



**FACULDADE CESAR SCHOOL**  
**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**DISCIPLINA DE SISTEMAS EMBARCADOS**  
**RELATÓRIO: SISTEMA IOT COM ESP32**

**JOÃO VICTOR FERRAZ GONÇALVES**  
**MARIA JÚLIA DE OLIVEIRA TELES DE MENEZES**  
**MARIA LUÍSA COIMBRA LIMA**

**RECIFE**  
**DEZEMBRO/2025**

**JOÃO VICTOR FERRAZ GONÇALVES**  
**MARIA JÚLIA DE OLIVEIRA TELES DE MENEZES**  
**MARIA LUÍSA COIMBRA LIMA**

**RELATÓRIO: SISTEMA IOT COM ESP32**

Relatório apresentado para obtenção de nota da 2ª unidade da disciplina de Sistemas Embarcados do Curso de Graduação em Ciência da Computação da Faculdade Cesar School, sob orientação dos Profs. Jimmy Barreto e Bella Nunes.

**RECIFE**  
**DEZEMBRO/2025**

# 1. INTRODUÇÃO

## 1.1 Contexto

A Doença de Parkinson (DP) representa um desafio crescente para a saúde pública mundial. Reconhecida como a segunda doença neurodegenerativa mais prevalente, superada apenas pela Doença de Alzheimer, a DP tem apresentado rápido crescimento em sua incidência e prevalência nas últimas décadas (YANG *et al.*, 2020). Estimativas do estudo Global Burden of Disease indicam que, em 2021, aproximadamente 11,77 milhões de pessoas viviam com a doença em todo o mundo, representando um aumento substancial em relação aos 6,1 milhões registrados em 2016 (LI *et al.*, 2024). A Organização Mundial da Saúde destaca que a prevalência da DP dobrou nos últimos 25 anos, com projeções apontando para 8,5 milhões de indivíduos afetados globalmente em 2019 (WORLD HEALTH ORGANIZATION, 2022).

A doença caracteriza-se pela degeneração progressiva de neurônios dopaminérgicos na substância cinzenta do cérebro, resultando em uma variedade de sintomas motores e não motores que comprometem significativamente a qualidade de vida dos pacientes. Entre as manifestações motoras principais, destaca-se o tremor de repouso, presente em aproximadamente 70% dos casos, que tipicamente apresenta frequência entre 4 e 6 Hz, iniciando-se de forma assimétrica e progredindo ao longo do curso da doença (RIEDER *et al.*, 2003). Este tremor característico, frequentemente descrito como movimento de "contar moedas", diminui ou desaparece durante movimentos voluntários, diferenciando-se de outras formas de tremor patológico.

A prevalência da DP aumenta exponencialmente com a idade, sendo mais comum em indivíduos acima de 65 anos, com incidência particularmente elevada na população acima de 80 anos (YANG *et al.*, 2020). Além disso, observa-se predominância masculina, com homens apresentando risco 1,5 vezes maior de desenvolver a doença em comparação às mulheres (MARRAS *et al.*, 2018). Com o envelhecimento populacional global, projeta-se que o número de casos possa aumentar em 1,5 vezes até 2035, intensificando a carga socioeconômica e de cuidados em saúde associada à doença (LI *et al.*, 2024).

## 1.2 Motivação

O gerenciamento clínico da DP apresenta desafios particulares decorrentes da natureza flutuante dos sintomas e da variabilidade interpessoal na progressão da doença. Tradicionalmente, a avaliação da intensidade e frequência dos tremores parkinsonianos fundamenta-se em escalas clínicas aplicadas de forma episódica durante consultas médicas, sendo a Unified Parkinson's Disease Rating Scale (UPDRS) a mais amplamente utilizada. Entretanto, esta abordagem apresenta limitações importantes, incluindo a subjetividade inerente às escalas de avaliação, a dependência da memória do paciente e cuidadores, e a incapacidade de capturar flutuações sintomáticas que ocorrem no ambiente domiciliar ao longo do dia.

Nas últimas décadas, tem-se observado crescente interesse no desenvolvimento de dispositivos vestíveis equipados com sensores inerciais para monitoramento objetivo e contínuo de sintomas motores em pacientes com DP. Revisões sistemáticas da literatura demonstram que sensores inerciais, particularmente acelerômetros e giroscópios, têm se mostrado ferramentas promissoras para avaliação, diagnóstico e monitoramento de tremores, com aplicações abrangendo desde o diagnóstico precoce até o acompanhamento longitudinal e a avaliação de resposta terapêutica (DEL DIN *et al.*, 2016; ROVINI; MAREMMANI; CAVALLO, 2017).

