



INSTITUTO HARDAWARE BR – EMBARCATECH

Monitoramento e Assistência Homecare para pacientes idosos e crônicos

Aluno:

Felipe Leme Correa da Silva
Vitor Gomes de Souza

Campinas

2025
SUMÁRIO

1. Introdução	Página 2
2. Arquitetura do Hardware	Página 3
3. Arquitetura do Software	Página 4

1. Introdução

Nosso sistema consiste na leitura de sensores de batimentos cardíacos e oxigenação junto com acelerômetro para identificar quedas, apresentando uma solução para monitoramento de saúde residencial baseando-se na simplicidade e assertividade necessária na área médica.

2. Arquitetura do Hardware

Utilizando a BitDogLab, os sensores externos e o microcontrolador ESP32, elaboramos a arquitetura do sistema de forma modular, definindo como cada componente interage e evidenciando de maneira clara suas interconexões e processos.

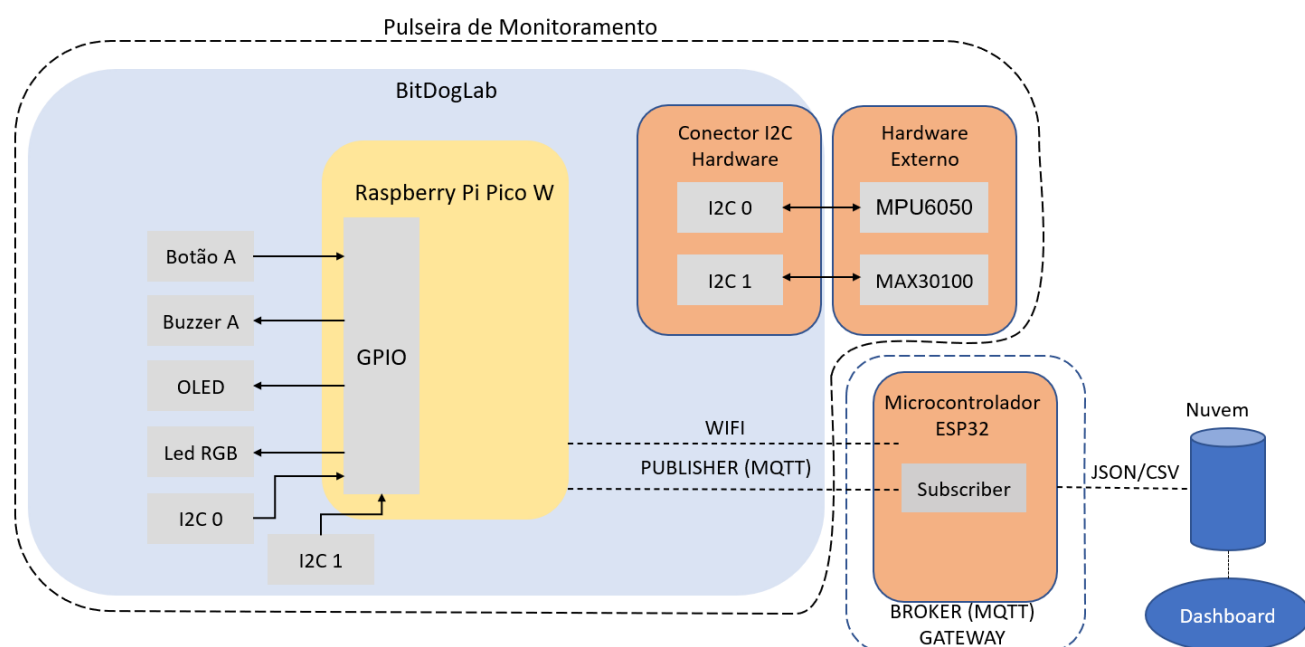


Foto 01: Arquitetura de Hardware

O esquema acima aborda os seguintes tópicos:

1. Pulseira de Monitoramento (BitDogLab + Hardware):

- Botão A: Permite interação manual do usuário, como cancelar alarmes de queda.
- Buzzer A: Emite alertas sonoros em caso de anomalias ou quedas detectadas.

- OLED: Mostra informações ao usuário, como BPM, SpO₂ e alertas.
- LED RGB: Fornece feedback visual (ex.: vermelho = alerta).
- GPIO: Faz a interface entre os periféricos internos (botão, buzzer, OLED, LED RGB).
- I2C 0 e I2C 1: Canais de comunicação para os sensores externos.

