, o teste apresentado nesta seção abrange a média por dia das dez medições realizadas em cada um dos trinta sujeitos, totalizando 300 registros. Em cada dia de experimento, o SMACare realizou inferências para os quatro sinais fisiológicos (SpO_2 , Temperatura, Pressão Arterial Estimada e Frequência Cardíaca), para o conjunto combinado dos quatro sinais no diagnóstico multi-label e para o histórico individual utilizado na tarefa de prognóstico. Os resultados obtidos estão apresentados nas Figuras 39 a 44 em formato de matriz de confusão.

A Figura 39 ilustra a matriz de confusão para a classificação do sinal BPM (frequência cardíaca) em três categorias: Bradicardia, Normal e Taquicardia. A matriz foi construída com base em 10 medições por participante, totalizando 300 amostras provenientes de 30 indivíduos. Os valores representam as contagens de classificações corretas e incorretas entre as classes reais (eixo Y) e previstas (eixo X), permitindo avaliar o desempenho do modelo no reconhecimento de padrões cardíacos em um cenário de monitoramento multiusuário.

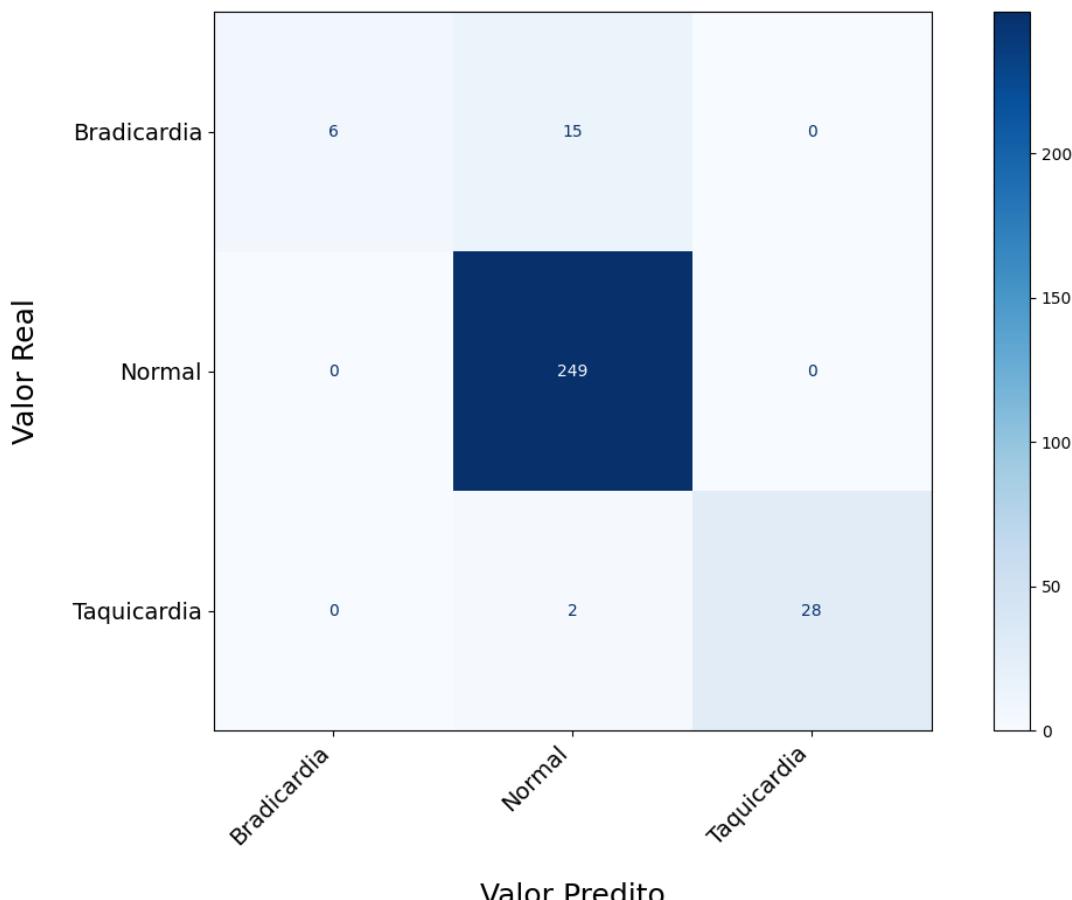
Observa-se que a classe *Normal* apresenta um bom desempenho, com 249 acertos e ausência total de classificações incorretas para as outras categorias, refletindo um recall de 100% para essa classe. Essa sensibilidade indica que o modelo é confiável em reconhecer estados normais de frequência cardíaca, o que é desejável para reduzir alarmes falsos positivos em condições normais. Em contraste, a classe *Bradicardia* apresentou 6 acertos e 15 classificações equivocadas como *Normal*, resultando em um recall modesto de apenas 28,6%. Esse baixo desempenho se deve a valores de *Bradicardia* muito próximos do *Normal*, por exemplo, valores 59,9 que são tidos como *Bradicardia* e 60 como *Normal*, a inferência algumas vezes interpretou 59,9 como já sendo um valor *Normal*. Isso sugere que o modelo tem dificuldade em capturar os padrões mais sutis e próximos, necessitando de calibração.

A classe *Taquicardia* obteve 28 acertos com apenas 2 erros (classificados como *Normal*), atingindo um recall de aproximadamente 93,3%, o que demonstra boa capacidade do modelo em identificar eventos de frequência cardíaca elevada. Esse resultado indica que a taquicardia apresenta padrões mais discrimináveis no conjunto de características, ou que os dados de treinamento foram mais representativos para essa classe.

De forma geral, a acurácia global calculada sobre os 300 exemplos foi elevada (aproximadamente 94,3%). A análise da matriz de confusão evidencia a necessidade de estratégias para melhorar o recall em classes minoritárias ou com valores próximos, como técnicas de balançamento de dados, ajuste de limiares de decisão, uso de perdas ponderadas ou estratégias de aprendizado focalizado. Essas medidas podem garantir que o SMACare forneça alertas clínicos minimamente enviesados para todos os estados fisiológicos de interesse.

A Figura 40 ilustra a matriz de confusão para a classificação do sinal BPT (pressão arterial estimada) em três categorias: Hipertensão, Hipotensão e Normal. A matriz foi gerada a

Figura 39 – Matriz de confusão para a classificação do sinal BPM (frequência cardíaca) em três categorias: Bradicardia, Normal e Taquicardia

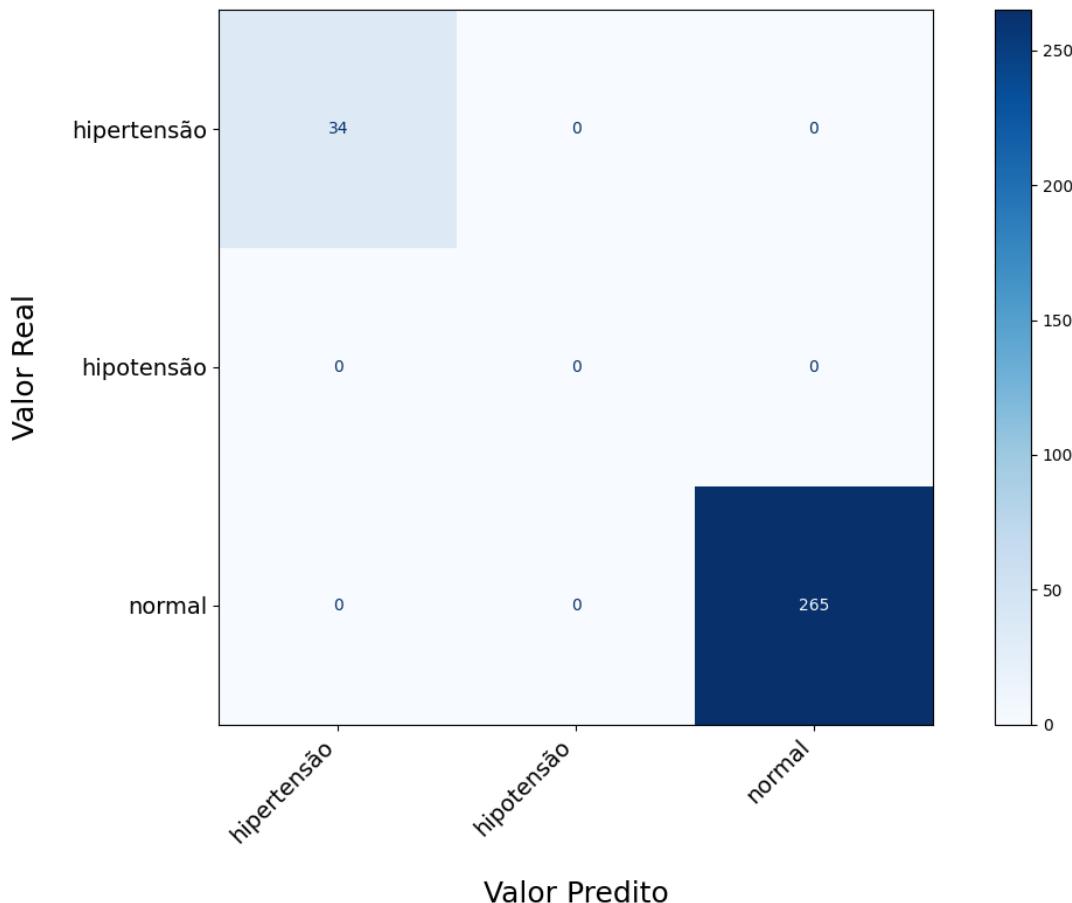


Fonte: Elaborado pela autora (2025).

partir de 10 medições por participante, totalizando 300 amostras de 30 indivíduos. Os valores indicam as contagens de classificações corretas e incorretas entre as classes reais (eixo Y) e previstas (eixo X), permitindo avaliar o desempenho do modelo na detecção de padrões de pressão arterial em um contexto de monitoramento multiusuário. Nessa matriz, duas classes se destacam: *Hipertensão* e *Normal*, enquanto *Hipotensão* está ausente, sem exemplos reais ou previstos no conjunto de teste. Essa ausência inviabiliza o cálculo de métricas para a classe hipotensão e evidencia uma limitação no conjunto de dados dos participantes da pesquisa, sugerindo a necessidade de coleta com mais participantes que apresentem essa condição clínica mais evidente.

Para a classe *Hipertensão* o modelo obteve ótimo desempenho, com 34 acertos e recall de 100%, não cometendo erros de classificação para nenhuma outra categoria. Isso indica que os padrões associados à hipertensão arterial foram bem aprendidos, resultando em alta sensibilidade e especificidade. Esse resultado é positivo do ponto de vista clínico, pois erros de omissão nesse estado poderiam atrasar intervenções em pacientes hipertensos. De forma semelhante, a classe *Normal* também apresentou desempenho ótimo, com 265 acertos e recall igualmente de 100%, sem qualquer confusão com as outras classes. Essa separação clara sugere que o modelo é

Figura 40 – Matriz de confusão para a classificação do sinal BPT (pressão arterial estimada) em três categorias: Hipertensão, Hipotensão e Normal.



Fonte: Elaborado pela autora (2025).

capaz de discriminar com segurança entre pacientes com pressão arterial normal e aqueles com hipertensão, garantindo elevada confiabilidade para aplicações de triagem ou monitoramento remoto.

A acurácia global para o conjunto de teste, considerando as duas classes efetivamente presentes, é de 100%, refletindo um desempenho sem erros. No entanto, essa métrica deve ser interpretada com cautela. A ausência de exemplos de *Hipotensão* do conjunto de teste impede qualquer avaliação realista do modelo para essa condição. Embora os resultados para *Hipertensão* e *Normal* sejam excelentes, a análise da matriz de confusão destaca a necessidade de incluir amostras representativas de *Hipotensão* na validação. Sem isso, o SMACare pode apresentar lacunas na sua capacidade de detecção abrangente de anomalias da pressão arterial, limitando sua aplicação como ferramenta de suporte à decisão clínica.

Observando a matriz de confusão gerada para a tarefa de classificação do sinal SpO₂ (Figura 41), a classe *Nível Bom de Saturação* apresenta o melhor desempenho, com 202 acertos e nenhuma predição incorreta para outras classes, resultando em recall de 100%. Esse resultado indica que o modelo é bem confiável para identificar estados normais de oxigenação, o que é importante para reduzir falsos alarmes em monitoramento contínuo.

Para *Nível Baixo de Saturação*, o desempenho também é bom, com 13 acertos sem qualquer confusão com outras categorias (recall de 100%). A classe *Nível Médio de Saturação*, por exemplo, obteve 59 acertos mas também 20 casos classificados incorretamente como *Nível Bom de Saturação*, reduzindo seu recall para aproximadamente 74,7%. Essa confusão reflete a proximidade fisiológica entre níveis médios e bons de saturação, o que dificulta a separação precisa com base nas características extraídas.

Para a classe *Hipoxemia*, foi registrado apenas um único caso, que foi classificado corretamente. Já para *Hipoxemia Severa*, não houve nenhuma instância classificada. A classe *Nível Anormal de Saturação* apresentou quatro casos, todos classificados corretamente com 100% de acurácia.

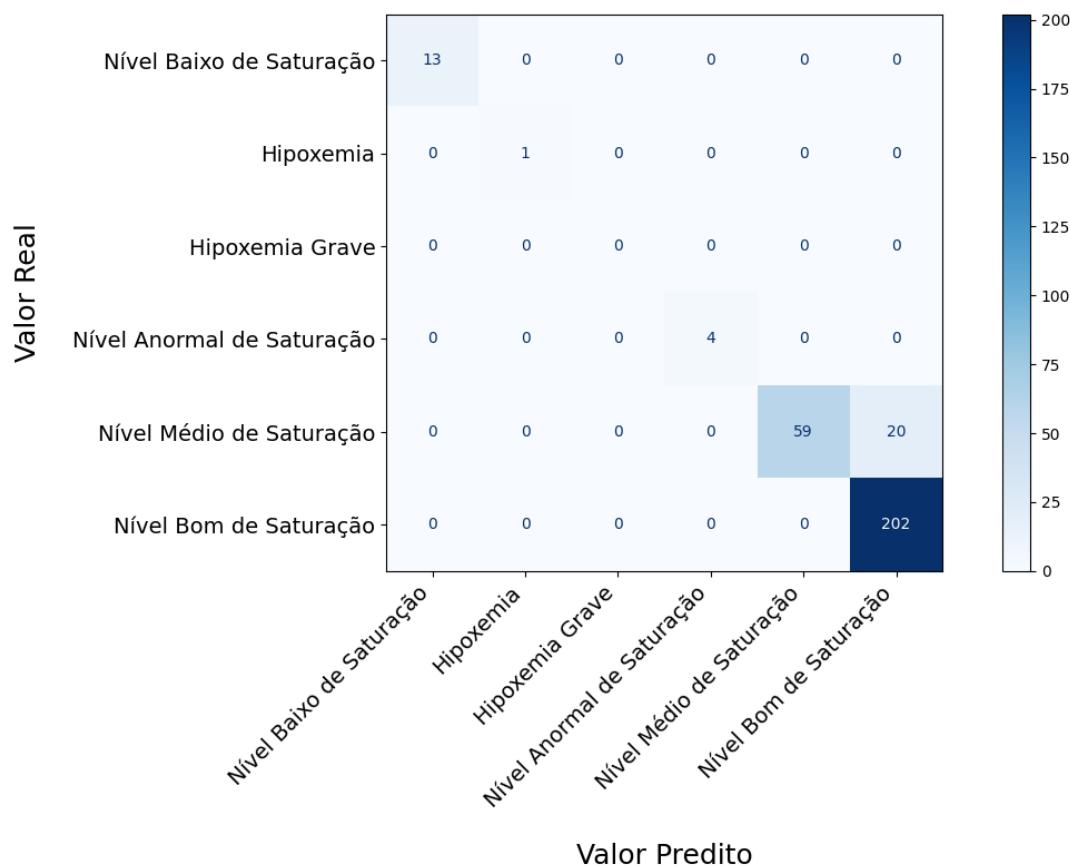
Em termos de acurácia global, o desempenho do modelo apresenta-se elevado (93,3%), reflexo direto da predominância de exemplos da classe *Nível Bom de Saturação* (202/300) no conjunto analisado. A análise detalhada da matriz de confusão mostra que o SMACare foi treinado e avaliado majoritariamente com dados de participantes sem hipóxia significativa, o que explica a ausência de registros para a classe *Hipoxemia Grave* e o número reduzido de casos para *Hipoxemia* (1 caso) e *Nível Anormal de Saturação* (4 casos). Assim, a aparente dificuldade em identificar estados hipóxicos não decorre de falha do modelo, mas reflete a composição amostral do estudo, em que os 30 participantes não incluíram indivíduos com hipóxia severa ou prevalência significativa de hipóxia leve.