A integração de tecnologias de Internet das Coisas (IoT) com sensores vestíveis amplia ainda mais as possibilidades de monitoramento remoto e em tempo real. Estudos recentes, como o projeto ELENA, demonstraram a viabilidade de sistemas baseados em sensores MPU6050 integrados a microcontroladores ESP32 para transmissão de dados via Wi-Fi para plataformas em nuvem, permitindo monitoramento contínuo por profissionais de saúde (SANTISTEBAN-ESPEJO *et al.*, 2025). Adicionalmente, a aplicação de algoritmos de processamento de sinais, incluindo análise por Transformada Rápida de Fourier (FFT), possibilita a identificação de frequências dominantes nos tremores, fornecendo biomarcadores objetivos para diferenciação diagnóstica e avaliação de progressão da doença (RIEDER *et al.*, 2014).

A literatura aponta que aproximadamente 75% dos estudos sobre sensores vestíveis em DP focam em demonstrar sensibilidade diagnóstica, estabelecendo correlações entre dados sensoriais e parâmetros clínicos, enquanto apenas uma minoria investiga utilidade clínica definida como melhoria efetiva em desfechos de saúde ou decisões clínicas (POLHEMUS *et al.*, 2024). Este panorama evidencia a necessidade de desenvolvimento de sistemas que não apenas capturem dados objetivos, mas que também os transformem em informações clinicamente relevantes e acessíveis para pacientes, cuidadores e profissionais de saúde.

### 1.3 Objetivos

O presente trabalho tem como objetivo geral o desenvolvimento de um sistema integrado de monitoramento de tremores parkinsonianos denominado Aura IoT, composto por dispositivo vestível com sensores inerciais, backend de processamento e armazenamento de dados, e frontend para visualização e análise. O sistema visa proporcionar monitoramento objetivo, contínuo e em tempo real de tremores, gerando métricas e visualizações clinicamente relevantes para suporte à tomada de decisão no manejo da doença.

Como objetivos específicos, propõe-se desenvolver um dispositivo vestível baseado no sensor MPU6050, capaz de capturar dados triaxiais de aceleração e velocidade angular, com transmissão em tempo real via protocolo MQTT sobre rede Wi-Fi. No âmbito do backend, busca-se implementar uma API em Python para recepção, processamento e armazenamento de dados sensoriais, incorporando algoritmos de processamento de sinais incluindo análise FFT para identificação de frequências dominantes, cálculo de intensidade de tremores, detecção automática de episódios críticos e agregações temporais para análise de padrões.

Quanto ao frontend, pretende-se desenvolver um dashboard interativo e responsivo que apresente visualizações em tempo real dos tremores, indicadores intuitivos de intensidade em escala de 0 a 10, análise espectral para identificação de frequências características entre 4 e 6 Hz, gráficos de amplitude temporal, heatmaps

de intensidade horária, comparações semanais, calendário com indicação visual de dias com diferentes níveis de sintomas, e funcionalidade de exportação de relatórios em formato PDF para compartilhamento com equipe médica

A integração destes componentes busca preencher uma lacuna importante no cuidado de pacientes com Parkinson, transformando dados sensoriais brutos em insights clínicos que possam auxiliar na compreensão de padrões sintomáticos, otimização de esquemas terapêuticos e empoderamento de pacientes e cuidadores através do acesso a informações objetivas sobre a evolução da doença.

