



Monitoramento Residencial

Sua saúde, Nossa missão

Monitoramento e Assistência
Homecare

Felipe Leme
Vitor Gomes



Introdução:

O projeto surgiu da urgência em melhorar os cuidados com idosos e pessoas com doenças crônicas, que demandam atenção constante.

Motivações principais:

- Crescimento da população idosa e manutenção e acompanhamento de pessoas com doenças crônicas;
- Maior risco de quedas e emergências cardiorrespiratórias;
- Necessidade de monitoramento contínuo de batimentos cardíacos e oxigenação do sangue.

Necessidades identificadas:

- Resposta rápida em situações críticas;
- Registro histórico confiável dos dados médicos;
- Acionamento imediato em casos de quedas ou desmaios.

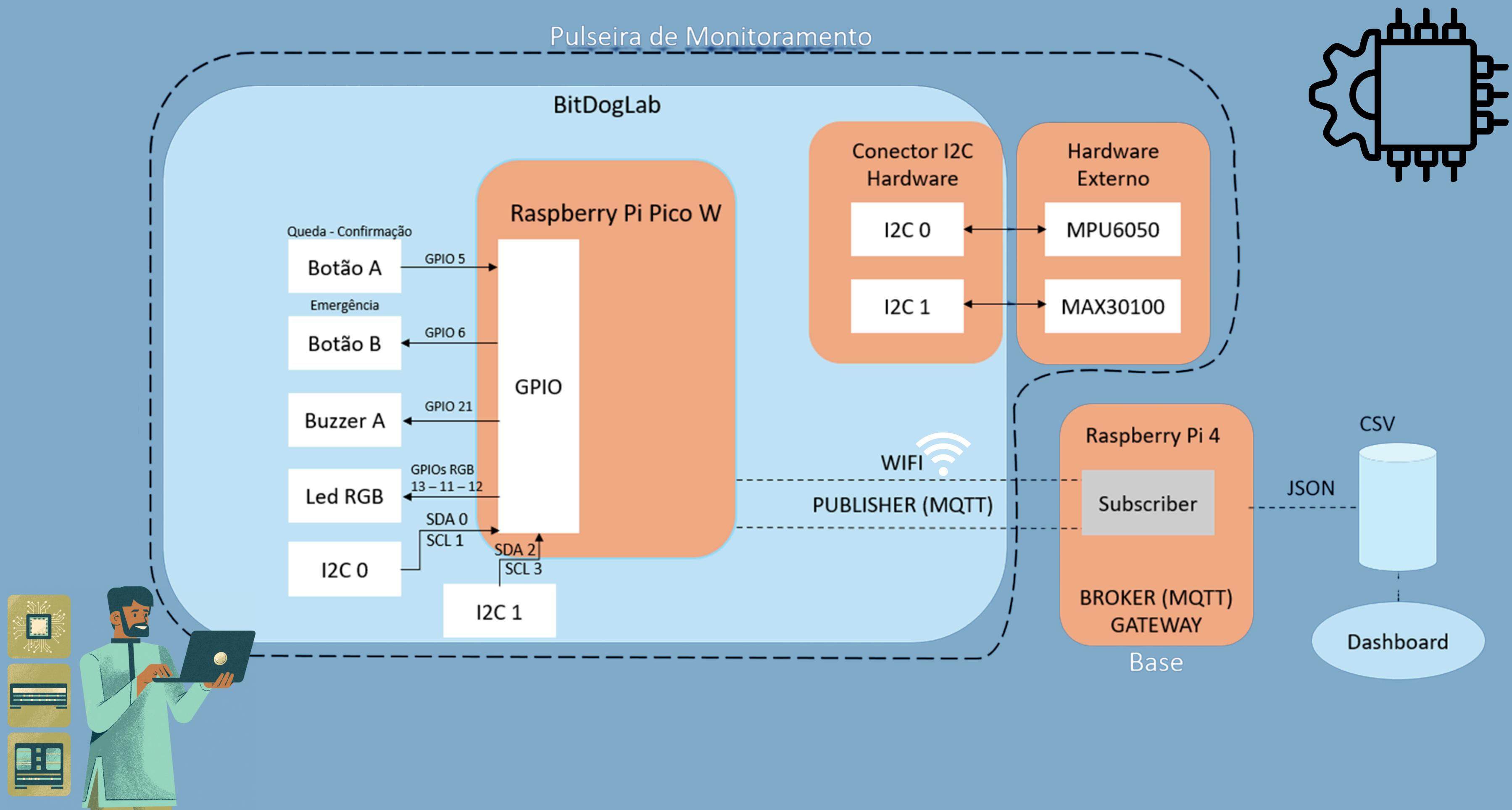


O sistema foi desenvolvido para garantir segurança e precisão no monitoramento, com os seguintes objetivos:

- Medir BPM e SpO₂ com confiabilidade;
- Detectar quedas reais, evitando falsos alarmes;
- Permitir acionamento manual (botão de emergência);
- Emitir alarmes temporizados (ex.: lembrete de beber água);
- Enviar dados via MQTT;
- Registrar informações em CSV na Raspberry Pi 4;
- Disponibilizar um dashboard para análise médica.



Arquitetura do Hardware – Pulseira (RP2040 – BitDogLab) e Base (Raspberry Pi 4)



Arquitetura da pulseira:

Microcontrolador:

- Raspberry Pi Pico W (Wi-Fi integrado).

Sensores:

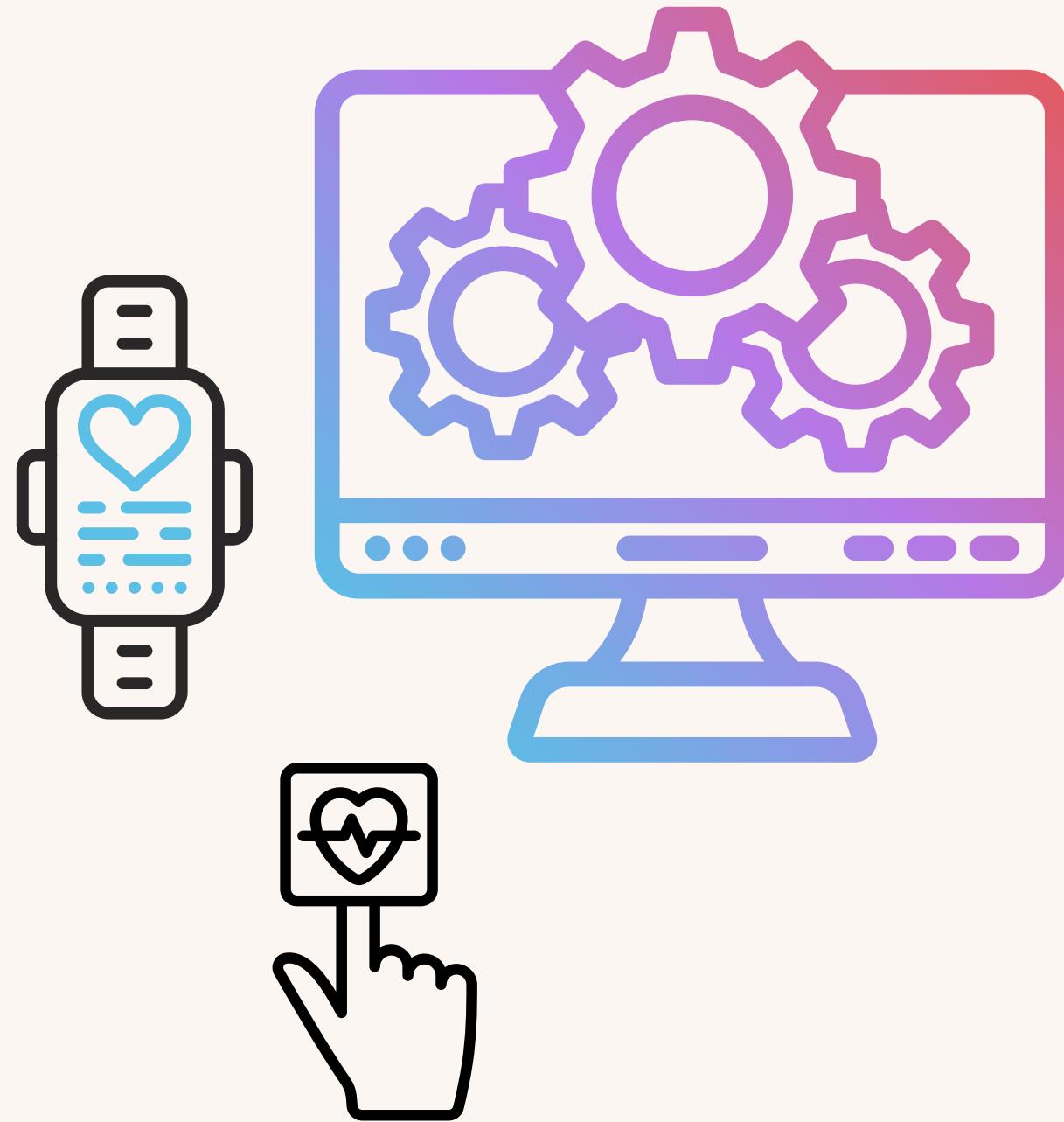
- MAX30100 → BPM e SpO₂ (fotopletismografia);
- MPU6050 → aceleração e giroscópio (quedas).