A matriz de confusão gerada para a tarefa de classificação de temperatura corporal (Figura 42) demonstra características relevantes sobre o desempenho do modelo. Com um total de 30 participantes e 300 medições, observa-se que várias classes, como *Hipotermia Moderada* e *Hipotermia Severa*, não apresentam registros nem no eixo das verdadeiras classes nem como predições. Essa ausência não reflete erro de classificação do modelo, mas sim ausência real de exemplos no conjunto de dados — ou seja, os participantes simplesmente não exibiram tais condições durante a coleta.

A classe Normal é claramente dominante, com 236 acertos e apenas 2 erros (preditos como *Hipotermia Leve*), resultando em um recall alto de aproximadamente 99,2%. Essa capacidade de detectar o estado normal com elevada sensibilidade é desejável em aplicações de triagem, pois reduz alarmes falsos em situações sem risco. A classe *Febre Leve ou Febrícula* obteve 32 acertos e nenhum erro para outras categorias, garantindo recall de 100%, enquanto *Febre Baixa* obteve 4 acertos sem confusão (recall também de 100%). Já *Febre Alta* e *Hiperpirexia ou Febre Muito Alta* apareceram com apenas um caso cada, ambos classificados corretamente, o que também representa recall de 100%, embora esses números muito baixos impeçam qualquer conclusão estatística robusta. A classe *Hipotermia Leve*, com 21 acertos e 3 erros classificados como Normal, apresentou recall de aproximadamente 87,5%, demonstrando bom desempenho geral, mas com alguma sobreposição entre os padrões térmicos de normalidade e leve hipotermia. Essa confusão pode refletir a proximidade fisiológica entre temperaturas limítrofes, exigindo ajustes no modelo ou a incorporação de dados adicionais para melhor discriminação.

Em termos de acurácia global, o desempenho do modelo é bom, com 93,6%. Contudo, para uso em cenários clínicos reais, seria essencial coletar dados adicionais que incluam episódios reais de hipotermia moderada ou grave, além de febres extremas, permitindo validar o modelo para detecção confiável de todo o espectro clínico.

Figura 41 – Matriz de confusão para a classificação do sinal SpO2 em seis categorias.

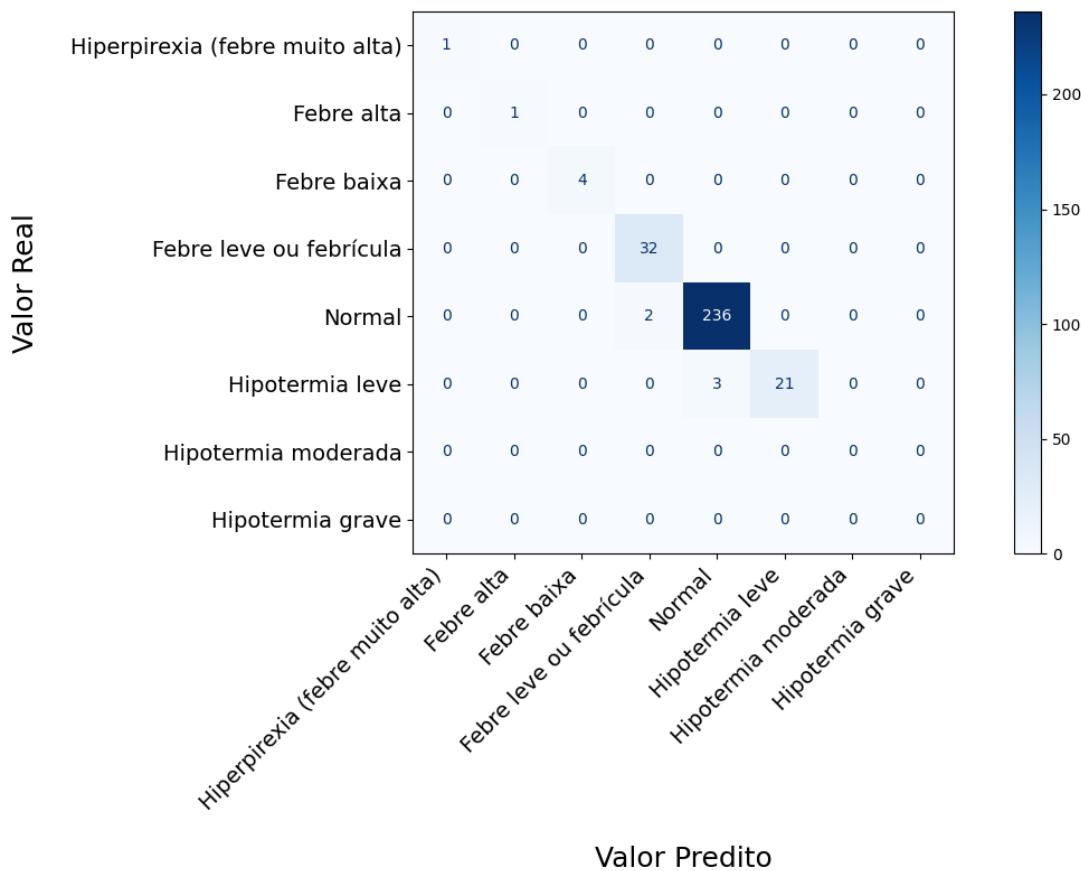


Fonte: Elaborado pela autora (2025).

A matriz de confusão referente ao diagnóstico multi-label (Figura 43) revela a capacidade do modelo em identificar corretamente uma ampla gama de condições clínicas a partir da inferência sobre múltiplos sinais fisiológicos. Observa-se uma forte predominância da classe *Sem Anormalidades*, com 209 acertos e nenhum erro de classificação, o que representa um recall de 100% para essa classe. Esse resultado indica uma ótima sensibilidade na detecção do estado de normalidade, o que é desejável para reduzir alarmes falsos positivos em monitoramento contínuo.

Outras classes com bom desempenho incluem *Taquicardia Sinusal Isolada*, com 23 acertos e nenhum erro (recall de 100%), e *Condição Hipertensiva*, com 19 acertos (recall também de 100%). Ainda que essas condições apresentem frequência inferior à classe *Sem Anormalidades*, o modelo mostrou-se eficaz em identificá-las corretamente. Da mesma forma, observam-se resultados positivos em condições como *Insuficiência Respiratória Progressiva* (12 acertos), *Pressão de Pulso Ampla* (11 acertos), *Síndrome Hipermetabólica Leve* (4 acertos) e *Febre Moderada Isolada* (6 acertos), todas com predições corretas, resultando em recall de 100%

Figura 42 – Matriz de confusão para a classificação do sinal Temperatura Corporal em oito categorias.



Fonte: Elaborado pela autora (2025).

para cada uma delas.

Em contrapartida, algumas classes — embora tenham exemplos no conjunto de teste — apresentaram desempenho inferior. A classe *Hipertensão Sistólica Isolada*, por exemplo, apresentou apenas 1 acerto entre 3 ocorrências, indicando um recall de aproximadamente 33%, com confusões principalmente com a classe *Sem Anormalidades*. Esses dois valores eram valores limítrofes, indicando que o sistema apresenta dificuldade na inferência neste cenário. É importante destacar que diversas classes clínicas — como *Infecção Grave/Choque Séptico*, *Hipotermia/Má Nutrição Graves*, *Reflexo de Cushing*, *Insuficiência Respiratória*, entre outras — não apresentaram exemplos no conjunto de teste, resultando em linhas zeradas na matriz de confusão. Isso reflete a ausência real desses quadros clínicos na amostra coletada, e não falhas do modelo em predizê-los.

A classe *Bradicardia Sinusal* apresentou um ótimo desempenho, com todos os 9 casos identificados corretamente. Isso sugere que os padrões relacionados estão bem definidos no espaço de características utilizado. Como se trata de um diagnóstico com importância clínica significativa e moderada prevalência, o desempenho do modelo é bastante promissor para uso em monitoramento em tempo real. Quanto a *Crise de Ansiedade*, o modelo identificou um

único caso. Apesar de ser uma classe com baixa ocorrência, a predição correta mostra que o modelo foi capaz de reconhecer os padrões associados, mesmo em condições psicofisiológicas não necessariamente caracterizadas por alterações fisiológicas extremas. Este é um resultado interessante, pois indica sensibilidade do modelo a alterações sutis nos sinais monitorados, como frequência cardíaca ou SpO₂, que podem acompanhar estados ansiosos.

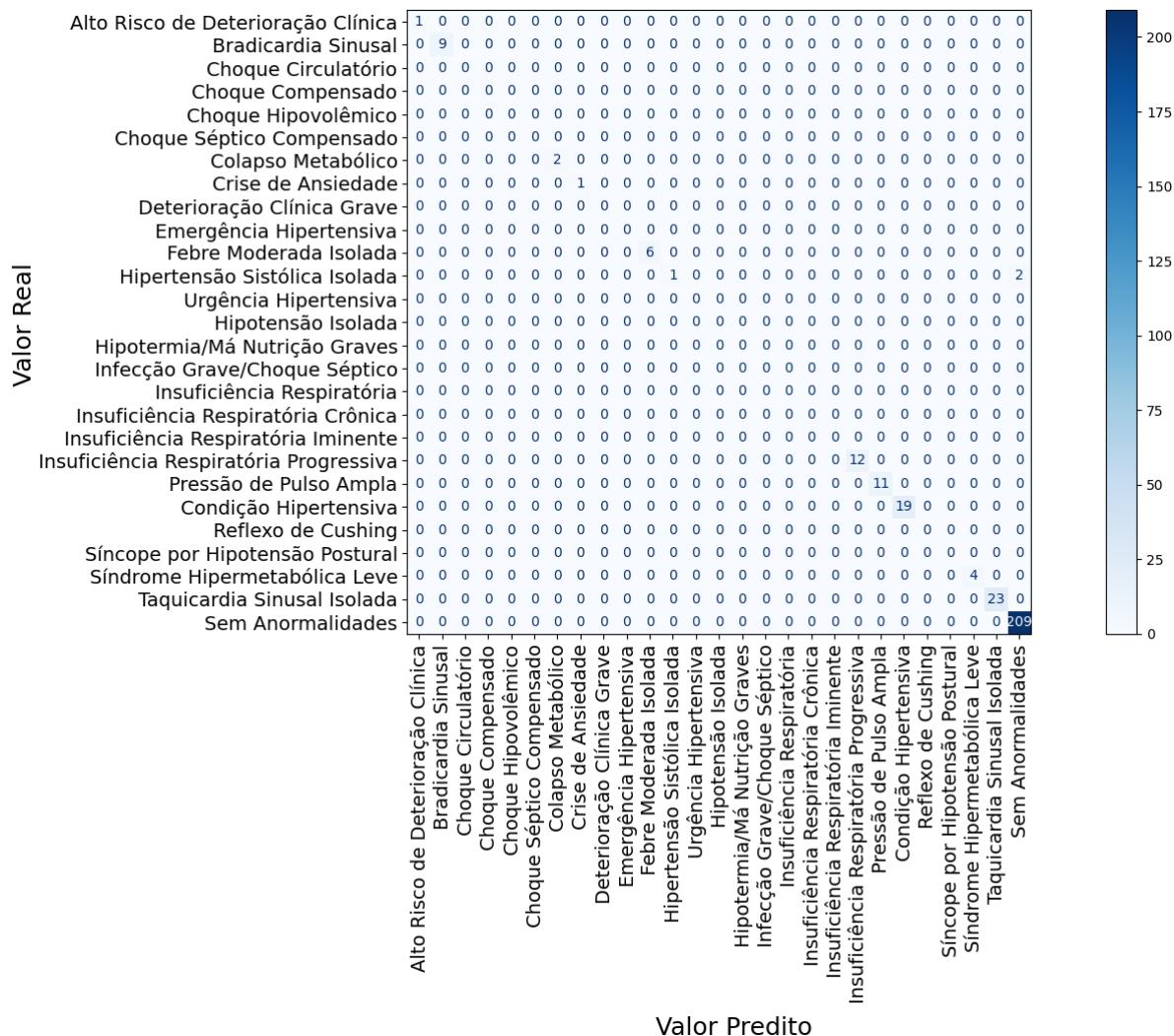
Com dois casos corretamente classificados, a classe *Colapso Metabólico* apresentou desempenho perfeito tanto em recall quanto em precisão. O modelo foi capaz de identificar corretamente ambas as instâncias sem gerar classificações equivocadas para outras classes, o que é um resultado promissor para uma condição de alta gravidade clínica. O reconhecimento preciso de *Colapso Metabólico* é fundamental para sistemas de suporte à decisão médica, dado que essa condição pode rapidamente evoluir para choque ou falência de múltiplos órgãos. Um dos participantes da pesquisa em que esse diagnóstico foi inferido pelo SMACare também obteve o diagnóstico pelo sistema de *Insuficiência Respiratória Progressiva*. Durante a escrita desta Tese o participante havia sido entubado e, infelizmente, veio a falecer. Dois dos motivos do falecimento foram falência múltipla de órgãos e pneumonia (Apêndice G). O segundo participante com esta condição passou pela COVID-19 entubado e, atualmente, encontra-se em observação mais intensa devido ao diagnóstico inferido.

A acurácia global do modelo sobre as amostras efetivamente classificadas é elevada, refletindo o bom desempenho em condições majoritárias e moderadamente frequentes. No entanto, como essa métrica é fortemente influenciada pela classe *Sem Anormalidades*, que representa quase 70% do conjunto, é necessário interpretá-la com cautela. Mais relevante, portanto, é a análise da sensibilidade (*recall*) por classe, que evidenciou um bom desempenho. A matriz de confusão apresentada demonstra a robustez do modelo para classes majoritárias e de média frequência, ao mesmo tempo em que revela a necessidade de expansão e balanceamento do conjunto de dados para garantir cobertura adequada e desempenho confiável nas classes críticas, porém raras. Estratégias como aumento da amostra pode ser explorada para fortalecer a capacidade do SMACare em detectar com precisão uma gama mais abrangente de diagnósticos clínicos.

Na classificação de prognóstico, o processo de inferência é realizado em intervalos de cinco minutos e gera apenas um prognóstico por participante, o que faz com que a quantidade total de amostras corresponda diretamente ao número de participantes da pesquisa, ou seja, 30 amostras. A validação dos prognósticos foi realizada com base em critérios extraídos da literatura (Gomes et al., 2016; Silva et al., 2018; Martinelli, 2019; Borges et al., 2023; Patel; Albert, 2024; MSD Manuals, 2024), considerando os desfechos clínicos esperados na ausência de tratamento para os diagnósticos inferidos em tempo real. Observou-se que os prognósticos apresentaram maior convergência com as condições autorelatadas pelos participantes do que os diagnósticos obtidos por meio da abordagem multi-label, fator que também foi considerado na análise de corretude do modelo prognóstico.

Algumas classes prognósticas previstas pelo sistema apresentam ausência total de instâ-

Figura 43 – Matriz de confusão para a classificação de Diagnóstico Multi-label.