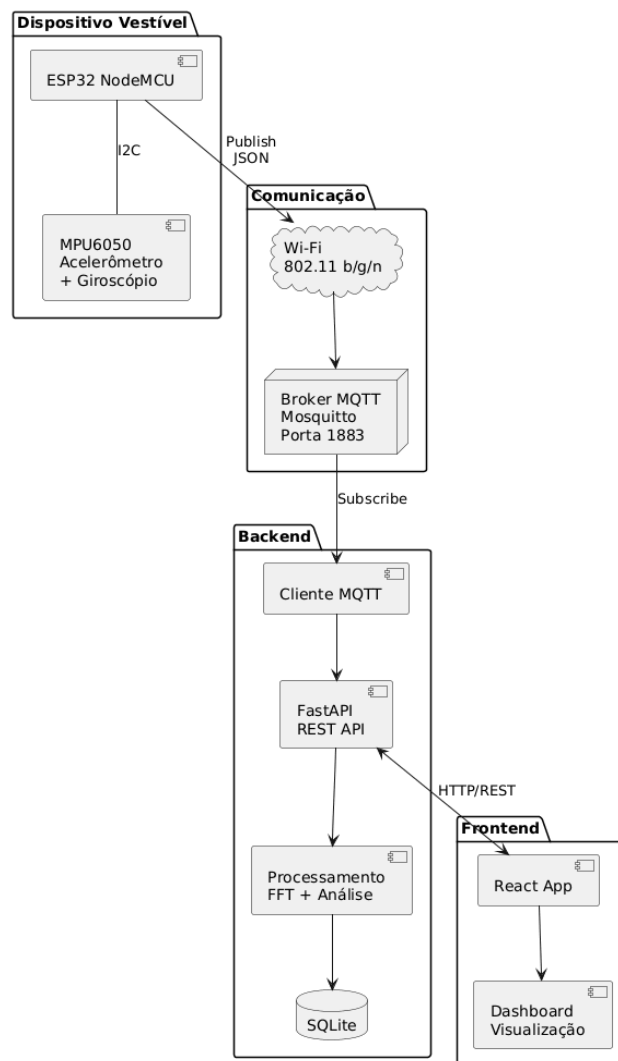
## **2. METODOLOGIA**

A metodologia empregada no desenvolvimento do sistema Aura IoT fundamenta-se na integração de três componentes principais: hardware de aquisição de dados, infraestrutura de comunicação e processamento, e interface de visualização. Esta seção detalha a arquitetura do sistema, os componentes utilizados e o fluxo de comunicação estabelecido para viabilizar o monitoramento contínuo e em tempo real de tremores parkinsonianos.

### **2.1 Diagrama do Sistema**

A arquitetura do sistema Aura IoT adota um modelo cliente-servidor baseado no paradigma publish-subscribe, amplamente utilizado em aplicações de Internet das Coisas. A Figura 1 apresenta o diagrama esquemático da arquitetura proposta, ilustrando a interação entre os componentes do sistema.

Figura 1: Diagrama de arquitetura do sistema Aura IoT



Fonte: Elaborado pelo próprio autor (2025)

O sistema estrutura-se em três camadas distintas: camada de sensoriamento, camada de comunicação e processamento, e camada de apresentação. Na camada de sensoriamento, o dispositivo vestível equipado com o sensor MPU6050 captura dados triaxiais de aceleração e velocidade angular em frequência configurável. Estes dados são transmitidos via Wi-Fi utilizando o protocolo MQTT para um broker local, responsável por gerenciar a distribuição de mensagens entre publicadores e assinantes.



Na camada de comunicação e processamento, o broker MQTT, implementado localmente através do software Mosquitto, atua como intermediário na arquitetura publish-subscribe. O protocolo MQTT caracteriza-se por seu design leve e eficiente, sendo ideal para dispositivos com recursos computacionais limitados e redes com largura de banda restrita (MQTT.ORG, 2024). A API desenvolvida em FastAPI subscreve aos tópicos MQTT relevantes, recebendo os dados sensoriais em tempo real, processando-os através de algoritmos de análise de sinais e armazenando-os em banco de dados SQLite para consultas históricas.

Por fim, na camada de apresentação, a aplicação frontend desenvolvida em React consome os dados processados através de endpoints REST da API, apresentando visualizações interativas e métricas clinicamente relevantes aos usuários finais. A comunicação entre frontend e backend ocorre via protocolo HTTP/HTTPS, enquanto atualizações em tempo real são viabilizadas através de mecanismos de polling ou websockets.

## 2.2 Hardware e Software

### 2.2.1 *Componentes de Hardware*

O dispositivo vestível desenvolvido utiliza como plataforma principal o microcontrolador ESP32. O ESP32 caracteriza-se por integrar conectividade Wi-Fi e Bluetooth e suporte nativo a Real-Time Operating System (RTOS), tornando-o adequado para aplicações IoT que demandam processamento paralelo de tarefas (ESPRESSIF SYSTEMS, 2024).

Para aquisição de dados inerciais, empregou-se o sensor MPU6050, um sistema micro-eletromecânico (MEMS) que integra acelerômetro triaxial e giroscópio triaxial em um único encapsulamento. O MPU6050 disponibiliza conversores analógico-digitais de 16 bits para cada eixo de medição, oferecendo escalas programáveis para o acelerômetro e para o giroscópio (INVENSENSE, 2013). A comunicação entre o MPU6050 e o ESP32 realiza-se através da interface I2C (Inter-Integrated Circuit),

protocolo serial síncrono que opera com apenas dois fios: SDA (Serial Data) e SCL (Serial Clock).

A montagem física do dispositivo seguiu o esquema de conexões apresentado na Tabela 1. O sensor MPU6050 foi conectado ao microcontrolador ESP32 através de jumpers, com alimentação de 3,3V proveniente do regulador de tensão do próprio ESP32. As linhas de comunicação I2C foram conectadas aos pinos GPIO 21 (SDA) e GPIO 22 (SCL), que correspondem às linhas I2C padrão do ESP32.