2. Sensores Externos (via I2C):

- MPU6050: Detecta movimentos bruscos ou quedas usando acelerômetro e giroscópio.
- MAX30100: Mede frequência cardíaca (BPM) e saturação de oxigênio (SpO₂).
- Conector I2C Hardware: Faz a ponte entre a BitDogLab e os sensores.

3. Comunicação via Wi-Fi e MQTT:

- A BitDogLab atua como Publisher MQTT, enviando leituras de BPM, SpO₂ e eventos de queda para o ESP32.
- Essa comunicação ocorre pela rede Wi-Fi local, com mensagens em tópicos específicos (ex.: homecare/pulseira/bpm).

4. Microcontrolador ESP32:

- Broker MQTT: Recebe mensagens publicadas pela BitDogLab.
- Subscriber: Armazena os dados em formatos JSON/CSV, gerando histórico para análise.
- Gateway: Atua como ponte, enviando os dados para a nuvem.

5. Nuvem e Dashboard:

- O ESP32 atua como gateway, enviando dados e arquivos estruturados para a nuvem.
- O Dashboard Médico acessa os dados da nuvem para mostrar informações em tempo real e relatórios históricos.

3. Arquitetura do Software

Para controlar este Hardware foi pensado em um Software esquematizado da seguinte maneira:

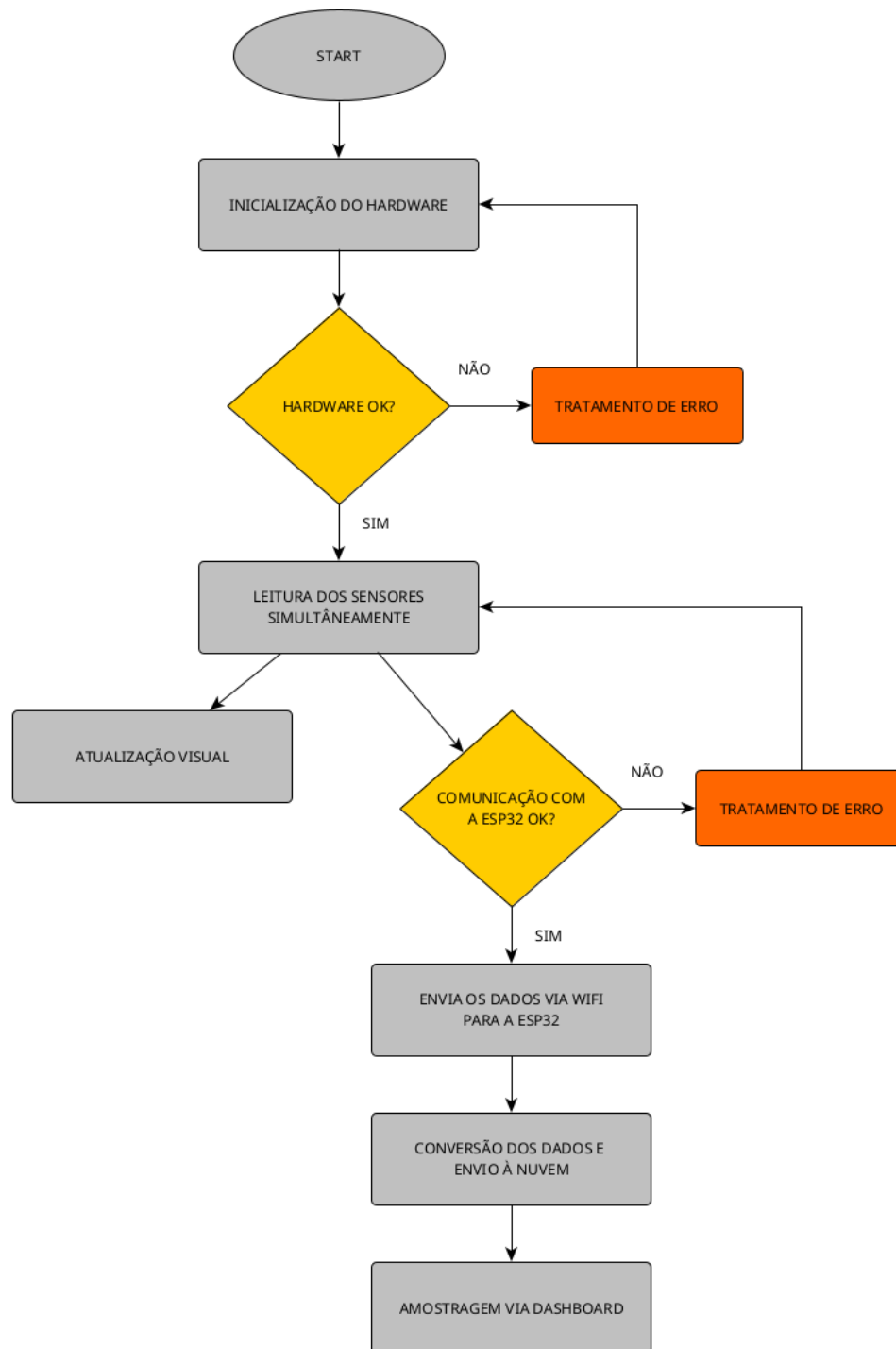


Foto 2 - Arquitetura de Software

No qual temos após o início a configuração desse hardware necessário para nossa aplicação, conteúdo inicialização e configuração dos componentes presentes na BitDogLab, como os I2C's, os LED's e o display OLED. Após isso verificamos qual o estado do hardware e o se está ok para começar a nossa aplicação, se não estiver tentamos novamente tratando o erro encontrado.

Após a validação do hardware, rodamos as principais funções da aplicação que são relacionadas aos dois sensores, obtendo suas respectivas leituras simultaneamente usando FreeRTOS para gerenciar associado ao resultado obtido, que será sinalizado usando o retorno visual presente na BitDogLab (LED's e OLED). Este mesmo resultado será transmitido via Wi-fi para a ESP32 caso a validação da comunicação com ela por parte da Raspberry Pi Pico W seja concluída com sucesso.

Na ESP32 teremos o tratamento prévia da informação dos sensores, retransmitindo-os para a nuvem, onde teremos conexão com um dashboard que será apresentado a um responsável. Neste dashboard está previsto as seguintes telas:



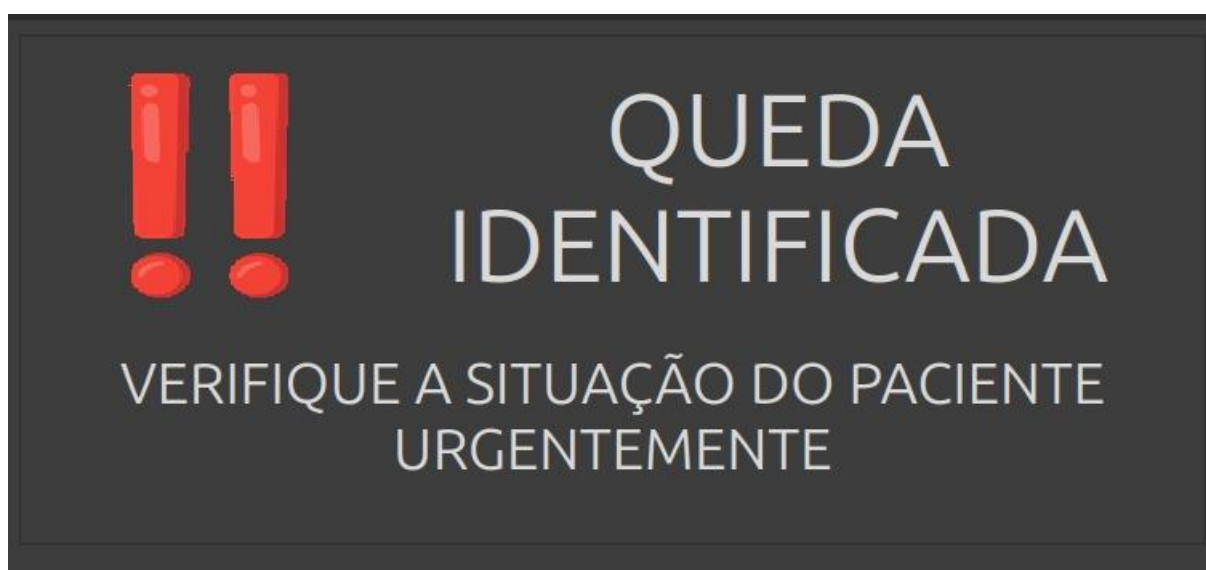
1 - Tela principal

Onde teremos dados importantes básicos.



2 - Tela de queda

Onde apareça o tempo restante para o paciente apertar o botão em caso de queda identificada informando que está bem.



3 - Tela de queda urgente

Na qual aparecerá caso ultrapasse o tempo previsto para o paciente apertar o botão informando que está bem após cair.

Além dessas ainda está prevista uma tela contendo o histórico de quedas e dos dados dos sensores.