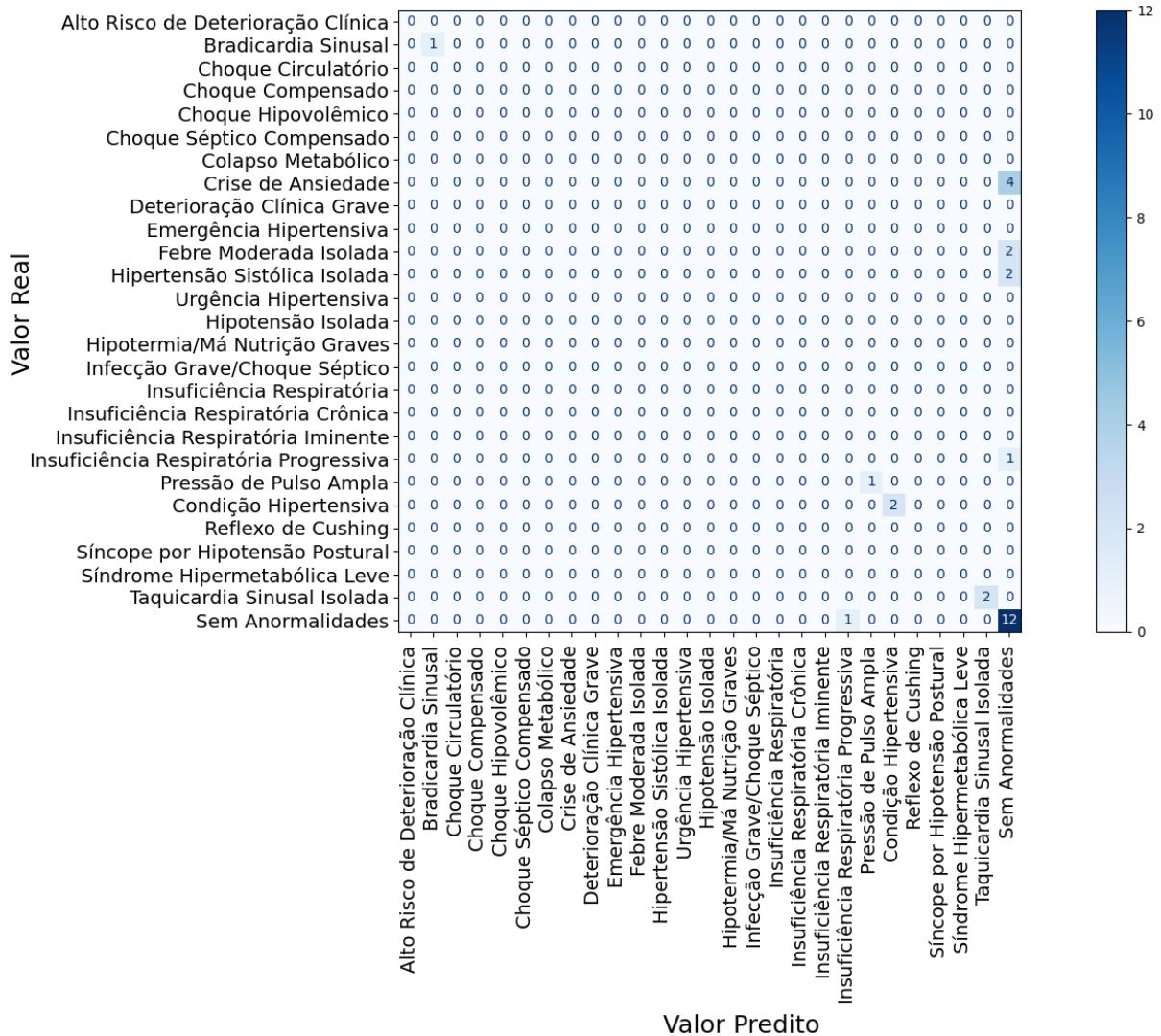


Fonte: Elaborado pela autora (2025).

cias reais no conjunto de teste, indicando que nenhum participante manifestou alguma dessas condições durante o período de coleta. Entre as classes efetivamente representadas, destaca-se a classe *Sem Anormalidades*, com 12 acertos entre 13 ocorrências reais, resultando em um recall de aproximadamente 92,3%. Trata-se do melhor desempenho numérico entre as classes com múltiplos exemplos, refletindo que o modelo é capaz de reconhecer estados estáveis com elevada confiabilidade, o que é desejável para reduzir falsos positivos. O único erro nesta classe foi uma predição equivocada para *Insuficiência Respiratória Iminente*, possivelmente influenciada por situação limítrofe.

Classes que destacam-se pelo erro na inferência são *Crise de Ansiedade*, apresentou o maior número de exemplos no conjunto de teste (4) e foi classificada incorretamente em todas as instâncias, *Hipertensão Sistólica Isolada* (2 erros de 2 amostras), *Febre Moderada Isolada* (2 erros de 2 amostras). Essas classes foram todas inferidas como Normal devido a valores limítrofes, indicando novamente um ponto fraco do SMA. Outras classes que pode-se citar são *Taquicardia Sinusal Isolada* (2 acertos) e *Condição Hipertensiva* (2 acertos), todas

Figura 44 – Matriz de confusão para a classificação de Prognóstico.



Fonte: Elaborado pela autora (2025).

com desempenho 100%. Embora o número de amostras para essas categorias seja reduzido, o modelo demonstrou alta sensibilidade a padrões específicos, mesmo em cenários com baixa frequência relativa. Outras classes, como *Bradicardia Sinusal* e *Pressão de Pulso Ampla*, foram corretamente identificadas com apenas um exemplo cada (recall de 100%), o que é positivo. Assim sendo, a matriz de confusão de prognóstico evidencia um desempenho satisfatório do modelo nas condições efetivamente presentes na amostra, com uma acurácia global de 86,7% e um recall macro médio de 94,2%. Esses valores indicam que o sistema é capaz de inferir corretamente a maioria dos prognósticos clínicos com boa sensibilidade, especialmente nas categorias mais frequentes ou menos ambíguas.

7 CONSIDERAÇÕES FINAIS

7.1 DISCUSSÕES GERAIS

Na seção 2 deste trabalho foram apresentados e discutidos os principais trabalhos relacionados ao tema. Nenhuma das abordagens analisadas, entretanto, contempla de forma integrada todos os aspectos propostos pelo SMACare. Algumas iniciativas apresentam interesses, características e objetivos convergentes com esta pesquisa. Diante do exposto, um breve comparativo entre os trabalhos apresentados e o trabalho desta tese é demonstrado nos Quadros 10 e 11, o que possibilitou verificar as lacunas existentes na literatura. A partir do Quadro 10, observa-se que o trabalho desta pesquisa se diferencia das abordagens analisadas em diversos aspectos centrais para aplicações de monitoramento de saúde mediado por sistemas multiagentes e inteligência artificial.

Primeiramente, no domínio de aplicação, grande parte dos trabalhos revisados se limita a contextos específicos ou simulados. Por exemplo, (Humayun et al., 2022) foca em assistência remota para idosos com ênfase em mobilidade urbana, enquanto (Haitam; Najat; Jaafar, 2023) simula interações populacionais para prever pandemias. Os autores de (Shaik; Kumar; Kaur, 2023) propõem um sistema de intervenção em tempo real, mas operando com dados simulados de sensores vestíveis. Em contraste, este trabalho apresenta um escopo mais amplo e integrado, ao abordar tanto o diagnóstico em tempo real quanto o prognóstico histórico, viabilizando cuidados personalizados em saúde individual de forma contínua.

No aspecto técnico de *Machine Learning*, observa-se uma variação entre os trabalhos. Alguns não incorporam aprendizado de máquina ((Humayun et al., 2022), (Marcon et al., 2017)), outros utilizam abordagens específicas como heurísticas ou priorização ((Mutlag et al., 2021)). Abordagens mais avançadas, como (Shaik; Kumar; Kaur, 2023), usam *Deep Reinforcement Learning* para sinais vitais individuais, mas sem previsão histórica. Em (Hou et al., 2025) é explorado o uso de LLMs para prognóstico multimodal em Alzheimer, porém sem tratar monitoramento contínuo via sensores vestíveis. Em contraponto, o trabalho desta tese propõe uma solução MLP integrada em subagentes especializados, permitindo tanto classificação em tempo real quanto prognóstico histórico, com consistência entre os cenários.

Outro diferencial marcante está na arquitetura multiagente. Enquanto várias propostas adotam agentes homogêneos e pouco adaptativos (Humayun et al., 2022; Shaik; Kumar; Kaur, 2023), este trabalho implementa uma arquitetura heterogênea e hierárquica, com agentes dinâmicos criados por dispositivo/usuário. Essa escolha promove modularidade, reuso e escalabilidade, viabilizando a gestão de múltiplos dispositivos e cenários distintos dentro do mesmo framework.

Em relação aos cenários suportados, destaca-se que muitos trabalhos se limitam a um único cenário (tempo real ou histórico, ou ainda apenas simulação). Por exemplo, (Shaik; Kumar; Kaur, 2023) opera exclusivamente em tempo real (com dados simulados), enquanto (Haitam; Najat; Jaafar, 2023) é puramente histórico/simulado. O trabalho desta tese cobre dois cenários reais complementares: (i) aquisição de dados em tempo real via BLE com sensores vestíveis

reais, e (ii) análise histórica a partir de arquivos CSV. Essa dualidade permite não apenas o monitoramento imediato, mas também análises preditivas baseadas em histórico clínico.

Na questão de base de dados, nenhum dos trabalhos analisados utiliza efetivamente um banco de dados relacional para organizar dados de forma dinâmica por usuário/dispositivo. Em contrapartida, este trabalho adota um banco PostgreSQL com tabelas dinâmicas e *triggers*, integradas ao sistema multiagente. Isso oferece um grau de rastreabilidade, escalabilidade e auditabilidade que atende melhor a requisitos de sistemas médicos.

Quadro 10 – Comparação entre trabalhos correlatos e o trabalho desta tese – Parte 1

| Trabalho | Domínio | Machine Learning | Agentes | Cenários | Base de Dados |
|-------------------------------|---|--|--|--|--|
| (Humayun et al., 2022) | Assistência remota para idosos e crônicos, com ênfase em mobilidade | Heurística sem DL ou ML | Agentes homogêneos e móveis para envio e alerta | Mobilidade/assistência para idosos em ambiente urbano | Não |
| (Shaik; Kumar; Kaur, 2023) | Intervenção em tempo real com alertas via DRL por sinais vitais individuais | Deep Reinforcement Learning (DRL) com agentes por sinal vital | Agentes DRL homogêneos , cada um para um sinal vital | Tempo real , com dados de sensores wearables simulados | Não |
| (Haitam; Najat; Jaafar, 2023) | Modelagem e simulação populacional para previsão de pandemia COVID-19 | ML supervisionado (árvores, RF, SVM) para prever casos e mortes COVID | SMA simula interações sociais e governamentais | Simulação de pandemia com dados históricos COVID-19 | Não |
| (Hou et al., 2025) | Alzheimer's (diagnóstico/prognóstico multimodal) | LLM + ferramentas DL multimodais | Modular com LLM, Reasoning Engine e ferramentas médicas | Multimodal (MRI/PET) | Não |
| (Mutlag et al., 2021) | Monitoramento ECG em fog-cloud | Priorização | SMA distribuído para fog-cloud | Fog/Cloud com processamento crítico | Não |
| (Marcon et al., 2017) | Home care (logística de rotas para cuidadores) | Não | Simulação de agentes cuidadores/pacientes | Simulação de rotas diárias de cuidado domiciliar | Não |
| Este Trabalho | Monitoramento remoto e diagnóstico em tempo real + prognóstico baseado em histórico . Saúde individual com BLE/CSV, triggers, ML integrado | MLP usando modelos de classificação e prognóstico integrados em subagentes por sinal vital e prognóstico geral | Heterogênea e hierárquica, agentes dinâmicos por dispositivo/usuário | Dois cenários: tempo real via BLE e análise histórica via CSV | PostgreSQL com tabelas dinâmicas por dispositivo/usuário e triggers |

Fonte: Elaborado pela autora (2025).

O Quadro 11 evidencia mais diferenças entre as abordagens analisadas no que diz respeito a funcionalidades essenciais para sistemas de monitoramento em saúde. Os resultados dessa comparação reforçam os aspectos inovadores do trabalho desta tese frente ao estado da arte. Em primeiro lugar, destaca-se o suporte ao prognóstico. A maioria dos trabalhos revisados não contempla mecanismos de prognóstico individual. Por exemplo, (Humayun et al., 2022), (Shaik; Kumar; Kaur, 2023), (Mutlag et al., 2021) e (Marcon et al., 2017) não implementam previsão de evolução de saúde. Mesmo (Haitam; Najat; Jaafar, 2023), embora apresente simulação de

prognóstico epidemiológico, o faz em nível populacional e não individual. Este trabalho, por outro lado, conta com um *PrognosisAgent* dedicado, capaz de oferecer prognóstico personalizado por usuário com base em histórico clínico armazenado, algo que poucas abordagens abordam com essa granularidade.

No aspecto de modularidade, a maioria dos trabalhos analisados permanecem restritos a arquiteturas estáticas. Por exemplo, (Shaik; Kumar; Kaur, 2023) opera com número fixo de agentes por variável monitorada, e (Haitam; Najat; Jaafar, 2023) define um modelo fixo para simulação populacional. Em contrapartida, o SMACare implementa criação dinâmica de *BrokerAgents* e *DBAgents* para cada dispositivo ou usuário, permitindo expansão ou redução conforme demanda. Essa característica confere flexibilidade e modularidade, essencial em cenários reais onde novos dispositivos podem ser incorporados continuamente.

Outro ponto diferencial é o mecanismo de notificação. Enquanto a maioria das pesquisas analisadas não implementam ou têm abordagens limitadas (Shaik; Kumar; Kaur, 2023) oferece apenas alerta MET sem sistema de notificação formal; (Mutlag et al., 2021) e (Marcon et al., 2017) não têm suporte), o SMACare adota um *NotifyAgent* especializado. Esse agente garante envio estruturado de notificações para profissionais de saúde ou usuários, promovendo ação rápida e integração com fluxos de cuidado.

Na questão de histórico clínico, algumas pesquisas analisadas ou não o suportam (como (Humayun et al., 2022), (Mutlag et al., 2021), (Marcon et al., 2017)), ou o tratam apenas de forma agregada ou limitada. Por exemplo, (Haitam; Najat; Jaafar, 2023) armazena histórico em nível agregado populacional para modelagem de políticas, e (Shaik; Kumar; Kaur, 2023) não retém histórico clínico para prognóstico. O trabalho desta tese diferencia-se por criar tabelas específicas por dispositivo/usuário em banco de dados relacional, armazenando não só o histórico de sinais vitais, mas também registros de diagnóstico e prognóstico. Esse grau de persistência e rastreabilidade é fundamental para aplicações clínicas reais, que exigem auditabilidade e personalização.

Em termos de escalabilidade, todas as pesquisas declaradamente suportam escalabilidade de alguma forma. Contudo, a estratégia da pesquisa desta tese se destaca ao empregar criação dinâmica de agentes e banco relacional com *triggers*, permitindo não apenas suportar mais usuários, mas fazê-lo de forma organizada, com consistência e rastreabilidade dos dados. Isso evita problemas comuns em arquiteturas fixas ou monolíticas que sofrem para crescer de forma controlada.

Os resultados experimentais, apresentados em detalhe no Capítulo 6, demonstraram métricas de acurácia compatíveis com aplicações clínicas de monitoramento remoto. A validação realizada com voluntários evidenciou que o SMACare é capaz de detectar anomalias em tempo real, emitir alertas com baixa latência e gerar prognósticos personalizados a partir de dados históricos, apresentando tempos médios de inferência adequados para uso em ambientes de home care e/ou instituições de longa permanência. Observou-se ainda que os diagnósticos apresentaram maior aderência aos sinais fisiológicos monitorados, enquanto os prognósticos mostraram maior

Quadro 11 – Comparaçao entre trabalhos correlatos e o trabalho desta tese – Parte 2

| Trabalho | Prognóstico | Modularidade | Notificações | Histórico Clínico | Escalabilidade |
|-------------------------------|--|---|--|--|---|
| (Humayun et al., 2022) | Não | Focado em coleta centralizada | Notificações para médicos | Não | Sim |
| (Shaik; Kumar; Kaur, 2023) | Não, foca apenas em alerta reativo | Não (número fixo de agentes por variável) | Alerta MET, mas sem sistema de notificação | Não | Cada experimento tem número fixo de agentes por variável |
| (Haitam; Najat; Jaafar, 2023) | Simulação epidemiológica (nível populacional) | Não (SMA é modelo fixo para simulação populacional) | Não (resultado da simulação para políticas públicas) | Sim (histórico agregado , nível populacional) | Sim |
| (Hou et al., 2025) | Prognóstico multimodal de AD | Plugáveis | Output interativo com médico | Sim (diagnóstico e prognóstico salvos em tempo de execução) | Sim |
| (Mutlag et al., 2021) | Não | Não | Não | Não | Sim |
| (Marcon et al., 2017) | Não | Agentes autônomos simulados | Não | Não | Sim |
| Este Trabalho | Com PrognosisAgent | Criação dinâmica de BrokerAgents e DBAgents para cada dispositivo/usuário | Sim, via NotifyAgent | Sim: tabelas por dispositivo/usuário, históricos de diagnóstico e prognóstico | Sim |

Fonte: Elaborado pela autora (2025).

correspondência com as condições autorelatadas pelos participantes.

A principal limitação deste trabalho reside no fato de que o sistema desenvolvido trata-se de um protótipo experimental, cujo hardware é composto por sensores de uso não clínico e que não possuem certificação médica. A plataforma proposta não se configura como um sistema de produção ou pronto para uso hospitalar, sendo destinada exclusivamente a fins de pesquisa e validação preliminar. Consequentemente, os resultados gerados pelo sistema não substituem, em nenhuma hipótese, a avaliação médica especializada, tampouco se propõem a exercer papel diagnóstico definitivo. A atuação da equipe médica permanece essencial, sendo o sistema concebido como uma ferramenta de apoio à tomada de decisão clínica e não como um substituto da mesma.