Tabela 1: Esquema de conexões entre MPU6050 e ESP32

Pino MPU 6050	Pino ESP32
VCC	3.3V
GND	GND
SDA	GPIO 21
SCL	GPIO 22

Para garantir conforto e usabilidade durante períodos prolongados de monitoramento, o conjunto sensor-microcontrolador foi fixado a uma pulseira elástica, permitindo posicionamento estável no punho do paciente enquanto mantém liberdade de movimento.

### 2.2.2 Firmware e Bibliotecas

O firmware embarcado no ESP32 foi desenvolvido utilizando o framework Arduino sobre o PlatformIO, ambiente de desenvolvimento integrado para sistemas embarcados que oferece gerenciamento avançado de bibliotecas e suporte a múltiplas plataformas. A implementação incorporou o sistema operacional de tempo real FreeRTOS, que possibilita a execução concorrente de tarefas com diferentes

prioridades, essencial para garantir a coleta contínua de dados sensoriais simultaneamente à transmissão via Wi-Fi.

As bibliotecas incorporadas ao firmware incluem: `Arduino.h` para funções básicas do framework Arduino; `Wire.h` para comunicação I2C; `Adafruit_MPU6050.h` e `Adafruit_Sensor.h` para interface de alto nível com o sensor MPU6050; `WiFi.h` para conectividade sem fio; e `PubSubClient.h` para implementação do protocolo MQTT. A biblioteca `Adafruit_MPU6050` abstrai a complexidade da comunicação de baixo nível com o sensor, oferecendo métodos simplificados para configuração de escalas, frequência de amostragem e leitura de dados.

### *2.2.3 Infraestrutura de Backend*

A camada de backend foi implementada utilizando FastAPI, framework Python moderno para desenvolvimento de APIs RESTful que oferece alto desempenho, validação automática de dados. A arquitetura do backend estrutura-se em módulos especializados: `main.py` configura a aplicação e define middlewares; `models.py` define os modelos de dados utilizando SQLAlchemy ORM; `schemas.py` especifica os esquemas de validação com Pydantic; `db.py` gerencia conexões com o banco de dados SQLite; `mqtt_client.py` implementa o cliente MQTT que subscreve aos tópicos relevantes; e os diretórios `routes/` e `services/` organizam respectivamente os endpoints da API e a lógica de negócio.

O processamento de sinais incorpora análise por Transformada Rápida de Fourier (FFT) implementada através da biblioteca NumPy, permitindo identificação das frequências dominantes nos tremores. Adicionalmente, foram implementados algoritmos para cálculo de magnitude vetorial total da aceleração e velocidade angular, detecção automática de episódios de tremor através de limiares adaptativos, e agregações temporais para análise de padrões diários e semanais.

O armazenamento persistente utiliza SQLite, sistema de gerenciamento de banco de dados relacional embarcado que não requer servidor separado, adequado

para aplicações de pequeno a médio porte. A estrutura do banco contempla tabelas para armazenamento de leituras sensoriais brutas com timestamps, métricas processadas, e eventos detectados.

#### *2.2.4 Interface Frontend*

A interface de usuário foi desenvolvida como Single Page Application (SPA) utilizando React 18, biblioteca JavaScript declarativa para construção de interfaces interativas. O gerenciamento de estado utiliza hooks nativos do React (useState, useEffect, useContext), enquanto a estilização emprega Tailwind CSS, framework utilitário que permite desenvolvimento ágil de interfaces responsivas. A comunicação com o backend realiza-se através da API Fetch nativa do navegador, com requisições organizadas em módulos de serviço para promover reutilização e manutenibilidade.

As visualizações de dados incorporam bibliotecas especializadas, incluindo Recharts para gráficos de linha e área, permitindo acompanhamento temporal da intensidade dos tremores, e bibliotecas de heatmap para visualização de padrões horários. A interface foi projetada seguindo princípios de design responsivo, garantindo usabilidade em diferentes dispositivos, desde smartphones até monitores desktop.

### 2.3 Fluxo de Comunicação (MQTT + Wi-Fi)

O fluxo de comunicação do sistema Aura IoT fundamenta-se no protocolo MQTT (Message Queuing Telemetry Transport), padrão ISO/IEC 20922 amplamente adotado em aplicações IoT devido à sua leveza, confiabilidade e eficiência em redes com latência elevada ou largura de banda limitada. O MQTT implementa arquitetura publish-subscribe, na qual dispositivos clientes publicam mensagens em tópicos específicos gerenciados por um broker central, que as distribui aos clientes assinantes interessados (BANKS; GUPTA, 2014).