Atuadores:

- LED RGB + Buzzer;
- Botão de emergência para interação manual;
- Botão de atuação – Confirmação de queda.

Software:

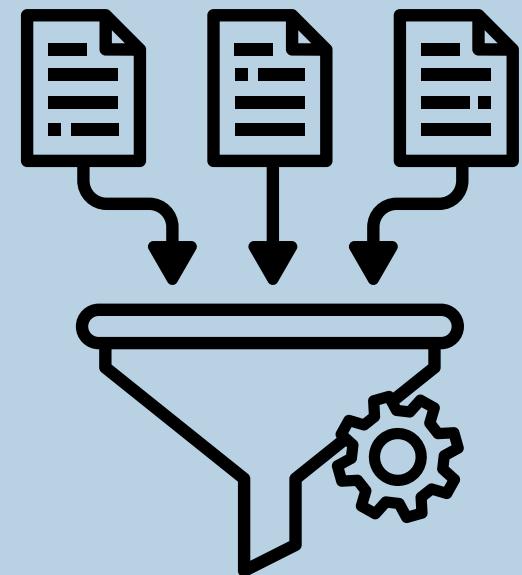
- FreeRTOS, Wi-Fi e MQTT.



Coleta de dados-MPU6050

Com o objetivo de verificar a condição do paciente, realizamos coletas iniciais que mostraram:

- Leituras simples, sem thresholds, geravam falsos positivos e se mostraram instáveis.
- Solução adotada: Implementamos uma sequência de thresholds, garantindo parametrizações mais precisas.



Realizamos uma série de testes em diferentes condições: correndo, sentado, deitado (cama e chão), levantando e agachando. A partir disso, definimos os thresholds ideais para detecção de quedas.



Coleta de dados—MPU6050

Thresholds finais:

- Impacto > 1.8 g
- Instabilidade > 0.6 g
- Imobilidade < 0.55 g

Máquina de estados (sequência obrigatória):

Monitoramento → Impacto → Instabilidade → Imobilidade → Janela de 10 s p/ cancelar → Queda confirmada.

Defesas contra falso positivo:

- Botão: usuário pode cancelar em 10 s (pós-evento).
- Debounce de arranque: suprime alarme logo ao ligar o dispositivo.



Coleta de sinais-MAX30102

Com o objetivo de coletar dados continuamente e em tempo real, o MAX30102 foi definido para:

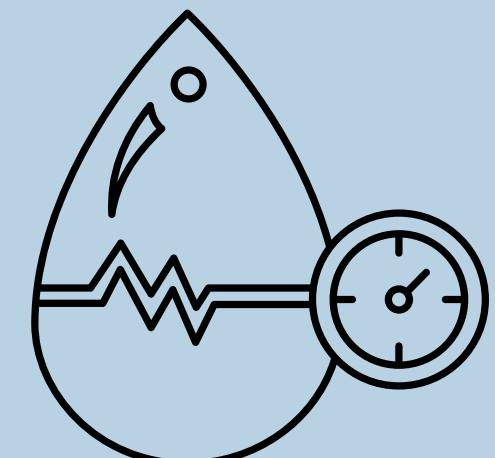
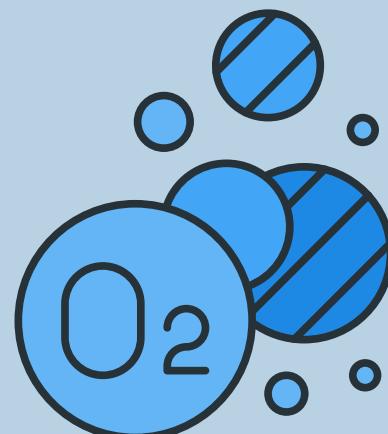
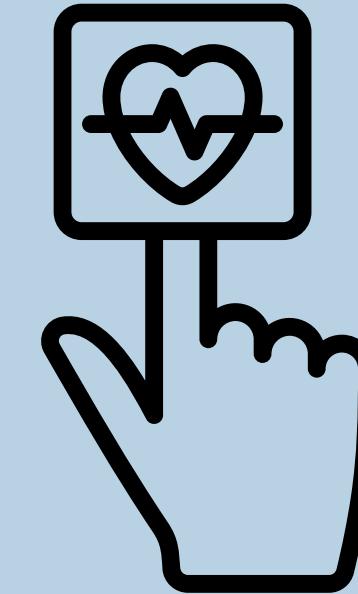
- Medir batimentos cardíacos (BPM) e oxigenação do sangue (SpO_2).

Desafios e soluções:

- Leituras no pulso apresentavam ruído → solução: coleta preferencial no dedo.

Melhorias na coleta:

- Uso de filtros e suavização para garantir estabilidade;
- Envio periódico dos dados via MQTT;
- Registro em CSV com timestamp, permitindo histórico confiável.



Transporte dos Dados

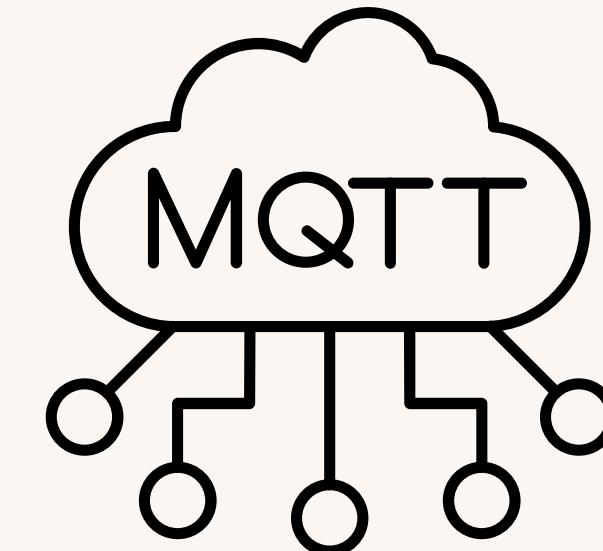
Adotamos o protocolo MQTT para comunicação e transferência de dados via Wi-Fi (biblioteca lwIPopts para configuração da rede).

Estrutura definida:

- Tópicos claros:
- homecare/bracelet/<id>/vitals → BPM
- SpO₂homecare/bracelet/<id>/event/emergency → botão/queda

Payload (conteúdo da mensagem):

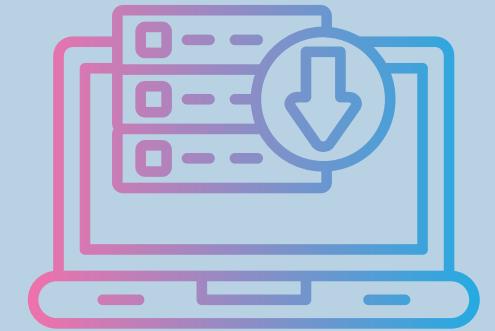
- JSON com ts_ms (timestamp de origem) → garante rastreabilidade.
- QoS (Quality of Service) sob medida (garante a entrega da mensagem):
 - QoS 1 → eventos críticos (garantia de entrega).
 - QoS 0 → sinais vitais (baixa latência e menos overhead) – não são críticos, então, podem ser enviados apenas uma vez.