Entre as principais limitações identificadas neste estudo, destaca-se que, embora o sistema tenha sido avaliado com um número significativo de voluntários, ainda são necessários estudos em maior escala, envolvendo diferentes perfis clínicos, a fim de assegurar a robustez e a capacidade de generalização do modelo para ambientes hospitalares e contextos clínicos mais diversos. Outra limitação refere-se ao tratamento de valores limítrofes. Por exemplo, enquanto um valor de SpO₂ = 96,9% pode ser clinicamente interpretado como indicativo de saturação baixa, o sistema tende a classificá-lo como *Saturação Normal*, por estar ligeiramente acima do limiar de corte. Esse comportamento evidencia uma sensibilidade reduzida do modelo para lidar com

valores próximos aos limites de decisão, o que pode impactar a precisão diagnóstica em situações borderline e ressalta a necessidade de estratégias adicionais de refinamento, como a definição de zonas de incerteza ou a adoção de abordagens probabilísticas complementares.

A fim de delimitar de forma objetiva as características funcionais e não funcionais do sistema SMACare, bem como suas capacidades e restrições operacionais, apresenta-se, nas Tabelas 12 e 13, um quadro-resumo com os pares "É / Não É , Faz / Não Faz". Esse quadro tem como propósito esclarecer os limites da arquitetura proposta, diferenciando claramente o que está contemplado no projeto atual e o que se encontra fora do seu escopo de implementação e validação.

Tabela 12 – Quadro-resumo do escopo funcional e conceitual do sistema SMACare: delimitação entre o que o sistema é/não é

| É | Não É |
|---|---|
| Um sistema distribuído baseado em arquitetura multiagente | Um sistema monolítico ou centralizado |
| Um sistema de apoio diagnóstico e prognóstico baseado em sinais vitais | Um substituto direto de decisões médicas ou sistema de prescrição automatizada |
| Um sistema que integra inferência neural (MLP, LSTM etc.) para análise de dados fisiológicos | Um sistema baseado exclusivamente em regras ou limiares fixos |
| Um sistema orientado a dispositivos vestíveis de baixo custo (ex.: Arduino Nano BLE Sense Rev2) | Um sistema restrito a sensores comerciais ou dispositivos médicos certificados |
| Um sistema que opera em tempo real (cenário 1) e com dados históricos (cenário 2) | Um sistema exclusivo para análise offline ou pós-processamento |
| Um sistema adaptável, escalável e modular com suporte a múltiplos usuários e dispositivos | Um sistema estático, de uso único ou com topologia rígida |
| Um sistema com triggers de banco de dados e armazenamento distribuído (PostgreSQL + CSV) | Um sistema baseado em banco de dados relacional tradicional sem comportamento reativo |
| Um sistema com tabelas dinâmicas para cada par dispositivo-paciente | Um sistema com banco de dados fixo e estrutura única |
| Um sistema com diagnósticos realizados por subagentes especializados (BPMAgent, SpO2Agent etc.) | Um sistema com único agente generalista ou sem modularidade por tipo de sinal |

Fonte: Elaborado pela autora (2025).

Tabela 13 – Resumo das funcionalidades implementadas e das ações não contempladas pelo sistema SMACare, com o objetivo de esclarecer suas capacidades e delimitações práticas.

| Faz | Não Faz |
|---|--|
| Coleta sinais vitais em tempo real via BLE (bpm, SpO ₂ , temperatura, pressão arterial estimada) | Coleta de sinais invasivos ou que requerem calibração médica específica |
| Armazena dados automaticamente em microSD e PostgreSQL | Armazenamento em nuvem pública ou com dependência de internet |
| Executa inferência neural embarcada e em ambiente distribuído (via SMA com JADE + Keras) | Treinamento dos modelos durante o uso (apenas inferência é executada em tempo de execução) |
| Notifica eventos críticos por meio do agente NotifyAgent | Envio de notificações via SMS, e-mail ou integração direta com prontuários eletrônicos |
| Suporta múltiplos pacientes e dispositivos com criação automática de tabelas por par (dv/sub) | Compartilhamento simultâneo de um mesmo dispositivo com múltiplos usuários |
| Classifica condições como taquicardia, bradicardia, febre, hipotensão e hipoxia | Diagnóstico médico legal ou laudos formais com valor clínico direto |
| Gera histórico e prognóstico com base em séries temporais anteriores | Previsão de eventos em horizonte longo sem base histórica mínima |
| Permite simulação e reprocessamento de dados gravados em arquivos .csv (Cenário 2) | Acesso simultâneo por múltiplos analistas ao mesmo tempo na interface do SMA |
| Integra sensores via BLE com campos DATA_SERVICE e DATA_CHAR | Integração com protocolos proprietários como ANT+, ZigBee ou LoRa sem customização |

Fonte: Elaborado pela autora (2025).

7.2 CONCLUSÃO

As características analisadas demonstram que o SMACare supre lacunas técnicas observadas nas abordagens existentes ao integrar diagnóstico em tempo real com prognóstico histórico, utilizando dados reais de sensores vestíveis, uma arquitetura multiagente heterogênea e hierárquica com criação dinâmica de agentes, notificações estruturadas via NotifyAgent e armazenamento relacional detalhado com histórico clínico por usuário; esses avanços não apenas fortalecem sua contribuição original para o estado da arte em sistemas inteligentes distribuídos para monitoramento em saúde, mas também evidenciam um potencial real de aplicação prática em sistemas de saúde conectados, reforçando sua originalidade, relevância e impacto na melhoria do cuidado e suporte ao diagnóstico médico.

7.3 PRINCIPAIS CONTRIBUIÇÕES DA TESE

De forma a divulgar os conceitos e resultados obtidos com a proposta e, principalmente, submeter seus resultados para uma avaliação crítica da comunidade científica que se ocupa das questões discutidas nesta tese, dois documentos em forma de artigo, dois documentos em forma de capítulo de livro e um artigo completo de periódico foram produzidos, dos quais resultaram em aceite e publicação, sendo eles:

1. FILGUEIRAS, Tatiana Pereira; BERTEMES FILHO, Pedro. *Intelligent agents in biomedical engineering: a systematic review. International Journal of Biosensors & Bioelectronics*, v. 6, p. 123-128, 2020. (Filgueiras; Bertemes-Filho, 2020a).

2. FILGUEIRAS, Tatiana Pereira; TORRES, Caroline Ruella Paiva; BERTEMES-FILHO, Pedro. *Low Cost System for Fall Detection in the Elderly*. In: *IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*, 2020, Cincinnati. Cincinnati: IEEE, 2020. p. 697. (Filgueiras; Torres; Bertemes-Filho, 2020).
3. FILGUEIRAS, Tatiana Pereira; BERTEMES-FILHO, Pedro. *Internet of Things na engenharia biomédica*. In: *Ampliação e Aprofundamento de Conhecimentos nas Áreas das Engenharias*. 1. ed. Atena Editora, 2020. p. 68-76. (Filgueiras; Bertemes-Filho, 2020b).
4. BERTEMES-FILHO, P.; CAVALIERI, R.; FILGUEIRAS, T. P.; CASTANEDA, W. A. C. *Dispositivos vestíveis com bioimpedância elétrica: conceitos e aplicações*. In: LEITE, Cícilia Raquel Maia; REIS, Célia Aparecida dos; BINSFELD, Pedro Canisio; ROSA, Suélia de Siqueira Rodrigues Fleury (Orgs.). *Novas tecnologias aplicadas à saúde: inovação, internet das coisas, horizontes e desafios*. 1. ed. Rio Grande do Norte: Universidade do Estado do Rio Grande do Norte, 2021. v. 4, p. 285-308. (Bertemes-Filho et al., 2021).
5. FILGUEIRAS, Tatiana Pereira; BERTEMES-FILHO, Pedro; NOVELETTO, Fabrício. *Evaluating the accuracy of low-cost wearable sensors for healthcare monitoring. Micro-machines*, v. 16, n. 7, p. 791, 2025. Disponível em: <<https://doi.org/10.3390/mi16070791>>. (Filgueiras; Bertemes-Filho; Noveletto, 2025).

7.4 TRABALHOS FUTUROS

Mesmo com os resultados promissores obtidos, existem diversas oportunidades para evolução do sistema proposto. Como trabalhos futuros, destacam-se:

- Investigar como outros modelos testados, que também apresentaram bons níveis de acurácia durante o treinamento, se comportam em situações-limite, como nos casos dos sinais de SpO₂, em que valores de 96,9 foram classificados como normais, embora o limite clínico estabelecido seja de 97 em diante;
- Integração de sensores para detecção de quedas, com a criação do FallAgent;
- Adicionar novos prognósticos na IA do PrognosisAgent como, por exemplo, *Falência Múltipla* em caso de Hipoxemia não tratada;
- Reestruturação do DBAgent, subdividindo suas funções em agentes especializados, devido à alta demanda de operações de leitura e escrita;
- Desenvolvimento de uma interface gráfica amigável para usuários e profissionais de saúde;

- Inclusão do Cenário 3 – Sistema Multiagente "mais" distribuído: Cada dispositivo/usuário possuiria instâncias dedicadas do *AnalysisAgent* e seus respectivos subagentes, exigindo infraestrutura em múltiplos servidores e comunicação entre nós distribuídos;
- Inclusão do Cenário 4 – Integração com outros dispositivos vestíveis: O sistema seria adaptado para operar com *wearables* distintos dos utilizados nesta pesquisa, exigindo maior generalização na leitura e interpretação dos dados;
- Incorporação de agentes móveis para suportar a comunicação entre múltiplos servidores no cenário distribuído;
- Criação de dois subgrupos de participantes com saúde autorelatada: em tratamento e sem tratamento, para fins de comparação mais refinada;
- Incluir transmissão orientada a eventos, a fim de economizar bateria;
- Criar um modelo abstrato para especificação de dispositivos de aquisição de sinais vitais, com o objetivo de desacoplar a arquitetura do sistema em relação aos modelos físicos e marcas específicas dos sensores utilizados;
- Definição de modelos de falhas para os sensores integrados, com vistas à detecção precoce de desvios operacionais, degradação de precisão, ou falhas de transmissão;
- Automatizar o envio do arquivo .csv do dispositivo para o SMA;
- Realização de testes com mais participantes em um intervalo de tempo maior a fim de aumentar a robustez e a capacidade de generalização dos modelos de inferência.

REFERÊNCIAS

- ABDELMONEIM, Ahmed A. et al. **Comparative Analysis of Soil Moisture- and Weather-Based Irrigation Scheduling for Drip-Irrigated Lettuce Using Low-Cost Internet of Things Capacitive Sensors**. MDPI, 2025. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/25/5/1568>>. Citado na página 34.
- ABRAHAMS, John; SMITH, Laura; ZHANG, Wei. **A Comparative Study of Optimization Algorithms in Deep Learning**. [S.l.]: JMLR, 2016. v. 17. 1–20 p. Citado 2 vezes nas páginas 53 e 54.
- ADIONO, Trio et al. **Cloud System Design using AMQP Protocol for Smart Devices System Applications**. [s.n.], 2020. Disponível em: <<https://www.osf.io/z824j>>. Citado na página 29.
- ALIAS, M. N. A. M.; MOHYAR, S. N. **Architectural design proposal for real time clock for wireless microcontroller unit**. [s.n.], 2017. v. 162. 01072-01072 p. Disponível em: <<https://doi.org/10.1051/epjconf/201716201072>>. Acesso em: 1 de janeiro de 2023. Citado na página 44.
- Analog Devices. **Wearable Optical Measurement Solution Based on the MAX32664**. 2021. Disponível em: <<https://www.analog.com/en/resources/technical-articles/wearable-optical-measurement-solution-based-on-the-max32664.html>>. Citado 2 vezes nas páginas 36 e 37.
- ARAS, M. S. M. et al. **Analysis performances of Laser Range Finder and blue LED for Autonomous Underwater Vehicle (AUV)**. [s.n.], 2016. Disponível em: <<https://doi.org/10.1109/usys.2016.7893913>>. Acesso em: 17 de junho de 2024. Citado na página 42.
- Arduino. **SD — Control SD cards**. 2019. Disponível em: <<https://www.arduino.cc/reference/en/libraries/sd/>>. Citado 2 vezes nas páginas 43 e 44.
- Arduino Forum. **Discussion on Collecting and Analyzing Accelerometer Data on Arduino Nano BLE Sense**. 2021. Disponível em: <<https://forum.arduino.cc/t/discussion-on-collecting-and-analyzing-accelerometer-data-on-arduino-nano-ble-sense/720892>>. Acesso em: 17 de junho de 2024. Citado na página 42.
- Arduino Forum. **Reading efficiently all IMU sensors in Arduino Nano 33 BLE Sense**. 2023. Disponível em: <<https://forum.arduino.cc/t/reading-efficiently-all-imu-sensors-in-arduino-nano-33-ble-sense/946531>>. Acesso em: 17 de junho de 2024. Citado na página 42.
- ARSHI, Oroos; MONDAL, Surajit. **Advancements in sensors and actuators technologies for smart cities: a comprehensive review**. [S.l.: s.n.], 2023. v. 1. Citado na página 34.
- BARRAGÁN, Hernando. **Wiring: Prototyping Physical Interaction Design**. Dissertação (Master's thesis) — Interaction Design Institute Ivrea, Ivrea, Italy, 2004. Citado na página 40.
- BENGIO, Yoshua. **Learning deep architectures for AI**. [S.l.]: Now Publishers, 2009. v. 2. 1–127 p. Citado na página 94.

BERTEMES-FILHO, Pedro et al. **Dispositivos vestíveis com bioimpedância elétrica: conceitos e aplicações**. Rio Grande do Norte: Universidade do Estado do Rio Grande do Norte, 2021. 285–308 p. Disponível em: <<https://portal.uern.br/wp-content/uploads/sites/14/2024/09/E-book-Novas-Tecnologias-Aplicadas-a-Saude.pdf>>. Citado na página 130.

BIGLARI, Amin; TANG, Wei. **A Review of Embedded Machine Learning Based on Hardware, Application, and Sensing Scheme**. [s.n.], 2023. v. 23. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/23/4/2131>>. Citado 2 vezes nas páginas 33 e 39.

Bluetooth SIG. **Bluetooth Core Specification 4.2**. 2014. <<https://www.bluetooth.com/specifications/>>. San Ramon, CA: Bluetooth Special Interest Group. Acesso em: 27 maio 2025. Citado 4 vezes nas páginas 19, 28, 40 e 41.

BLUM, Jeremy. **Exploring Arduino: Tools and Techniques for Engineering Wizardry**. 2. ed. [S.l.]: Wiley, 2019. Cobertura de novos projetos, conectividade e IoT. ISBN 978-1119405375. Citado na página 40.

BOND, Alan H; GASSER, Les. **Readings in distributed artificial intelligence**. [S.l.]: Morgan Kaufmann, 2014. Citado na página 49.

BORGES, L. A. et al. **Ausência de alteração no quociente respiratório durante a dependência do suprimento de oxigênio no choque endotóxico: subanálise de um estudo experimental controlado**. [s.n.], 2023. v. 35. 281–289 p. Disponível em: <<https://www.scielo.br/j/ccsci/a/XZD9pNdHjc5SVGFrhrGQrHN/?format=pdf&lang=pt>>. Citado na página 121.

CAIN, P. **Thin and Light Dual-Cell OLCDs: Bringing Ultrahigh Contrast Performance to TVs and Automotive Displays**. [s.n.], 2021. v. 37. 20-24 p. Disponível em: <<https://doi.org/10.1002/msid.1212>>. Acesso em: 17 de junho de 2024. Citado na página 43.

CHAI, Roderick; GHADZIATHANASIOU, Theodoros; BLACK, John. **Design and Development of a Low Cost, Non-Contact Infrared Thermometer with Distance Compensation Using the MLX90614 and HC-SR04**. [S.l.: s.n.], 2021. v. 21. 3817 p. Citado na página 38.

CHENG, Yuemeng et al. **Recent developments in sensors for wearable device applications**. [S.l.: s.n.], 2021. v. 413. 875–899 p. Citado na página 42.

CHO, Keuchul et al. **Analysis of Latency Performance of Bluetooth Low Energy (BLE) Networks**. [s.n.], 2015. v. 15. 59–78 p. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/15/1/59>>. Citado 2 vezes nas páginas 19 e 29.

CHOLLET, François. **Deep Learning with Python**. Second edition. Shelter Island, NY: Manning Publications, 2021. ISBN 9781617296864. Citado 2 vezes nas páginas 55 e 56.