No contexto do Aura IoT, o dispositivo vestível atua como cliente MQTT publicador, conectando-se ao broker Mosquitto executado localmente através da rede Wi-Fi doméstica. Após estabelecimento da conexão TCP/IP na porta padrão 1883, o dispositivo autentica-se no broker e publica mensagens contendo dados sensoriais no tópico parkinson/mpu6050. As mensagens são formatadas em JSON (JavaScript Object Notation), facilitando parsing tanto no backend Python quanto no frontend JavaScript. A estrutura típica das mensagens publicadas inclui campos para timestamp, aceleração nos eixos x, y e z, e velocidade angular nos mesmos eixos.

O protocolo MQTT oferece três níveis de Quality of Service (QoS) que determinam garantias de entrega: QoS 0 (at most once) não garante entrega nem retransmissão; QoS 1 (at least once) garante que a mensagem seja entregue pelo menos uma vez, podendo haver duplicatas; e QoS 2 (exactly once) garante entrega única através de handshake de quatro etapas (MQTT.ORG, 2024). Para o presente sistema, adotou-se QoS 1, balanceando confiabilidade e sobrecarga de comunicação, considerando que pequenas perdas de dados sensoriais individuais não comprometem a análise geral de padrões de tremores.

Do lado do servidor, o cliente MQTT implementado no backend Python subscreve ao tópico parkinson/mpu6050, recebendo as mensagens publicadas pelo dispositivo em tempo real. Ao receber cada mensagem, o backend executa sequencialmente: parsing do payload JSON; validação dos dados contra esquemas Pydantic; processamento através de algoritmos de análise de sinais; armazenamento no banco de dados SQLite; e disponibilização via endpoints REST para consumo pelo frontend. Este fluxo assíncrono permite que o frontend mantenha visualizações atualizadas através de requisições periódicas aos endpoints da API ou, alternativamente, através de conexões WebSocket para push de dados em tempo real.

A escolha da rede Wi-Fi como meio de transmissão justifica-se pela ubiquidade de redes domésticas e facilidade de configuração.

### 3. RESULTADOS

A implementação do sistema Aura IoT resultou em uma plataforma funcional capaz de realizar monitoramento contínuo de tremores parkinsonianos através da integração efetiva de hardware sensorial, infraestrutura de comunicação e interface de visualização. Esta seção apresenta os resultados obtidos em termos de funcionalidades do dashboard desenvolvido, características dos dados coletados e análise do funcionamento geral do sistema.

#### 3.1 Dashboard

O dashboard desenvolvido constitui a interface primária de interação entre o sistema e seus usuários finais, incluindo pacientes, cuidadores e profissionais de saúde. O dashboard foi estruturado para transformar dados sensoriais brutos em informações clinicamente relevantes e facilmente interpretáveis. A literatura aponta que dashboards efetivos em saúde devem priorizar clareza visual, evitar sobrecarga cognitiva e organizar informações de forma hierárquica, facilitando a rápida identificação de tendências e anomalias (THINKITIVE, 2024; FUSELAB CREATIVE, 2025).

A interface foi organizada em módulos funcionais distintos, cada um atendendo a necessidades específicas de monitoramento. O módulo de visualização em tempo real apresenta gráficos de linha atualizados continuamente, exibindo a magnitude vetorial total da aceleração nos três eixos espaciais. Esta visualização permite acompanhamento imediato dos tremores, possibilitando que cuidadores identifiquem rapidamente episódios de exacerbação sintomática.

O indicador de intensidade de tremor, implementado como escala visual de 0 a 10, utiliza representação tipo termômetro com gradiente de cores variando de verde (baixa intensidade) a vermelho (alta intensidade). A escala foi calibrada empiricamente, estabelecendo-se que valores entre 0-3 correspondem a tremor leve, 4-6 a tremor moderado e 7-10 a tremor severo, permitindo classificação rápida da condição do paciente.

Um painel de métricas instantâneas exibe continuamente os últimos valores registrados para aceleração total, velocidade angular total e frequência estimada do tremor. Adicionalmente, um indicador de status do sensor informa sobre a qualidade da conexão e taxa de perda de pacotes, permitindo identificação de problemas técnicos que possam comprometer a confiabilidade dos dados.

O módulo de análise espectral implementa visualização dos resultados da Transformada Rápida de Fourier aplicada aos dados de aceleração. A FFT constitui técnica fundamental em processamento de sinais, convertendo dados do domínio temporal para o domínio frequencial, permitindo identificação das componentes de frequência presentes no sinal (COOLEY; TUKEY, 1965). No contexto de tremores parkinsonianos, a análise FFT possibilita identificação objetiva da frequência dominante, característica diagnóstica importante dado que tremores parkinsonianos tipicamente manifestam-se na faixa de 4-6 Hz. A visualização apresenta o espectro de potência em gráfico de barras, com destaque visual para picos de frequência nesta faixa característica, facilitando identificação de padrões consistentes com Doença de Parkinson.