Base-Raspberry Pi 4 (Servidor Local/Elemento de Borda)

A Raspberry Pi 4 atua como nosso servidor local, acumulando três funções:

- Broker MQTT (Mosquitto): gerencia os tópicos.
- Subscriber: recebe os dados enviados pela pulseira.
- Publisher: organiza e encaminha os dados para armazenamento local.



Serviço em systemd – `homecare-logger.service`:

- Inicia automaticamente com o sistema.
- Reinicia em caso de falha.
- Permite monitoramento via journalctl.



Script Python (logger):

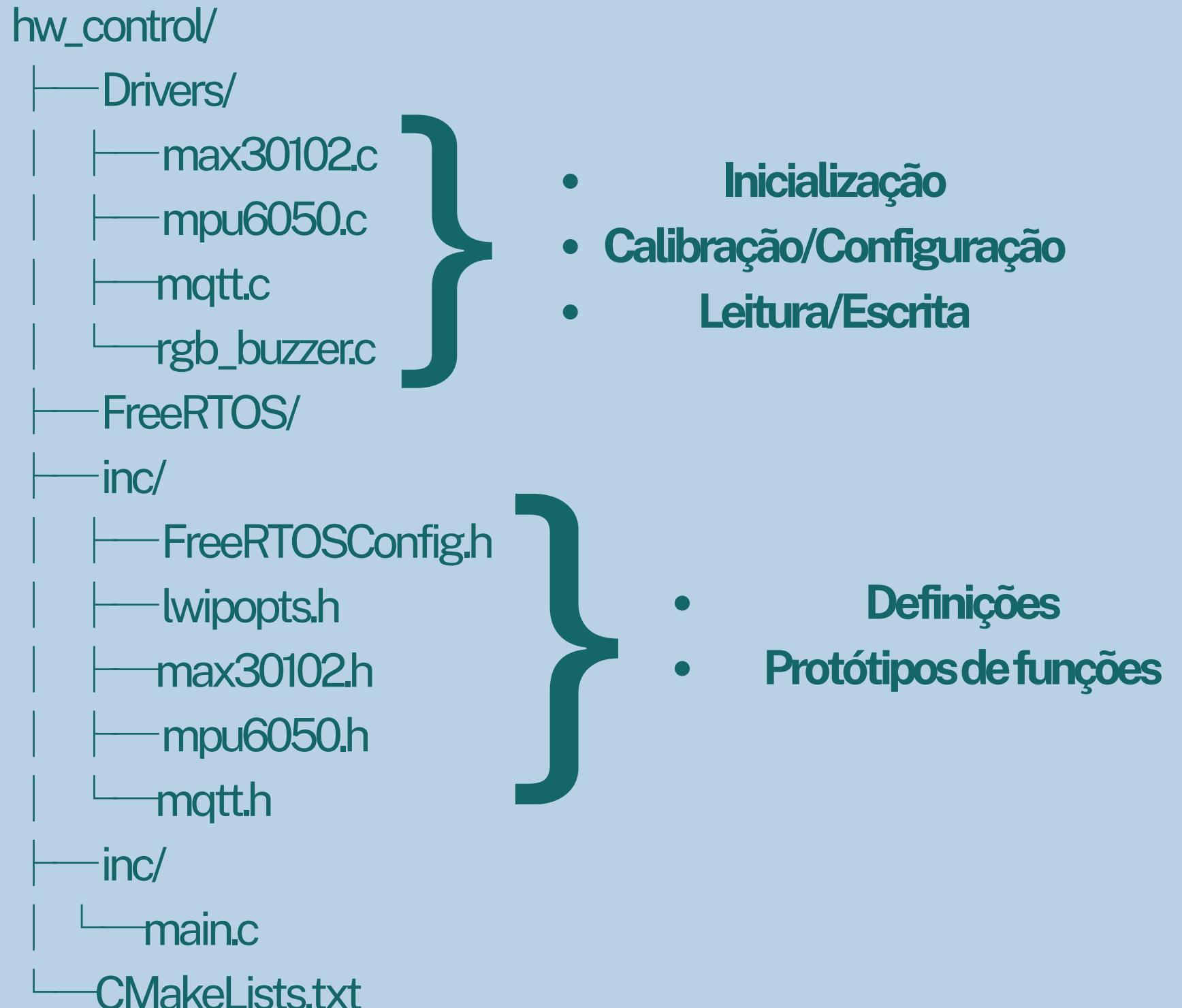
- Subscreve tópicos MQTT.
- Converte mensagens JSON → CSV.
- Cria arquivos diários com cabeçalho automático.
- Mantém histórico organizado (ex.: `vitals_2025-09-09.csv`).