COSTANZO, Sandra; FLORES, Alexandra Macarena. **A Non-Contact Integrated Body-Ambient Temperature Sensors Platform to Contrast COVID-19**. [S.l.: s.n.], 2020. v. 9. 1658 p. Citado 2 vezes nas páginas 38 e 39.

COULOURIS, George et al. **Distributed Systems: Concepts and Design**. 5th. ed. USA: Addison-Wesley Publishing Company, 2011. ISBN 0132143011. Citado 4 vezes nas páginas 19, 26, 27 e 28.

CROY, J. R.; ABOUIMRANE, A.; ZHANG, Z. **Next-generation lithium-ion batteries: The promise of near-term advancements.** [s.n.], 2014. v. 39. 407-415 p. Disponível em: <<https://doi.org/10.1557/mrs.2014.84>>. Acesso em: 17 de junho de 2024. Citado na página 45.

DAUER, Michael. **IoT Sensors and Actuators: A Critical Review of Use Cases.** [s.n.], 2021. Revisão crítica de casos de uso reais e hipotéticos em IoT. Disponível em: <https://www.researchgate.net/publication/371119710_IoT_Sensors_and_Actuators_A_Critical_Review_of_Use_Cases>. Citado 3 vezes nas páginas 19, 27 e 34.

DEMROZI, F. et al. **Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey.** [s.n.], 2020. v. 8. 210816-210836 p. Disponível em: <<https://doi.org/10.1109/access.2020.3037715>>. Acesso em: 17 de junho de 2024. Citado 2 vezes nas páginas 41 e 42.

DOMÍNGUEZ-BOLAÑO, Tomás et al. **An overview of IoT architectures, technologies, and existing open-source projects.** [S.l.: s.n.], 2024. Citado na página 40.

DOURISH, Paul; BELL, Genevieve. **"Resistance is futile": Reading science fiction alongside ubiquitous computing.** [S.l.: s.n.], 2014. v. 18. 769–778 p. Citado 2 vezes nas páginas 26 e 28.

DUTTA, R. et al. **Smart and Secure Fingerprint Attendance System using Arduino UNO with GSM Alert.** [s.n.], 2020. Disponível em: <<https://doi.org/10.1109/iciss49785.2020.9316127>>. Acesso em: 17 de junho de 2024. Citado 2 vezes nas páginas 44 e 45.

ELRON, Eyal et al. **Overestimation of Oxygen Saturation Measured by Pulse Oximetry in Hypoxemia. Part 1: Effect of Optical Pathlengths-Ratio Increase.** [S.l.: s.n.], 2023. v. 23. 1434 p. Citado na página 35.

FERRAND, Adrien; POLEKSIC, Jelena; RACINE, Eric. **Factors Influencing Physician Prognosis: A Scoping Review.** SAGE Publications, 2022. v. 7. 23814683221145158 p. Disponível em: <<https://doi.org/10.1177/23814683221145158>>. Citado na página 98.

FILGUEIRAS, Tatiana Pereira; BERTEMES-FILHO, Pedro. **Intelligent agents in biomedical engineering: a systematic review.** [s.n.], 2020. v. 6. 123–128 p. Disponível em: <<https://medcraveonline.com/IJBSBE/intelligent-agents-in-biomedical-engineering-a-systematic-review.html>>. Citado na página 129.

FILGUEIRAS, Tatiana Pereira; BERTEMES-FILHO, Pedro. **Internet of Things na engenharia biomédica.** Atena Editora, 2020. 68–76 p. Disponível em: <<https://atenaeditora.com.br/catalogo/post/internet-of-things-na-engenharia-biomedica>>. Citado na página 130.

FILGUEIRAS, Tatiana Pereira; BERTEMES-FILHO, Pedro; NOVELETTO, Fabrício. **Evaluating the Accuracy of Low-Cost Wearable Sensors for Healthcare Monitoring.** [s.n.], 2025. v. 16. 791 p. Disponível em: <<https://doi.org/10.3390/mi16070791>>. Citado 2 vezes nas páginas 59 e 130.

FILGUEIRAS, Tatiana Pereira; TORRES, Caroline Ruella Paiva; BERTEMES-FILHO, Pedro. **Low Cost System for Fall Detection in the Elderly.** Cincinnati: IEEE, 2020. 697 p. Citado 2 vezes nas páginas 57 e 130.

FRANKLIN, Stan; GRAESSER, Art. **Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents.** [S.l.], 1996. 21–35 p. Citado na página 48.

GARCIA-ESPINOSA, Eduardo et al. **Power Consumption Analysis of Bluetooth Low Energy Commercial Products and Their Implications for IoT Applications.** [s.n.], 2018. v. 7. ISSN 2079-9292. Disponível em: <<https://www.mdpi.com/2079-9292/7/12/386>>. Citado na página 28.

GOMES, M. C. et al. **Fisiopatologia do choque.** 2016. Disponível em: <<https://docs.bvsalud.org/biblioref/2016/09/1892/2403-13547-1-pb.pdf>>. Citado na página 121.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning.** MIT Press, 2016. ISBN 9780262035613. Disponível em: <<https://www.deeplearningbook.org/>>. Citado 4 vezes nas páginas 50, 51, 91 e 94.

GROSOF, Benjamin N et al. **Reusable architecture for embedding rule-based intelligence in information agents.** [S.l.]: IBM Thomas J. Watson Research Division, 1995. Citado na página 48.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.** 2nd. ed. Sebastopol, CA: O'Reilly Media, 2019. ISBN 978-1-492-03264-9. Citado 2 vezes nas páginas 55 e 56.

HAITAM, Ettazi; NAJAT, Rafalia; JAAFAR, Abouchabaka. **Harnessing Machine Learning and Multi Agent Systems for Health Crisis Analysis in North Africa.** [S.l.], 2023. v. 412. 01088 p. Citado 5 vezes nas páginas 24, 124, 125, 126 e 127.

HAMDANI, Samer; SBEYTI, Hassan. **A Comparative study of COAP and MQTT communication protocols.** [S.l.: s.n.], 2019. 1-5 p. Citado 3 vezes nas páginas 19, 31 e 32.

HAYES-ROTH, Barbara. **An architecture for adaptive intelligent systems.** [S.l.]: Elsevier, 1995. v. 72. 329–365 p. Citado na página 48.

HAYKIN, S. **Neural Networks and Learning Machines.** 3rd. ed. [S.l.]: Pearson, 2009. Citado na página 84.

HECHT-NIELSEN, Robert. **Theory of the backpropagation neural network.** [S.l.], 1989. 593–605 p. Citado na página 91.

HO, Tan-Jan; TSAI, Shao-Wei; LIN, Sheng-Ming. **A Power-Efficient IoT-Based Awakening System for Healthcare.** [S.l.: s.n.], 2023. 1-4 p. Citado 2 vezes nas páginas 28 e 42.

HOU, Wenlong et al. **ADAgent: LLM Agent for Alzheimer's Disease Analysis with Collaborative Coordinator.** [s.n.], 2025. Disponível em: <<https://arxiv.org/abs/2506.11150>>. Citado 4 vezes nas páginas 24, 124, 125 e 127.

HUHNS, Michael N; STEPHENS, Larry M. **Multiagent systems and societies of agents.** [S.l.]: MIT press Cambridge^ eMA MA, 1999. v. 1. 79–114 p. Citado na página 19.

HUINFINITO. **Módulos para Arduino e IoT.** 2025. Referência: MDLOLEDAZAM96; Controlador SSD1306; Interface SPI; Tensão 3,3–5 V; Resolução 128×64 px; Dimensões: 28×28×12 mm; Peso: 3,4 g. Disponível em: <<https://www.huinfinite.com.br/telas-lcd-display/1327-modulo-display-oled-128x64-096-azul-e-amarelo.html>>. Citado 5 vezes nas páginas 43, 44, 45, 46 e 47.

HUMAYUN, Mamoon et al. **Agent-Based Medical Health Monitoring System**. [s.n.], 2022. v. 22. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/22/8/2820>>. Citado 5 vezes nas páginas 23, 124, 125, 126 e 127.

IOFFE, Sergey; SZEGEDY, Christian. **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift**. Lille, France: JMLR Workshop and Conference Proceedings, 2015. 448–456 p. Citado 3 vezes nas páginas 50, 52 e 99.

ISLAM, Hasan M. A.; LAGUTIN, Dmitrij; FOTIOU, Nikos. **Transparent CoAP Services to IoT Endpoints through ICN Operator Network**. [S.l.: s.n.], 2019. v. 19. 1339 p. Citado na página 31.

ISLAM, Md. Milon et al. **Internet of Things: Device Capabilities, Architectures, Protocols, and Smart Applications in Healthcare Domain**. [S.l.: s.n.], 2022. v. 22. Figura sobre Arduino, Raspberry Pi e ESP8266. Citado 3 vezes nas páginas 33, 34 e 39.

JENNINGS, Nicholas R et al. **Automated negotiation: prospects, methods and challenges**. [S.l.: s.n.], 2001. v. 10. 199–215 p. Citado na página 48.

JOHANSSON, Lovisa; DOSSOT, David. **RabbitMQ Essentials**. [S.l.]: Packt Publishing, 2020. ISBN 978-1789131666. Citado 3 vezes nas páginas 19, 29 e 30.

KIM, Junho et al. **Photoplethysmography in Wearable Devices: A Review**. [S.l.: s.n.], 2023. v. 12. 2923 p. Citado 3 vezes nas páginas 19, 36 e 37.

KIM, Seon-Chil; CHO, Sung-Hyoun. **Blood Pressure Estimation Algorithm Based on Photoplethysmography Pulse Analyses**. Korea: [s.n.], 2020. v. 10. 1-10 p. Disponível em: <<https://www.mdpi.com/2076-3417/10/12/4068>>. Acesso em: 10 June 2020. Citado 2 vezes nas páginas 36 e 37.

KINGMA, Diederik P.; BA, Jimmy Lei. **Adam: A Method for Stochastic Optimization**. [s.n.], 2015. ArXiv:1412.6980. Disponível em: <<https://arxiv.org/abs/1412.6980>>. Citado 3 vezes nas páginas 50, 51 e 86.

KUMAR, Rakesh. **Integrating Actuators and Sensors in Enhancing Embedded Systems**. [s.n.], 2025. On-line article. Disponível em: <<https://www.oemsecrets.com/articles/integrating-actuators-and-sensors-in-enhancing-embedded-systems>>. Citado 2 vezes nas páginas 33 e 39.

KURNIAWAN, A. **IoT Projects with Arduino Nano 33 BLE Sense**. [s.n.], 2021. Disponível em: <<https://doi.org/10.1007/978-1-4842-6458-4>>. Acesso em: 17 de junho de 2024. Citado 2 vezes nas páginas 41 e 42.

MAES, Pattie. **Artificial life meets entertainment: lifelike autonomous agents**. [S.l.]: ACM New York, NY, USA, 1995. v. 38. 108–114 p. Citado na página 48.

MARCON, Eric et al. **A multi-agent system based on reactive decision rules for solving the caregiver routing problem in home health care**. [s.n.], 2017. v. 74. 134-151 p. ISSN 1569-190X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1569190X17300539>>. Citado 5 vezes nas páginas 25, 124, 125, 126 e 127.

- MARTINELLI, Bruno. **Alterações cardiorrespiratórias de pacientes submetidos à oxigenoterapia hiperbárica**. 2019. Universidade do Sagrado Coração. Disponível em: <<https://www.scribd.com/document/633610538/Alteracoes-cardiorrespiratorias-de-pacientes>>. Citado na página 121.
- MARTÍNEZ, Lucía; PÉREZ, Carlos. **MAX30102–Photometric Biosensor Coupled to ESP32–Webserver for IoT Vital Signs Monitoring**. [s.n.], 2021. v. 16. 9 p. Disponível em: <<https://www.mdpi.com/2673-4591/16/1/9>>. Citado na página 37.
- MEDICINE, London School of Hygiene & Tropical. **LSHTM Data Compass**. 2025. London, U.K. <<https://datacompass.lshtm.ac.uk/>>. Citado na página 75.
- MEHLIG, B. **Machine Learning with Neural Networks: An Introduction for Scientists and Engineers**. [S.l.]: Cambridge University Press, 2021. Citado na página 84.
- MIR, Md. Saym; JAWAD, Asef. **Trends in IoT Hardware and Software Platforms**. [s.n.], 2025. Full-text available on ResearchGate. Disponível em: <https://www.researchgate.net/publication/392708662_Trends_in_IoT_Hardware_and_Software_Platforms>. Citado na página 39.
- MOUHA, Radouan Ait Radouan Ait et al. **Internet of Things (IoT)**. [S.l.: s.n.], 2021. v. 9. 77 p. Citado 2 vezes nas páginas 19 e 27.
- MSD Manuals. **Choque hipovolêmico: fisiopatologia e causas**. 2024. Disponível em: <<https://www.msmanuals.com/pt/profissional/medicina-de-cuidados-cr%C3%ADticos/choque-e-reanima%C3%A7%C3%A3o-vol%C3%A3o-amb%C3%A9rica/choque>>. Citado na página 121.
- MUKHERJI, Samudra et al. **Smart Agriculture using Internet of Things and MQTT Protocol**. [S.l.: s.n.], 2019. 14-16 p. Citado 2 vezes nas páginas 19 e 31.
- MUTLAG, Ammar Awad et al. **Multi-Agent Systems in Fog–Cloud Computing for Critical Healthcare Task Management Model (CHTM) Used for ECG Monitoring**. MDPI, 2021. v. 21. 6923 p. Disponível em: <<https://www.mdpi.com/1424-8220/21/20/6923>>. Citado 5 vezes nas páginas 25, 124, 125, 126 e 127.
- NAIR, Karan et al. **Optimizing power consumption in iot based wireless sensor networks using Bluetooth Low Energy**. [S.l.: s.n.], 2015. 589-593 p. Citado na página 29.
- NAIR, Vinod; HINTON, Geoffrey E. **Rectified Linear Units Improve Restricted Boltzmann Machines**. Haifa, Israel, 2010. 807–814 p. Citado 2 vezes nas páginas 50 e 51.
- NHANES. **National Health and Nutrition Examination Survey (NHANES)**. 2025. Centers for Disease Control and Prevention, Atlanta, GA, USA. <<https://www.cdc.gov/nchs/nhanes/index.html>>. Citado na página 75.
- NOERGAARD, Tammy. **Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers Working with Embedded Systems**. 2. ed. Oxford, UK: Newnes, 2018. Citado 2 vezes nas páginas 39 e 40.
- OMARIBA, Z. B.; ZHANG, L.; SUN, D. **Review on Health Management System for Lithium-Ion Batteries of Electric Vehicles**. [s.n.], 2018. v. 7. 72-72 p. Disponível em: <<https://doi.org/10.3390/electronics7050072>>. Acesso em: 17 de junho de 2024. Citado na página 45.