O módulo de análise temporal foi implementado através de múltiplas visualizações complementares. Um gráfico de área exibe a evolução da intensidade dos tremores ao longo do dia, com o eixo horizontal representando o tempo e o eixo vertical a intensidade calculada. Esta visualização permite identificação de padrões e períodos de maior ou menor sintomatologia, informações valiosas para ajustes na posologia medicamentosa. A agregação temporal agrupa dados em intervalos de uma hora, calculando-se medidas estatísticas como média, desvio padrão, valor máximo e percentil 95 para cada intervalo.

A visualização tipo heatmap oferece representação matricial da intensidade de tremores ao longo das horas do dia. Células individuais são coloridas segundo escala de intensidade, permitindo identificação visual imediata de padrões temporais. Esta representação é particularmente útil para identificar se tremores concentram-se em períodos específicos do dia, informação relevante para correlação com horários de

medicação, refeições ou atividades específicas. Existe ainda a análise comparativa semanal que permite visualização de métricas agregadas por dias da semana, apresentando gráficos de barras que facilitam identificação de tendências de longo prazo

O módulo de detecção de episódios implementa algoritmo baseado em limiar adaptativo que identifica automaticamente períodos em que a intensidade do tremor excede valores predefinidos por duração mínima configurável. Cada episódio detectado é registrado com timestamp de início e fim, duração total, intensidade máxima atingida e frequência dominante durante o episódio. Estes episódios são apresentados em lista cronológica com indicadores visuais de severidade, permitindo que profissionais de saúde identifiquem rapidamente os períodos mais críticos.

O módulo de calendário com visão mensal traz informações sobre o conceito de "dias bons" versus "dias ruins", categorizando automaticamente cada dia baseado em métricas como intensidade média, número de episódios detectados e duração total de tremor ativo. A interface tipo calendário utiliza código de cores para representar visualmente a qualidade de cada dia, permitindo identificação rápida de padrões semanais ou mensais.

Finalmente, o sistema incorpora funcionalidade de geração de relatórios em formato PDF, permitindo compilação de dados e visualizações selecionadas para períodos específicos. Os relatórios gerados constituem documentação adequada para compartilhamento com equipe médica durante consultas de acompanhamento.

### 3.2 Dados Coletados

A operação do sistema Aura IoT resulta na coleta contínua de dados sensoriais triaxiais provenientes do acelerômetro e giroscópio integrados ao MPU6050. Cada amostra capturada consiste de seis valores numéricos de 16 bits representando aceleração linear nos eixos x, y e z, e velocidade angular nos mesmos três eixos.



Os dados brutos de aceleração são expressos em unidades de gravidade terrestre (g), enquanto os dados de velocidade angular são expressos em graus por segundo ( $^{\circ}/s$ ).

O formato das mensagens MQTT publicadas pelo dispositivo vestível segue estrutura JSON padronizada, incluindo timestamp para sincronização temporal precisa e arrays de valores para aceleração e velocidade angular.

A magnitude vetorial total da aceleração, calculada como raiz quadrada da soma dos quadrados das acelerações nos três eixos, constitui métrica fundamental derivada dos dados brutos. Esta métrica oferece representação unificada da intensidade total do movimento independente de orientação do sensor, facilitando comparações longitudinais e reduzindo sensibilidade a variações no posicionamento do dispositivo vestível. Similarmente, a magnitude vetorial da velocidade angular fornece métrica unificada da taxa de rotação.

A aplicação da Transformada Rápida de Fourier aos dados coletados permite extração de características espectrais relevantes. Estudos demonstram que análise FFT aplicada a dados de acelerômetros de pulso pode efetivamente detectar e quantificar tremores em pacientes com esclerose múltipla (BAKER *et al.*, 2021).

### 3.3 Funcionamento

O funcionamento integrado do sistema Aura IoT demonstrou-se estável e confiável durante testes operacionais. O processo inicia-se com a energização do dispositivo vestível, que executa sequência de inicialização incluindo configuração do sensor MPU6050, estabelecimento de conexão Wi-Fi com a rede doméstica e autenticação junto ao broker MQTT.

Uma vez estabelecida a conexão MQTT, o dispositivo inicia publicação periódica de mensagens no tópico configurado. No lado do servidor, o cliente MQTT implementado no backend Python mantém conexão persistente com o broker,

recebendo mensagens publicadas pelo dispositivo. Ao receber cada mensagem, o sistema executa pipeline de processamento que inclui parsing JSON, validação de integridade, cálculo de métricas derivadas e armazenamento em banco de dados.