Dashboard Integração dos dados em tempo real.

Feito usando PySide6, com parte gráfica desenvolvida no QtCreator e configurações/atualizações usando python. Possui 3 telas responsivas a um arquivo de dados no formato .csv, além de uma tela de histórico filtrável. O .csv é atualizado usando a biblioteca python “paho-mqtt”, que se comunica com o publisher MQTT, assina os tópicos e obtém os dados.



Estrutura de código



task emergency

- verifica botão
- publica mqtt

task fall

- leitura
- verificação do algoritmo
- resposta conforme usuário
- publica mqtt

task heart

- leitura média
- cálculo de bpm e spo2
- publica mqtt

task mqtt

- conecta MQTT
- verifica conexão

task water:

- verifica tempo
- aciona buzzer e led

Estrutura de código



mqtt.py

- conecta com mqtt (paho-mqtt)
- assina tópicos e extrai conteúdo
- salva em dados.csv

mainwindow.py

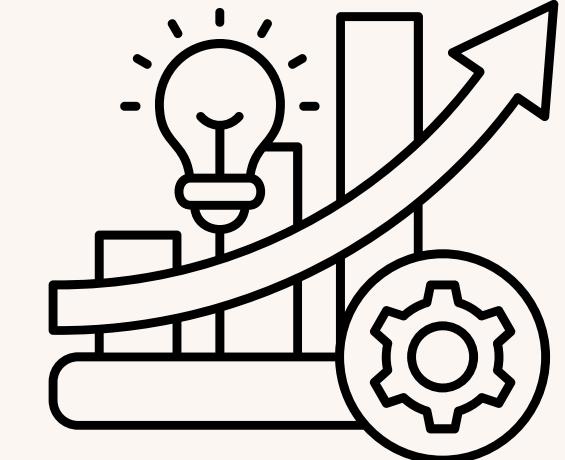
- leitura de dados.csv
- alteração de telas conforme dados

dados.csv

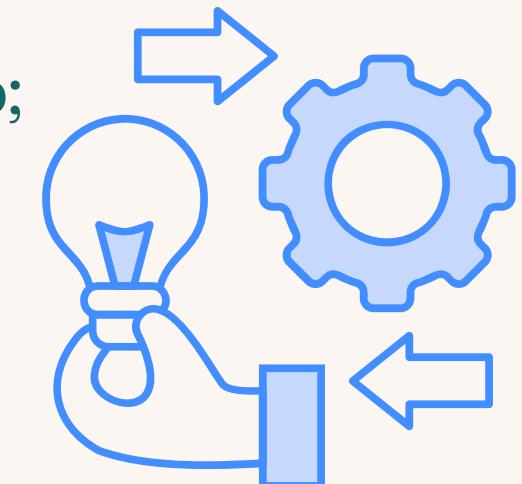
batimentos, oxigenação, horário, queda, quedas, emergencia

Próximos Passos e Evoluções

Com foco em melhorias contínuas e alta escalabilidade, projetamos as seguintes evoluções:



- Comunicação direta com o paciente por microfone e alto-falante (base);
- Banco de dados relacional para centralizar históricos e ajustar limiares personalizados (idade e perfil clínico);
- Integração de contato de emergência ao banco de dados, permitindo acionamento direto em caso de anomalias ou emergências (além da unidade móvel de resgate);
- Controle de medicamentos pelo dashboard, com atualização de firmware via OTA;
- Aplicativo móvel para acompanhamento familiar e médico;
- Localização do paciente (RSSI / Wi-Fi com API Google Geolocation);
- Vibracall para alertas táteis de lembrete de medicação;
- Arquitetura final minimizada, tornando o protótipo um sistema realmente vestível no pulso;
- Integração com bateria recarregável, permitindo autonomia da pulseira;
- Monitoramento do nível de bateria, com alertas de baixa carga;





Monitoramento Residencial

Obrigado! Dúvidas?

Felipe Leme
Vitor Gomes