- PARK, Junyung et al. **Photoplethysmogram Analysis and Applications: An Integrative Review**. [S.l.: s.n.], 2021. v. 12. 808451 p. Citado 3 vezes nas páginas 19, 35 e 36.
- PATEL, Bhakti K.; ALBERT, Richard K. **Insuficiência respiratória hipoxêmica aguda (IRHA, SARA)**. 2024. Disponível em: <<https://www.msdmanuals.com/pt/profissional/medicina-de-cuidados-cr%ADticos/insufici%C3%A3ncia-respirat%C3%A3ria-e-ventila%C3%A7%C3%A3o-mec%C3%A2nica/insufici%C3%A3ncia-respirat%C3%A3ria-hipox%C3%A3AAmica-aguda-irha-sara>>. Citado na página 121.
- PERERA, Charith et al. **Sensing as a Service Model for Smart Cities Supported by Internet of Things**. [S.l.: s.n.], 2014. v. 25. 81–93 p. Citado 3 vezes nas páginas 19, 26 e 27.
- PHYSIONET. **The Research Resource for Complex Physiologic Signals**. 2025. <<http://physionet.org/physiobank/>>. Citado na página 75.
- PRABOWO, Norbertus Krisnu; IRWANTO, Irwanto. **The Implementation of Arduino Microcontroller Boards in Science: A Bibliometric Analysis from 2008 to 2022**. Rajarambapu Institute of Technology, 2023. v. 37. 106–123 p. ISSN 2394-1707. Disponível em: <<http://dx.doi.org/10.16920/jeet/2023/v37i2/23154>>. Citado 2 vezes nas páginas 40 e 41.
- PRATAMA, D. W.; ARIYANTO, E.; ABDUROHMAN, M. **Digital clock synchronization using IP with wireless fidelity**. [s.n.], 2017. Disponível em: <<https://doi.org/10.1109/icoict.2017.8074706>>. Acesso em: 17 de junho de 2024. Citado 2 vezes nas páginas 44 e 45.
- ProtoCentral. **Pulse Express – Pulse-Ox & Heart-Rate Sensor with MAX32664**. 2024. <<https://protocentral.com/product/pulse-express-pulse-ox-heart-rate-sensor-with-max32664/>>. Citado na página 46.
- RAZA, M. S. **Advanced Design of Smart Digital Application Using PIC 16F887A Microcontroller and DS 1307 RTC**. [s.n.], 2018. v. 6. 30-30 p. Disponível em: <<https://doi.org/10.11648/j.ajesa.20180601.15>>. Acesso em: 17 de junho de 2024. Citado na página 44.
- REDDY, V. Sreelatha; HARIVARDHAGINI, S.; SREELAKSHMI, G. **IoT and Cloud Based Sustainable Smart Irrigation System**. [s.n.], 2023. v. 410. 01026 p. Mar 2023. Disponível em: <<https://doi.org/10.1051/e3sconf/202341001026>>. Citado na página 34.
- RILEY, Richard D. et al. **Prognosis Research in Healthcare: Concepts, Methods, and Impact**. Oxford University Press, 2019. ISBN 9780198796619. Disponível em: <<https://doi.org/10.1093/med/9780198796619.001.0001>>. Citado na página 98.
- RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. **Learning representations by back-propagating errors**. Cambridge, MA: MIT Press, 1986. 318–362 p. Citado 2 vezes nas páginas 50 e 51.
- RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach, Global Edition 4th**. [S.l.: s.n.], 2021. v. 19. 23 p. Citado na página 19.
- SAJJA, Priti Srinivas. **Multi-agent system for knowledge-based access to distributed databases**. [S.l.: s.n.], 2008. v. 3. Citado na página 49.
- SARKAR, Chandra; DAS, Amitava; JAIN, Ravi Kant. **Development of CoAP protocol for communication in mobile robotic systems using IoT technique**. [S.l.: s.n.], 2025. v. 15. 9269 p. Citado na página 30.

- SATYANARAYANAN, Mahadev. **The Emergence of Edge Computing**. [S.l.: s.n.], 2017. v. 50. 30-39 p. Citado na página 26.
- SCROSATI, B.; GARCHE, J. **Lithium batteries: Status, prospects and future**. [s.n.], 2010. v. 195. 2419-2430 p. Disponível em: <<https://doi.org/10.1016/j.jpowsour.2009.11.048>>. Acesso em: 17 de junho de 2024. Citado 2 vezes nas páginas 45 e 46.
- SHAIK, Sk Masiullah; KUMAR, A V Krishna; KAUR, Parneet. **Adaptive Multi-Agent Deep Reinforcement Learning for Timely Healthcare Interventions**. [s.n.], 2023. Disponível em: <<https://arxiv.org/abs/2309.10980>>. Citado 5 vezes nas páginas 23, 124, 125, 126 e 127.
- SILVA, F. M. et al. **Choque – princípios gerais de diagnóstico precoce e manejo inicial**. 2018. Disponível em: <<https://docs.bvsalud.org/biblioref/2018/04/882566/choque-principios-gerais-de-diagnostico-precoce-e-manejo-inicial.pdf>>. Citado na página 121.
- SRIVASTAVA, Nitish et al. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**. [S.l.: s.n.], 2014. v. 15. 1929–1958 p. Citado 3 vezes nas páginas 50, 53 e 99.
- STEYERBERG, Ewout W. **Clinical Prediction Models**: A practical approach to development, validation, and updating. 2. ed. Springer Cham, 2019. XXXIII, 558 p. (Statistics for Biology and Health). ISBN 978-3-030-16398-3. Disponível em: <<https://doi.org/10.1007/978-3-030-16399-0>>. Citado na página 98.
- SUDIANTO, Agus et al. **Smart Temperature Measurement System for Milling Process Application Based on MLX90614 Infrared Thermometer Sensor with Arduino**. [S.l.: s.n.], 2020. v. 72. 10–24 p. Citado na página 38.
- TARIQ, Muhammad Ashar et al. **Enhancements and Challenges in CoAP—A Survey**. [s.n.], 2020. v. 20. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/20/21/6391>>. Citado na página 30.
- TSEBESEBE, Nkgaphe T. et al. **Arduino-based devices in healthcare and environmental monitoring: a review**. [s.n.], 2025. Disponível em: <<https://link.springer.com/article/10.1007/s43926-025-00139-z>>. Citado na página 42.
- TSUJIMURA, T. **OLED Display Module**. [s.n.], 2017. 99-141 p. Disponível em: <<https://doi.org/10.1002/9781119187493.ch4>>. Acesso em: 17 de junho de 2024. Citado na página 43.
- VALSAN, V.; EMMANUEL, A. O. **Design and Simulation of a Single-Output Multichannel Charger for Lithium-Ion Batteries**. [s.n.], 2017. 525-536 p. Disponível em: <https://doi.org/10.1007/978-981-10-4762-6_50>. Acesso em: 17 de junho de 2024. Citado na página 45.
- VOHRA, Jasmine; CHAVAN, Ayush; UPADHYAY, Ishan. **Real-Time Health Monitoring with Arduino Uno and MAX30102**. [s.n.], 2024. v. 19. 1-13 p. Disponível em: <<https://tanzj.net/wp-content/uploads/2024/04/TJ-1548.pdf>>. Citado na página 37.
- WAN, J. et al. **Wearable IoT enabled real-time health monitoring system**. [s.n.], 2018. v. 2018. Disponível em: <<https://doi.org/10.1186/s13638-018-1308-x>>. Acesso em: 17 de junho de 2024. Citado na página 42.

- WEISER, Mark; BROWN, John Seely. **Das kommende Zeitalter der Calm Technology.** Bielefeld: Transcript, 2015. 59–62 p. Citado na página 26.
- WYTRĘBOWICZ, Jacek; CABAJ, Krzysztof; KRAWIEC, Jerzy. **Messaging Protocols for IoT Systems—A Pragmatic Comparison.** [s.n.], 2021. v. 21. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/21/20/6904>>. Citado 3 vezes nas páginas 29, 30 e 31.
- YASAR, Sami; GILLIS, Alexander. **Internet of Things (IoT).** 2024. <<https://www.techtarget.com/whatis/definition/Internet-of-Things-IoT>>. Acesso em: 27 maio 2025. Citado 3 vezes nas páginas 27, 33 e 34.
- ZORMATI, Mohamed Ali; LAKHLEF, Hicham; OUNI, Sofiane. **Review and Analysis of Recent Advances in Intelligent Network Softwarization for the Internet of Things.** [S.l.: s.n.], 2024. Citado na página 40.
- ZUBI, G. et al. **The lithium-ion battery: State of the art and future perspectives.** [s.n.], 2018. v. 89. 292-308 p. Disponível em: <<https://doi.org/10.1016/j.rser.2018.03.002>>. Acesso em: 17 de junho de 2024. Citado na página 45.

APÊNDICES

APÊNDICE A – PSEUDO-CÓDIGOS DE TREINAMENTO DAS RNAS

```

1 INICIAR
2
3 IMPORTAR bibliotecas necessarias (NumPy, TensorFlow/Keras, Scikit
-learn, Pandas, Pickle)
4
5 CARREGAR dataset de entrada a partir de "blood_pressure_processed
.csv"
6
7 SELECCIONAR as colunas de entrada (features):
8     - Pressao sistolica media (avg_systolic)
9     - Pressao diastolica media (avg_diastolic)
10    - Frequencia cardiaca media (avg_pulse)
11
12 SELECCIONAR a coluna de saida (classe da pressao arterial)
13
14 INICIALIZAR um codificador de rotulos (LabelEncoder)
15 TRANSFORMAR as classes categoricas em valores numericos (
y_encoded)
16
17 INICIALIZAR um normalizador (StandardScaler)
18 NORMALIZAR os dados de entrada (X_scaled)
19
20 DIVIDIR os dados normalizados em:
21     - Conjunto de treino (80%)
22     - Conjunto de teste (20%)
23
24 CRIAR um modelo de rede neural sequencial com as seguintes
camadas:
25     - Camada densa com 16 neuronios e ativacao ReLU
26     - Camada densa com 8 neuronios e ativacao ReLU
27     - Camada de saida com numero de neuronios igual ao numero de
classes e ativacao Softmax
28
29 COMPILAR o modelo com:
30     - Optimizador Adam
31     - Funcao de perda: Entropia cruzada categorica esparsa
32     - Metrica: Acuracia
33
34 TREINAR o modelo com os dados de treino por 50 epochas, em lotes

```

```

    de 8 amostras, validando com o conjunto de teste
35
36 SALVAR o modelo treinado em "blood_pressure_model.h5"
37
38 SALVAR o normalizador (scaler) em "scaler.pkl"
39 SALVAR o codificador de rotulos em "label_encoder.pkl"
40
41 AVALIAR o modelo com o conjunto de teste
42 IMPRIMIR a acuracia obtida no teste
43
44 FINALIZAR

```

Figura A.1 – Pseudocodigo do processo de treinamento de um modelo de classificacao para sinais de pressao arterial. Todos os demais sinais seguem o mesmo padrao.

```

1 INICIAR
2
3 IMPORTAR bibliotecas:
4     - Pandas, NumPy
5     - TensorFlow Keras (Sequential, Dense, Dropout,
6         BatchNormalization)
7     - Scikit-learn (StandardScaler, compute_class_weight)
8     - Pickle para serializacao
9
10 CARREGAR os dados do arquivo CSV "dataset.csv"
11
12 DEFINIR colunas de entrada (features):
13     - SpO2, HeartRate, Temperatura, PAS, PAD
14
15 DEFINIR nome da coluna de diagnostico (Diagnostico)
16
17 VERIFICAR se todas as colunas de entrada estao presentes no
18     dataset
19     SE FALTAR colunas, GERAR ERRO e ENCERRAR
20
21 EXTRAIR valores dos dados de entrada (data_values)
22
23 INICIALIZAR um normalizador (StandardScaler)
24 NORMALIZAR os dados (data_scaled)
25 MAPEAR os valores de diagnostico para indices numericos
26 ADICIONAR coluna "diagnosis_numeric" ao dataset com os valores

```

```

    numericos

26
27 CALCULAR pesos das classes para balanceamento (class_weight_dict)
28 DEFINIR numero total de classes (num_classes)

29
30 CRIAR modelo MLP sequencial com as seguintes camadas:
    - Camada densa com 128 neuronios, ativacao ReLU
31    - Normalizacao em lote (BatchNormalization)
32    - Dropout de 35%
33    - Camada densa com 64 neuronios, ativacao ReLU
34    - BatchNormalization
35    - Dropout de 35%
36    - Camada densa com 32 neuronios, ativacao ReLU
37    - Dropout de 35%
38    - Camada de saida com num_classes e ativacao Softmax

40
41 COMPILAR o modelo com:
42    - Otimizador Adam
43    - Perda: Entropia cruzada categorica esparsa
44    - Metrica: Acuracia

45
46 TREINAR o modelo com:
47    - Dados normalizados (data_scaled)
48    - Rotulos numericos (diagnosis_numeric)
49    - 60 epochas
50    - Tamanho de lote: 32
51    - Divisao de validacao: 20%
52    - Pesos de classe: class_weight_dict

53
54 SALVAR o modelo treinado como "prognosis_model.h5"
55
56 SALVAR o normalizador como "scaler_prognosis.pkl"
57
58 SALVAR o dicionario de diagnosticos como "label_encoder_prognosis
    .pkl"
59
60 EXIBIR mensagem "Treinamento concluido e modelo salvo!"
61
62 FINALIZAR

```

Figura A.2 – Pseudocodigo para treinamento de modelo de prognostico com diagnosticos multiclassse

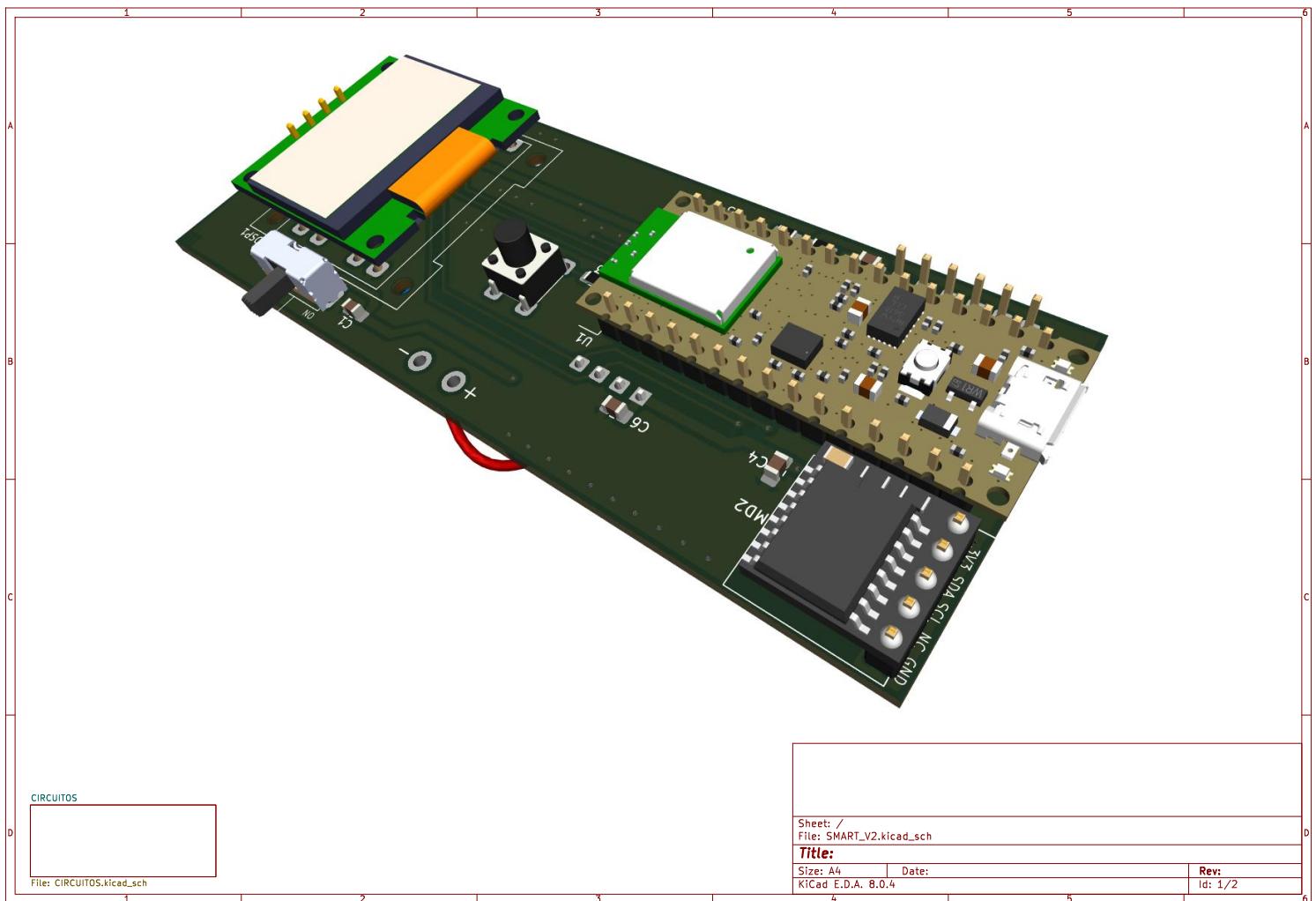
APÊNDICE B – TERMOS TÉCNICOS DOS TREINAMENTOS DO APÊNDICE A

Tabela 14 – Explicação dos principais termos técnicos usados no treinamento do modelo de prognóstico