O frontend React consulta a API backend em intervalos regulares para obtenção de dados atualizados, utilizando endpoints REST que retornam JSON contendo métricas recentes e resultados de análises. A renderização das visualizações ocorre de forma fluida, sem travamentos, mesmo quando múltiplos gráficos são atualizados simultaneamente.

A funcionalidade de exportação de relatórios PDF foi testada com sucesso, gerando documentos estruturados contendo visualizações e métricas selecionadas. Os relatórios gerados apresentaram formatação adequada e qualidade visual satisfatória para impressão ou compartilhamento digital com profissionais de saúde.

#### 4. CONCLUSÃO

O presente trabalho apresentou o desenvolvimento e implementação do sistema Aura IoT, uma plataforma integrada para monitoramento contínuo de tremores parkinsonianos baseada em tecnologias de Internet das Coisas. A solução proposta demonstrou viabilidade técnica ao integrar efetivamente dispositivo vestível equipado com sensor MPU6050, infraestrutura de comunicação via protocolo MQTT sobre rede Wi-Fi e interface web interativa para visualização de dados. O sistema mostrou-se capaz de capturar dados sensoriais triaxiais em tempo real, processá-los através de algoritmos de análise de sinais incluindo Transformada Rápida de Fourier, e apresentar informações clinicamente relevantes através de dashboard estruturado segundo princípios de design centrado no usuário.

Os resultados obtidos evidenciam que a abordagem proposta oferece alternativa promissora aos métodos tradicionais de avaliação episódica baseados em escalas clínicas subjetivas. A capacidade de monitoramento contínuo em ambiente domiciliar, aliada à quantificação objetiva de parâmetros como intensidade, frequência dominante e identificação automática de episódios críticos, fornece informações complementares valiosas que podem subsidiar decisões terapêuticas mais fundamentadas. A visualização temporal através de gráficos de tendência, heatmaps e análise comparativa entre dias possibilita identificação de padrões e correlações com fatores externos como medicação e atividades, informações frequentemente não capturadas durante consultas médicas de curta duração. Adicionalmente, a funcionalidade de geração de relatórios em formato PDF facilita o compartilhamento estruturado de dados com equipes multidisciplinares de saúde, promovendo melhor comunicação entre pacientes, cuidadores e profissionais.

Apesar dos resultados promissores, reconhecem-se limitações e oportunidades para desenvolvimentos futuros. A validação clínica do sistema com população de pacientes diagnosticados com Doença de Parkinson constitui etapa essencial para estabelecimento de correlações robustas entre métricas calculadas e severidade clínica avaliada por escalas padronizadas como UPDRS. A expansão das funcionalidades de

processamento para incluir técnicas avançadas de aprendizado de máquina poderia aprimorar capacidades de classificação automática de tipos de tremor e previsão de episódios. A integração com dispositivos vestíveis comerciais como smartwatches poderia reduzir barreiras de adoção ao eliminar necessidade de hardware dedicado. Finalmente, estudos longitudinais com uso prolongado do sistema seriam valiosos para avaliar impacto na qualidade de vida dos pacientes e na efetividade do manejo clínico. O sistema Aura IoT estabelece fundação tecnológica sólida para evolução de soluções de monitoramento remoto que contribuam efetivamente para melhoria do cuidado de pessoas vivendo com Doença de Parkinson.

## 6. APÊNDICE

### APÊNDICE A – Estrutura do Repositório

O código-fonte completo do sistema Aura IoT encontra-se disponível publicamente no repositório GitHub: [https://github.com/mjuliamezenezes/Aura\\_IoT](https://github.com/mjuliamezenezes/Aura_IoT)