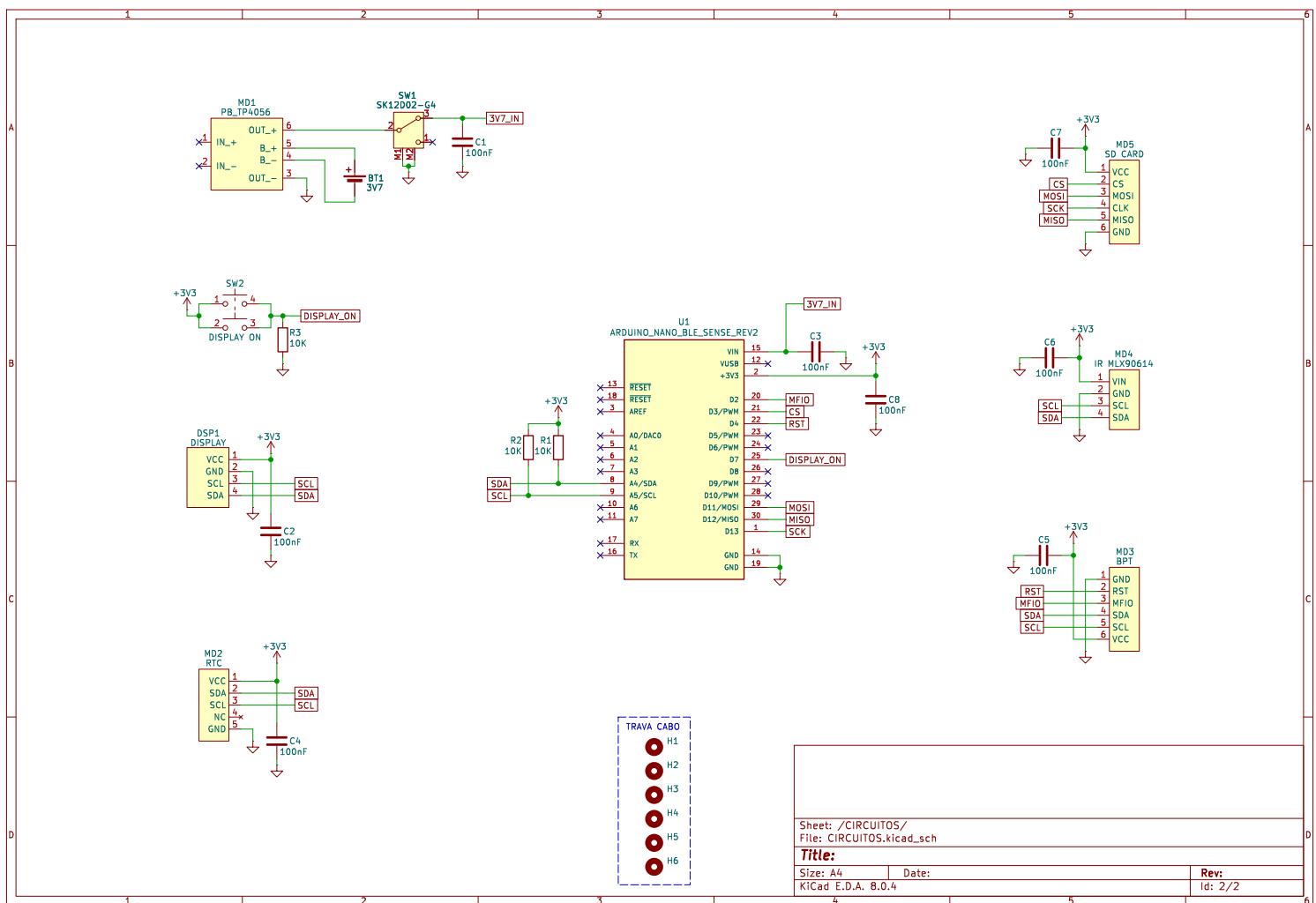
| Termo | Descrição |
|---------------------------------|--|
| Dropout | Técnica de regularização que reduz overfitting desativando aleatoriamente uma porcentagem de neurônios durante o treinamento. No pseudocódigo, um Dropout de 35% é aplicado após cada camada oculta. |
| BatchNormalization | Camada que normaliza as saídas intermediárias da rede neural para melhorar a estabilidade e acelerar o treinamento. Evita variações internas na distribuição dos dados entre camadas. |
| class_weight | Mecanismo de balanceamento que atribui pesos diferentes a cada classe, compensando desigualdade na distribuição de dados entre as categorias. Classes raras recebem maior peso. |
| StandardScaler | Técnica de normalização que transforma os atributos para que tenham média zero e desvio padrão um, permitindo que a rede trate todas as variáveis em escala comparável. |
| sparse_categorical_crossentropy | Função de perda utilizada para classificação multiclasse com rótulos inteiros (não one-hot). Mede a divergência entre a previsão (softmax) e a classe verdadeira. |

Fonte: Elaborado pela autora (2025).

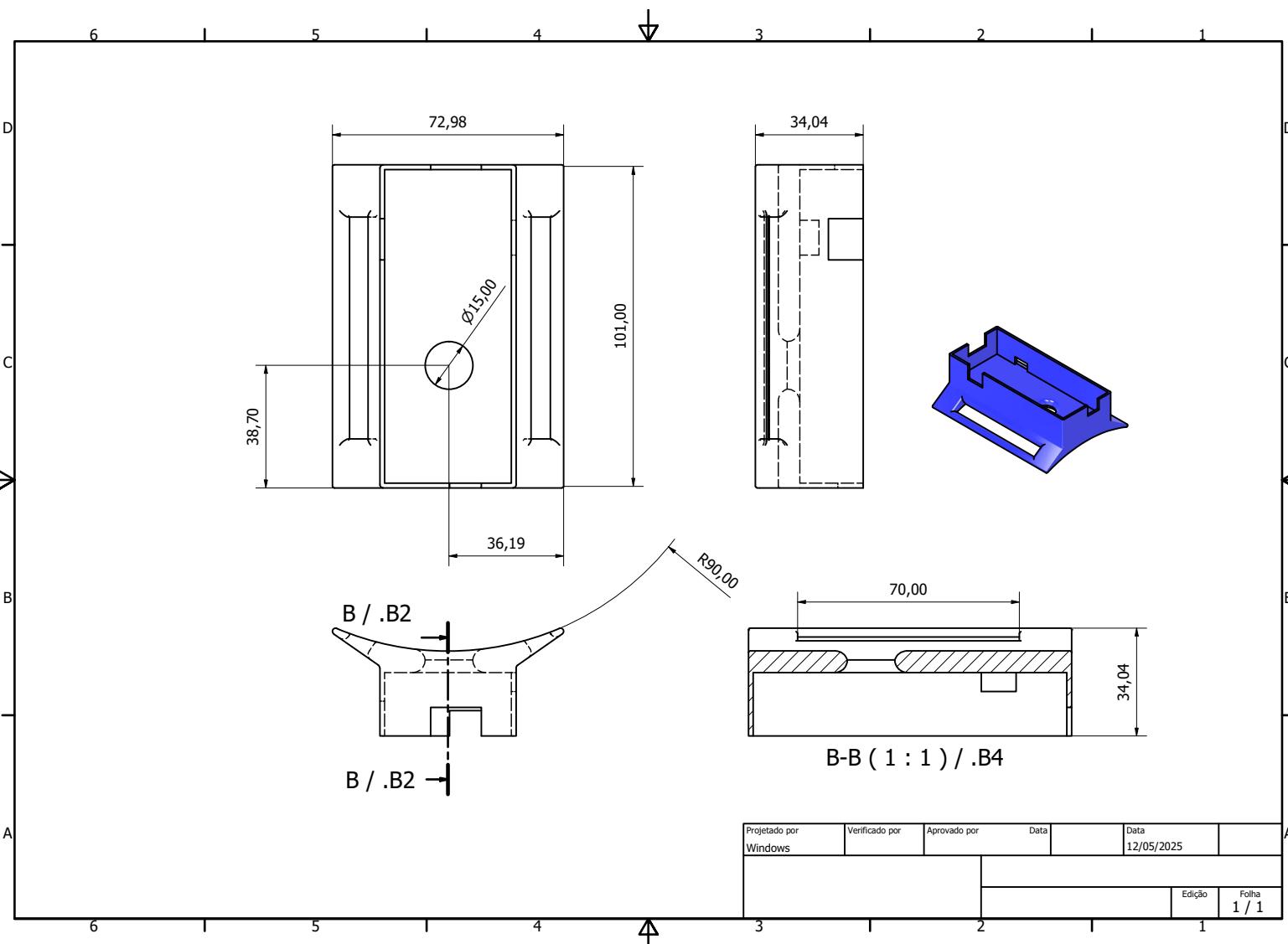
**APÊNDICE C – DIAGRAMAS DA PLACA DO DISPOSITIVO UTILIZADO NESSA
TESE**

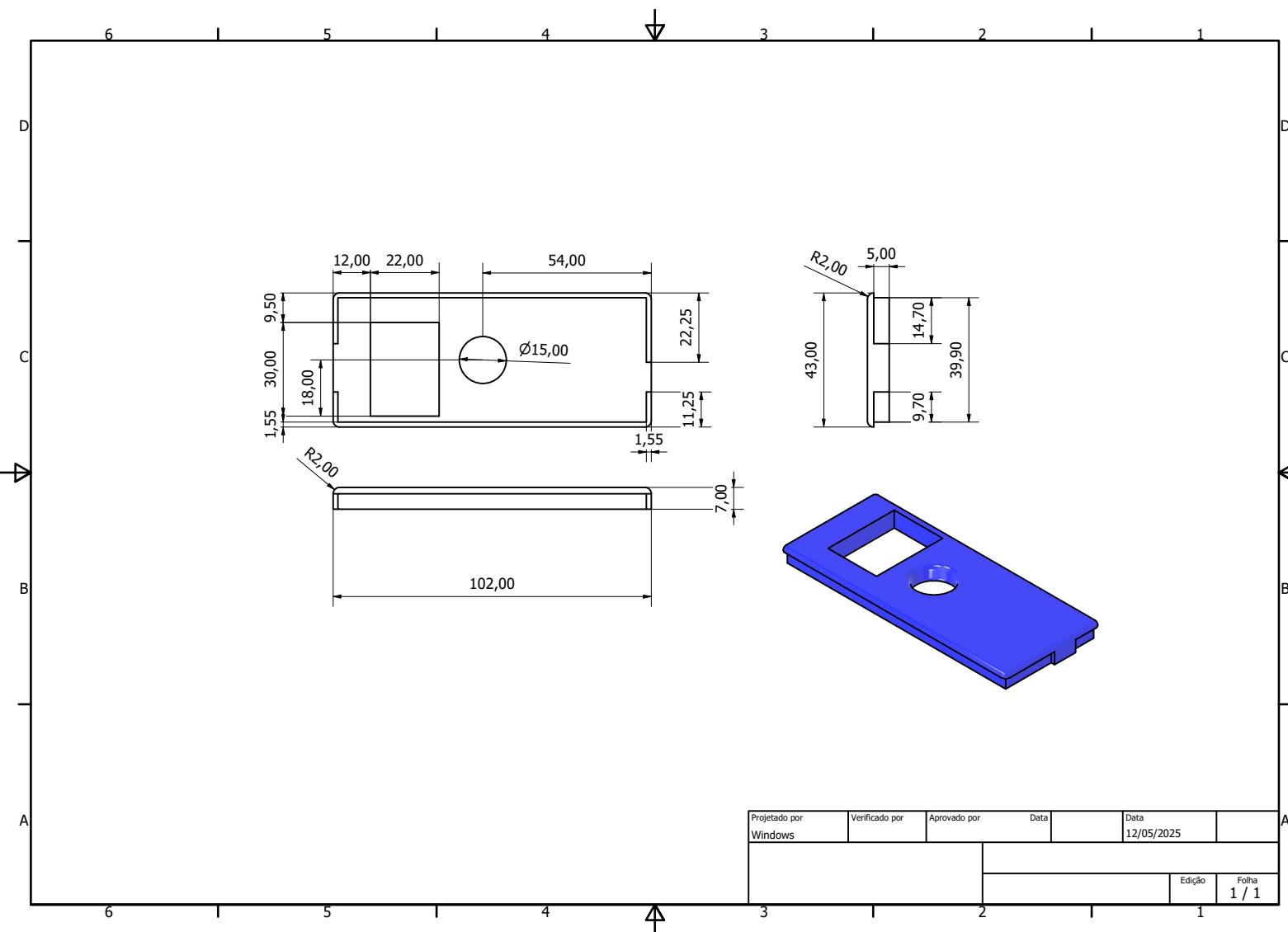


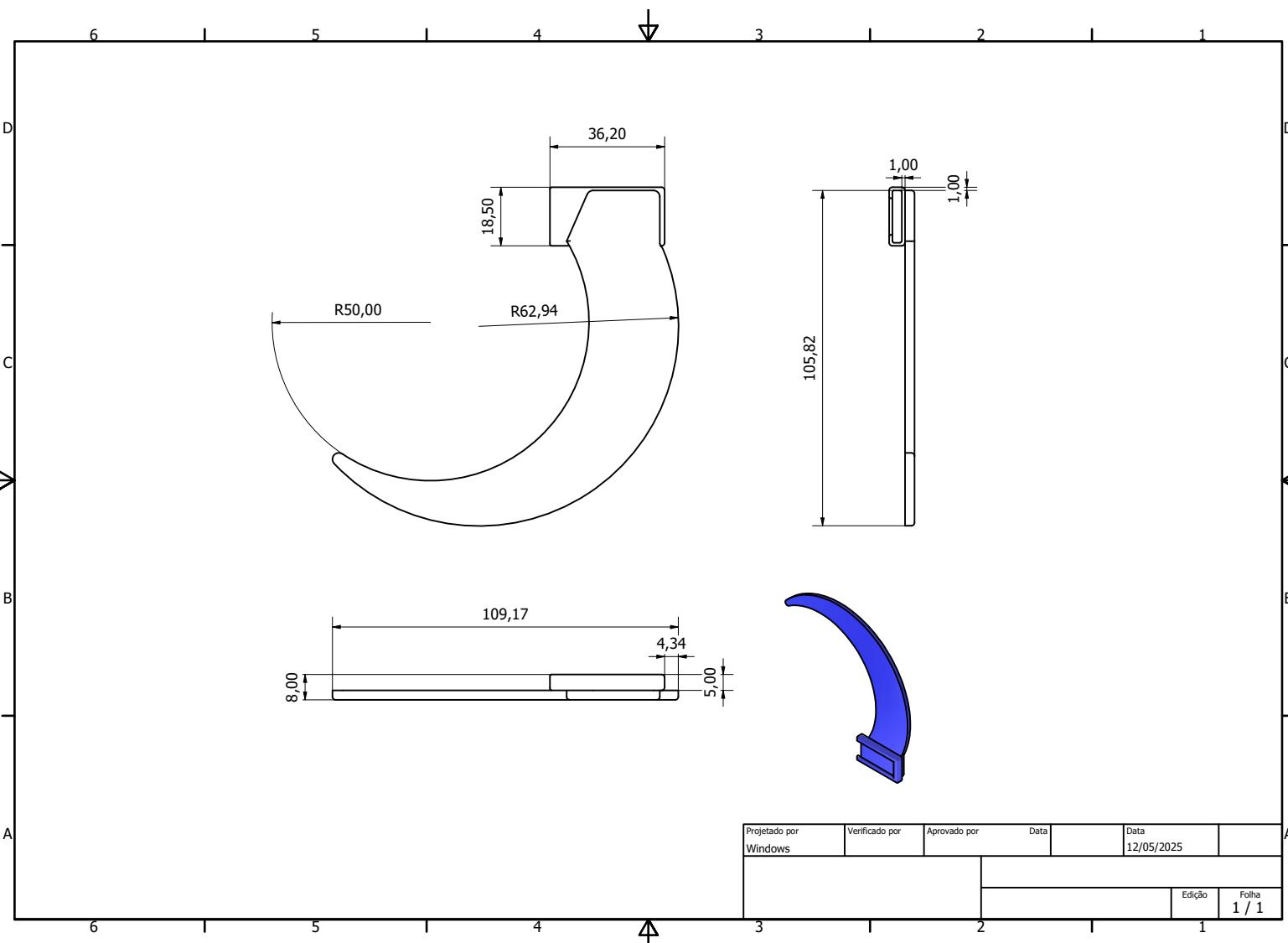
Sheet: /
File: SMART_V2.kicad_sch
Title:
Size: A4 Date:
KiCad E.D.A. 8.0.4 Rev:
Id: 1/2



**APÊNDICE D – DIAGRAMAS DA CASE DO DISPOSITIVO UTILIZADO NESSA
TESE**







APÊNDICE E – QUESTIONÁRIO DE TRIAGEM

Formulário de Triagem - seleção de voluntários: **SMACare**

Questionário adaptado do Ministério da Saúde (VIGITEL) para a triagem e seleção de voluntários considerados elegíveis à pesquisa, baseado nos critérios de inclusão e exclusão do estudo.

Por favor, responda às perguntas com atenção e sinceridade de acordo com o que é pedido. O questionário é parte importante e imprescindível da pesquisa.

O preenchimento do questionário de forma adequada é uma etapa que garante segurança à sua participação no estudo.

DADOS DE IDENTIFICAÇÃO E CADASTRAMENTO.