A estrutura de diretórios do projeto está organizada conforme apresentado a seguir:

```
Aura_IoT/
|
|— backend/
|   |— app/
|       |— routes/
|           |— episodes_routes.py
|           |— features_routes.py
|           |— heatmap_routes.py
|           |— realtime_routes.py
|           |— stats_routes.py
|           |— services/
|               |— episodes_service.py
|               |— features_repository.py
|               |— features_service.py
|               |— heatmap_service.py
|               |— realtime_service.py
|               |— stats_service.py
|           |— db.py
|           |— main.py
|           |— models.py
|           |— mqtt_client.py
|           |— schemas.py
|       |— Dockerfile
|       |— requirements.txt
|       |— quick_populate.py
|       |— test_stats.py
|
```

```
├── device/
│   ├── firmware/
│   │   ├── .vscode/
│   │   ├── include/
│   │   ├── lib/
│   │   ├── src/
│   │   │   └── main.cpp
│   │   ├── test/
│   │   ├── .gitignore
│   │   └── platformio.ini
│
├── frontend/
│   ├── public/
│   ├── src/
│   │   ├── assets/
│   │   ├── components/
│   │   ├── lib/
│   │   ├── services/
│   │   ├── types/
│   │   ├── App.css
│   │   ├── App.tsx
│   │   ├── index.css
│   │   └── main.tsx
│   ├── .gitignore
│   ├── components.json
│   ├── eslint.config.js
│   ├── index.html
│   ├── package-lock.json
│   ├── package.json
│   ├── postcss.config.js
│   ├── tailwind.config.js
│   ├── tsconfig.app.json
│   ├── tsconfig.json
│   └── tsconfig.node.json
```

```
|   └─ vite.config.ts
|
|─ infra/
|   └─ docker-compose.yml
|   └─ mosquitto.conf
|
└─ README.md
```

## REFERÊNCIAS

BAKER, M. R. et al. Quantifying upper limb tremor in people with multiple sclerosis using Fast Fourier Transform based analysis of wrist accelerometer signals. **Clinical Neurophysiology**, v. 132, n. 5, p. 1133-1141, 2021.

BANKS, A.; GUPTA, R. (Ed.). **MQTT Version 3.1.1**. OASIS Standard, 2014. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Acesso em: 7 dez. 2024.

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of Computation**, v. 19, n. 90, p. 297-301, 1965.

DEL DIN, S. et al. **Measuring gait with an accelerometer-based wearable**: influence of device location, testing protocol and age. *Physiological Measurement*, v. 37, n. 10, p. 1785-1797, 2016.

ESPRESSIF SYSTEMS. **ESP32 Technical Reference Manual**. Version 4.8. Shanghai: Espressif Systems, 2024.

FUSELAB CREATIVE. **Healthcare Dashboard Design Best Practices and Key Considerations**. 2025. Disponível em: <https://fuselabcreative.com/healthcare-dashboard-design-best-practices/>. Acesso em: 7 dez. 2024.

INVENSENSE. **MPU-6000 and MPU-6050 Product Specification**. Revision 3.4. Sunnyvale: InvenSense Inc., 2013.

LI, Y. et al. **Global, regional, national epidemiology and trends of Parkinson's disease from 1990 to 2021**: findings from the Global Burden of Disease Study 2021. *Frontiers in Aging Neuroscience*, v. 16, 2024.

MARRAS, C. et al. **Prevalence of Parkinson's disease across North America**. *NPJ Parkinson's Disease*, v. 4, n. 1, p. 21, 2018.

MQTT.ORG. **MQTT: The Standard for IoT Messaging**. 2024. Disponível em: <https://mqtt.org/>. Acesso em: 7 dez. 2024.

POLHEMUS, A. et al. **Assessing the clinical utility of inertial sensors for home monitoring in Parkinson's disease**: a comprehensive review. *NPJ Parkinson's Disease*, v. 10, n. 1, p. 162, 2024.

RIEDER, C. R. M. et al. **Diagnóstico diferencial dos tremores**. *Arquivos de Neuro-Psiquiatria*, v. 61, n. 3B, p. 898-904, 2003.

RIEDER, C. R. M. et al. **Electrophysiologic characteristics of tremor in Parkinson's disease and essential tremor**. *Arquivos de Neuro-Psiquiatria*, v. 72, n. 4, p. 301-306, 2014.



ROVINI, E.; MAREMMANI, C.; CAVALLO, F. **How wearable sensors can support Parkinson's disease diagnosis and treatment:** a systematic review. *Frontiers in Neuroscience*, v. 11, p. 555, 2017.

SANTISTEBAN-ESPEJO, A. et al. **A wearable Internet of Things-based device for the quantitative assessment of hand tremors in Parkinson's disease:** the ELENA project. *Sensors*, v. 25, n. 9, p. 2763, 2025.

THINKITIVE. **Best Practices in Healthcare Dashboard Design.** 2024. Disponível em: <https://www.thinkitive.com/blog/best-practices-in-healthcare-dashboard-design/>. Acesso em: 7 dez. 2024.

WORLD HEALTH ORGANIZATION. **Launch of WHO's Parkinson disease technical brief.** Geneva: WHO, 2022. Disponível em: <https://www.who.int/news/item/14-06-2022-launch-of-who-s-parkinson-disease-technical-brief>. Acesso em: 7 dez. 2024.

YANG, W. et al. **Global trends in the incidence, prevalence, and years lived with disability of Parkinson's disease in 204 countries/territories from 1990 to 2019.** *Frontiers in Public Health*, v. 9, p. 776847, 2020.