Dentro dos aspectos éticos, garantimos sigilo aos dados pessoais aqui dispostos. Serão utilizados para cadastramento do voluntário ao sistema de armazenamento de dados da pesquisa

Qual é o seu nome completo (sem abreviações)?

Sua resposta

Qual a sua data de nascimento?

Sua resposta

Por favor, insira o seu CPF para cadastro.

Sua resposta

Digite seu telefone celular.

Insira, preferencialmente, um número de celular com WhatsApp, pois podemos entrar em contato com você por lá! (o número pode ser do responsável legal/cuidador do candidato)

Sua resposta

Qual é seu e-mail?

Por favor, insira um endereço de e-mail válido e com acesso frequente (o endereço de e-mail pode ser do responsável legal/cuidador do candidato).

Sua resposta

QUESTIONÁRIO SOCIODEMOGRÁFICO PARA IDENTIFICAÇÃO DE FATORES DE RISCO.

As perguntas a seguir servem para nortear a avaliação dos critérios de inclusão e exclusão da pesquisa, baseadas em fatores de risco relacionados à hábitos e estilo de vida, e também história clínica.

Qual é o seu peso (Kg)?

Por favor, insira o valor mais próximo do real. Digite no formato xxx,x

Sua resposta

Qual é a sua altura (metros)?

Por favor, insira o valor mais próximo do real. Digite no formato x,xx

Sua resposta

Como você classificaria seu estado geral de saúde?

- Muito bom
- Bom
- Regular
- Ruim
- Muito ruim

Já recebeu diagnóstico de alguma doença crônica em algum momento da vida?

Como é o caso de hipertensão arterial, tuberculose, hanseníase, bronquite, asma, insuficiência renal, sopro cardíaco, entre outros. Não considerar doenças pontuais e agudas como viroses gastrointestinais, resfriados, gripe, intoxicação alimentar...

- Sim
- Não

Se você respondeu "**SIM**" para a pergunta anterior, descreva abaixo qual(is) a(s) doença(s) de diagnóstico.

Sua resposta

Faz uso de alguma medicação contínua e controlada?

- Sim
- Não

Se a resposta da pergunta anterior foi "**SIM**", liste o(s) medicamento(s) abaixo.

Sua resposta

O(a) Sr.(a) tem alguma deficiência ou limitação física?

Considere limitações de mobilidade/mobilidade reduzida, deficiência visual e/ou auditiva. Isso é importante para fazermos adaptações de protocolos acessíveis no caso de elegibilidade do voluntário.

- Sim
 Não

Se a resposta à pergunta anterior foi "SIM", liste abaixo qual(is) limitação(ões) apresentada(s).

Sua resposta

O(a) Sr.(a) faz uso de algum acessório de auxílio para a mobilidade?

Considere: bengala, muleta, andador, cadeira de rodas, etc.

- Sim
 Não

Se a resposta à pergunta anterior foi "**SIM**", liste abaixo qual(is) acessório(s) são utilizados.

Sua resposta

O(a) Sr.(a) possui cuidador em tempo integral ou parcial?

- Integral
- Parcial

O cuidador responsável é um profissional contratado independente ou é familiar ao sr.(a)?

- Profissional cuidador contratado
- Familiar

A sua residência/lar de permanência possui escadas?

- Sim
- Não

A sua residência/lar de permanência possui rampas de acesso (externas e/ou internas)?

- Sim
- Não

Em relação a mobilidade e acesso, liste as maiores dificuldades no seu dia a dia

Considere os desafios encontrados durante a mobilidade e a execução de tarefas diárias (subir e descer degraus, tomar banho, locomoção dentro da casa, etc)

Sua resposta

O(a) Sr.(a) já sofreu algum trauma ou fratura em decorrência de quedas?

- Sim
- Não

[Voltar](#)

[Enviar](#)

Página 3 de 3 [Limpar formulário](#)

APÊNDICE F – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO



GABINETE DO REITOR

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

O(a) senhor(a) está sendo convidado a participar de uma pesquisa intitulada “SMACare - Sistema Multiagente para Monitoramento”, que fará a coleta de dados relacionados a sinais vitais (Pressão Arterial, SpO2, Temperatura Corporal e Frequência Cardíaca). A coleta será realizada de forma não-invasiva, através de um dispositivo vestível, no formato de uma braçadeira com conexão ao lóbulo da orelha, que deverá ser usado pelo(a) senhor(a) por no mínimo 4h por dia, durante 7 dias.

O(a) Senhor(a) e seu/sua acompanhante não terão despesas e nem serão remunerados pela participação na pesquisa. Todos os custos relacionados a coleta de dados serão cobertos pela pesquisa, sendo o(a) senhor(a) isento de qualquer cobrança. Em caso de danos, relacionados e decorrentes dos procedimentos realizados durante a pesquisa, será garantida a indenização.

Risco é a possibilidade de dano, prejuízo, lesão ou outra consequência. Dado que a coleta de dados será feita de forma vestível e não-invasiva, o risco de qualquer dano físico é minimizada, porém, a equipe pesquisadora estará à disposição para a solução de qualquer desconforto.

No caso de não adaptação ao dispositivo em sua forma vestível (desconforto físico, peso, irritabilidade, etc), o participante e/ou o cuidador poderão entrar em contato com a equipe por meio do canal disponível e mais acessível para a interrupção da coleta de dados.

O atendimento da equipe poderá ser para: a) identificar e solucionar o problema em relação ao uso do dispositivo, corrigir e retomar a coleta de dados ou b) identificar, solucionar o problema em relação ao uso e interromper de forma imediata a coleta de dados; a escolha em relação a tomada de decisão fica toda a critério do voluntário.

O risco de queda é inerente ao público alvo do estudo. A equipe pesquisadora se responsabiliza por fazer orientações para a minimização do risco de queda dos idosos, a fim de evitar traumas, lesões, fraturas, etc. Em caso real de queda e necessidade de atendimento, os pesquisadores não podem se responsabilizar pelo atendimento de primeiros socorros, entretanto a equipe estará de pronto ofício para auxílio no acionamento da equipe de pronto atendimento, médico, socorristas e para agilizar o atendimento e encaminhamento do idoso ao serviço de saúde mais próximo.

Além do aspecto de riscos físicos e/ou relacionados à saúde e segurança dos voluntários, riscos e constrangimentos psicológicos podem surgir durante o preenchimento do questionário de triagem (desconforto emocional e psicológico, receio ou vergonha de fornecer informações pessoais e clínicas). Para evitar tais situações, nenhuma pergunta do questionário é obrigatória, o voluntário pode seguir o preenchimento e envio do questionário optando por pular perguntas que lhe causem desconforto ou estranheza.

Caso alguma pergunta pulada pelo candidato seja essencial para a decisão de elegibilidade, o grupo pesquisador entrará em contato com o candidato ou com o cuidador para entender a melhor forma de ter a resposta necessária, ficando por conta do voluntário escolher a melhor opção e que lhe traga maior conforto (reunião remota ou presencial, um novo questionário, ou outra opção sugerida pelo candidato ou escolhida em conjunto).

O questionário de triagem será atendido pelo candidato de forma remota e online, podendo-o responder no horário e local que julgue melhor e mais adequado para atividade, amenizando qualquer desconforto físico relacionado ao atendimento do questionário.

A sua identidade será preservada pois cada indivíduo será identificado por um identificador e uma numeração (Ex.: VO01)

Os benefícios da pesquisa incluem ganho de conhecimento, percepção, entendimento e compreensão da dinâmica em relação à mobilidade do(a) voluntário(a).

O uso do dispositivo melhora o envolvimento do(a) voluntário(a) e do cuidador em relação ao cotidiano de mobilidade no caso de o participante ser idoso, facilitando a percepção de movimento no ambiente, identificando fatores de risco e pontos a serem melhorados. Além, é claro, da aproximação pessoal, melhorando convívio e a comunicação entre a dupla.

Os dados coletados irão beneficiar o participante e o cuidador provendo o histórico de sinais vitais e informações trabalhadas sobre acelerometria. As informações relacionadas ao giroscópio e acelerometria podem



GABINETE DO REITOR

auxiliar na identificação de momentos do dia, ou em quais atividades, há maior oscilação de equilíbrio, e por consequência, risco de queda.

A participação no estudo engaja a sociedade com a comunidade científica, aproximando o público do processo de criação de um projeto, da testagem de dispositivos e, por consequência, aumenta o conhecimento da população sobre assuntos tão peculiares do mundo científico.

As pessoas que estarão acompanhando os procedimentos serão os pesquisadores Tatiana Pereira Filgueiras e a responsável pelo local Elisângela da Silva.

O(a) senhor(a) poderá se retirar do estudo a qualquer momento, sem qualquer tipo de constrangimento.

Solicitamos a sua autorização para o uso de seus dados para a produção de artigos técnicos e científicos. A sua privacidade será mantida através da não-identificação do seu nome.

Este termo de consentimento livre e esclarecido é feito em duas vias, sendo que uma delas ficará em poder do pesquisador e outra com o sujeito participante da pesquisa.

NOME DO PESQUISADOR RESPONSÁVEL PARA CONTATO: TATIANA PEREIRA FILGUEIRAS

NÚMERO DO TELEFONE: (47) 9 9702-4525

ENDEREÇO: .RUA CARLOS BOOS, 60, APTO 51, GRAVATÁ, NAVEGANTES, SC

ASSINATURA DO(A) PESQUISADOR(A):

Comitê de Ética em Pesquisa Envolvendo Seres Humanos – CEPSH/UDESC

Av. Madre Benvenuta, 2007 – Itacorubi – Florianópolis – SC -88035-901

Fone/Fax: (48) 3664-8084 / (48) 3664-7881 - E-mail: cepsh.reitoria@udesc.br / cepsh.udesc@gmail.com

CONEP- Comissão Nacional de Ética em Pesquisa

SRTV 701, Via W 5 Norte – lote D - Edifício PO 700, 3º andar – Asa Norte - Brasília-DF - 70719-040

Fone: (61) 3315-5878/ 5879 – E-mail: coneep@saude.gov.br

TERMO DE CONSENTIMENTO

Declaro que fui informado sobre todos os procedimentos da pesquisa e, que recebi de forma clara e objetiva todas as explicações pertinentes ao projeto e, que todos os dados a meu respeito serão sigilosos. Eu comprehendo que neste estudo, as medições dos experimentos/procedimentos de tratamento serão feitas em mim, e que fui informado que posso me retirar do estudo a qualquer momento.

Nome por extenso _____

Assinatura _____ Local: _____ Data: / / .

APÊNDICE G – ATESTADO DE UM DOS PARTICIPANTES DA PESQUISA



Declaração de Óbito

38437440-9

Identificação

1 Tipo de óbito
1 Fetal
2 Não Fetal

2 Data do óbito
23.09.2024 16:35

3 Hora

4 Cartão SUS

4 Naturalidade

CACHOEIRA/BA
Município / UF (se estrangeiro informar País)

Residência

8 Data de nascimento

9 Idade
Anos completos
74

Menores de 1 ano
Meses

Dias

Horas

Minutos

Ignorado

9

10 Sexo
M - Masc.
F - Fem.
I - Ignorado

11 Raça/Cor
1 Branca
2 Preta
3 Amarela

4 Separado judicialmente/
divorciado

5 União estável

6 Ignorada

7 Solteiro

15 Logradouro (rua, praça, avenida, etc)
Nível
0 Sem escolaridade
1 Fundamental I (1ª a 4ª Série)
2 Fundamental II (5ª a 8ª Série)

3 Médio (antigo 2º grau)
4 Superior incompleto
5 Superior completo

Ignorado
9

Série

14 Ocupação habitual
(Informar anterior, se aposentado / desempregado)

Disponível

Código CBO 2002

Ocorrência

17 Bairro/Distrito
Código
18 Município de residência
Código

19 UF
SC

20 Local de ocorrência do óbito
1 Hospital
2 Outros estab. saúde

3 Domicílio
4 Via pública
5 Outros

Ignorado
9

21 Estabelecimento

HOSPITAL UNIMED LITORAL

Código CNES

Fetal ou menor que 1 ano

22 Endereço de ocorrência (rua, praça, avenida, etc)

AVENIDA DO ESTADO

Condições do óbito

24 Bairro/Distrito
Código

25 Município de ocorrência
Código

26 UF
SC

PREENCHIMENTO EXCLUSIVO PARA ÓBITOS FETAIS E DE MENORES DE 1 ANO - INFORMAÇÕES SOBRE A MÃE

27 Idade (anos)
28 Nível
0 Sem escolaridade
1 Fundamental I (1ª a 4ª Série)
2 Fundamental II (5ª a 8ª Série)

29 Escolaridade (última série concluída)
3 Médio (antigo 2º grau)
4 Superior incompleto
5 Superior completo

Ignorado
9

Série

30 Número de filhos tidos
Nascidos
1 Perdas fetais/abortos
99 Ignorado

31 Nº de semanas de gestação

32 Tipo de gravidez
1 Única
2 Dupla
3 Tripla e mais
9 Ignorada

33 Tipo de parto
1 Vaginal
2 Cesáreo
9 Ignorado

34 Morte em relação ao parto
1 Antes
2 Durante
3 Depois
9 Ignorado

35 Peso ao nascer
Gramas

36 Número da Declaração de Nascido Vivo

ÓBITO DE MULHER EM IDADE FÉRIL

37 A morte ocorreu

1 Na gravidez
2 No parto

3 No abortamento
4 Até 42 dias após o término da gestação

5 De 43 dias a 1 ano após o término da gestação
8 Não ocorreu nestes períodos

ASSISTÊNCIA MÉDICA

38 Recebeu assist. médica durante a doença que ocasionou a morte?

39 Necrópsia?

1 Sim
2 Não
9 Ignorado

Tempo aproximado entre o início da doença e a morte

CID

40 CAUSAS DA Morte

PARTE I

Doença ou estado mórbido que causou diretamente a morte.

CAUSAS ANTECEDENTES

Estados mórbidos, se existirem, que produziram a causa acima registrada, mencionando-se em último lugar a causa básica.

ANOTE SOMENTE UM DIAGNÓSTICO POR LINHA

a FAISURITÍCOS hemal Aguda

Devido ou como consequência de:

b DISFUNÇÃO de MÚLTIPLOS ÓRGÃOS e SISTEMAS

Devido ou como consequência de:

c CHOCO SEPTICO

Devido ou como consequência de:

d PNEUMONIA

PARTE II

Outras condições significativas que contribuíram para a morte, e que não entraram, porém, na cadeia acima.

ASMA

41 Nome do Médico

CRM

42 CRM

43 Óbito atestado por Médico

1 Assistente

2 Substituto

3 IML

44 Município e UF do SVO ou IML

Bahia

UF

45 Meio de contato (telefone, fax, e-mail, etc)

3267-4100

46 Data do atestado

23.09.2024

47 Assinatura

Dr. Eduardo Bellotto
Reumatologista

CRM-SC 9988 RQE 4395

PROVÁVEIS CIRCUNSTÂNCIAS DE MORTE NÃO NATURAL (Informações de caráter estritamente epidemiológico)

48 Tipo

1 Acidente

3 Homicídio

Ignorado

49 Acidente do trabalho

1 Sim

2 Não

9 Ignorado

50 Fonte de informação

1 Ocorrência Policial N°

2 Hospital

3 Família

4 Amigo

5 Outro

6 Ignorada

Tipo de local de ocorrência do acidente ou violência

1 Via pública

2 Endereço de residência

3 Outro domicílio

4 Estabelecimento comercial

5 Outros

6 Ignorada

51 Descrição sumária do evento

ENDEREÇO DO LOCAL DO ACIDENTE OU VIOLENCIA

52 Logradouro (rua, praça, avenida, etc)

Número

Bairro

Município

UF

53 Cartório

Código

54 Registro

Data

55 Município

UF

56 Declarante

57 UF

58 Testemunhas

VII

Causas externas

VIII

Cartório

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
BIBLIOTECA UNIVERSITÁRIA
REPOSITÓRIO INSTITUCIONAL

CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT

ATESTADO DE VERSÃO FINAL

Eu, PEDRO BERTEMES FILHO, professor(a) do Programa de Pós Graduação em Engenharia Elétrica, declaro que esta é a versão final aprovada pela comissão julgadora da tese intitulada:
“SMACARE - SISTEMA MULTIAGENTE COM INFERÊNCIA NEURAL PARA APOIO DIAGNÓSTICO EM TEMPO REAL E PROGNÓSTICO CLÍNICO”
de autoria do(a) acadêmico(a) TATIANA PEREIRA FILGUEIRAS.

JOINVILLE, 02 de DEZEMBRO de 2025.

Assinatura digital do(a) orientador(a):

PEDRO BERTEMES FILHO