

Projeto Cardiolo – Parte 1: Edge Computing e Armazenamento Local

Curso: Inteligência Artificial – 3º Semestre

FIAP – Faculdade de Informática e Administração Paulista

Capítulo 1 – Cardiolo Conectada: IoT e Visualização de Dados para Saúde Digital

Integrantes:

Luana Porto Pereira Gomes,

Luma Oliveira

Priscilla Oliveira

Paulo Bernardes

Relatório Técnico

Este relatório apresenta o desenvolvimento da Parte 1 do projeto CardioIA, aplicando Edge Computing para coleta e processamento local de dados simulados de saúde. O protótipo foi construído no ESP32 (Wokwi), com foco em resiliência: o dispositivo continua operando e registrando amostras mesmo sem conexão com a nuvem, sincronizando os dados automaticamente quando volta ao modo online.

Sensores e Componentes Utilizados

- DHT22 (Temperatura e Umidade): escolhido pela integração simples e boa precisão em aplicações de IoT. No contexto do CardioIA, representa uma medição ambiental próxima ao paciente.
- MPU6050 (Acelerômetro + Giroscópio): utilizado para detectar movimento e atividade. Permite identificar deslocamentos e vibrações, úteis para correlacionar esforço físico com variações de BPM.
- Botão (batimentos): usado para simular batidas do coração. O sistema conta os cliques por janela de 10 s e converte para BPM (pulsos \times 6).
- LED de Alerta: indicação visual local. Acende quando temperatura $> 38^{\circ}\text{C}$ ou BPM > 120 .
- Módulo SD (armazenamento local): atua como buffer offline. Enquanto offline, cada amostra é gravada em arquivo; ao voltar online, os dados são enviados e o buffer é limpo.
- Chave deslizante (ONLINE/OFFLINE): usada para simular conectividade. Em OFF, o sistema opera em modo Edge, armazenando localmente no SD. Em ON, realiza a sincronização automática das amostras e limpa o buffer após o envio.

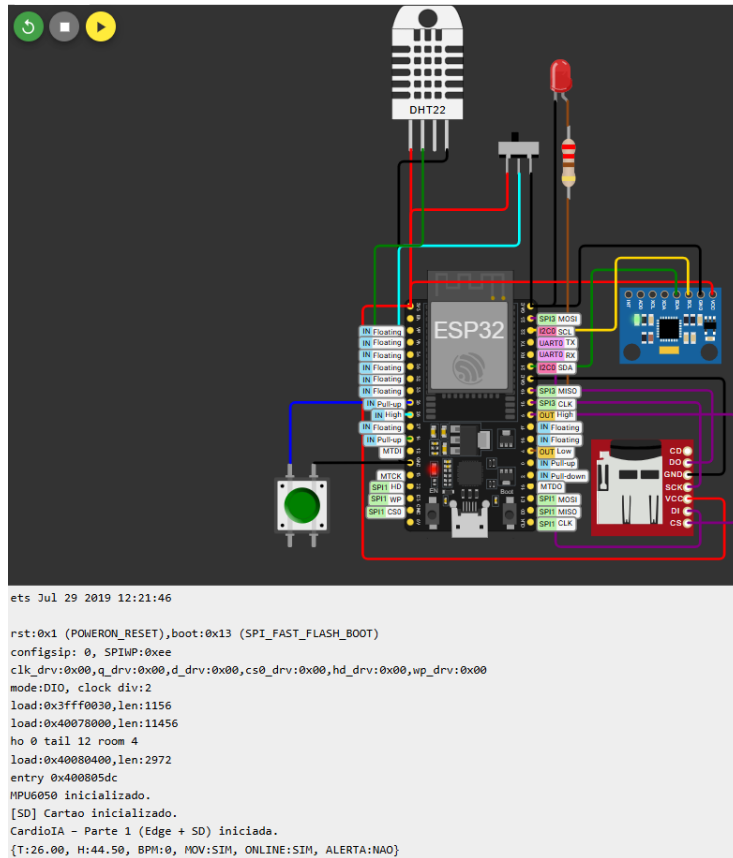
Justificativa Técnica e Desafios

A orientação inicial previa o uso do SPIFFS/LittleFS para o armazenamento em flash interna. Entretanto, no Wokwi essas partições não são emuladas de forma completa, gerando falhas de montagem e impedindo a escrita real. Por isso, optamos pelo cartão SD, que o simulador suporta estável. Essa troca manteve o objetivo pedagógico: provar resiliência offline (edge-first) e sincronização quando online.

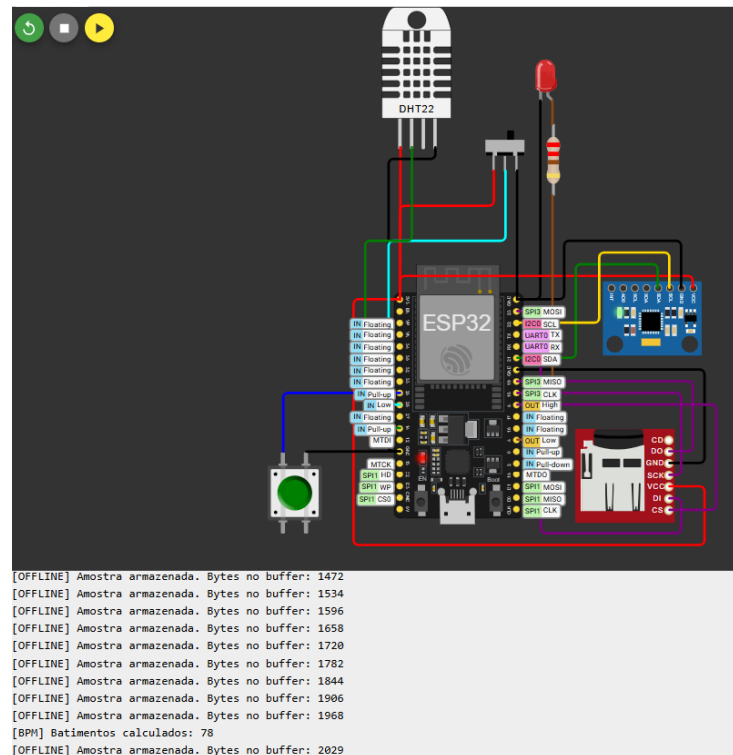
Em hardware físico (ESP32 real), o mesmo fluxo funcionaria com SPIFFS/LittleFS. Na simulação, o SD cumpriu o papel de armazenamento local com confiabilidade, permitindo evidenciar o ciclo captura \rightarrow armazenamento \rightarrow sincronização \rightarrow limpeza do buffer.

Resultados e Testes

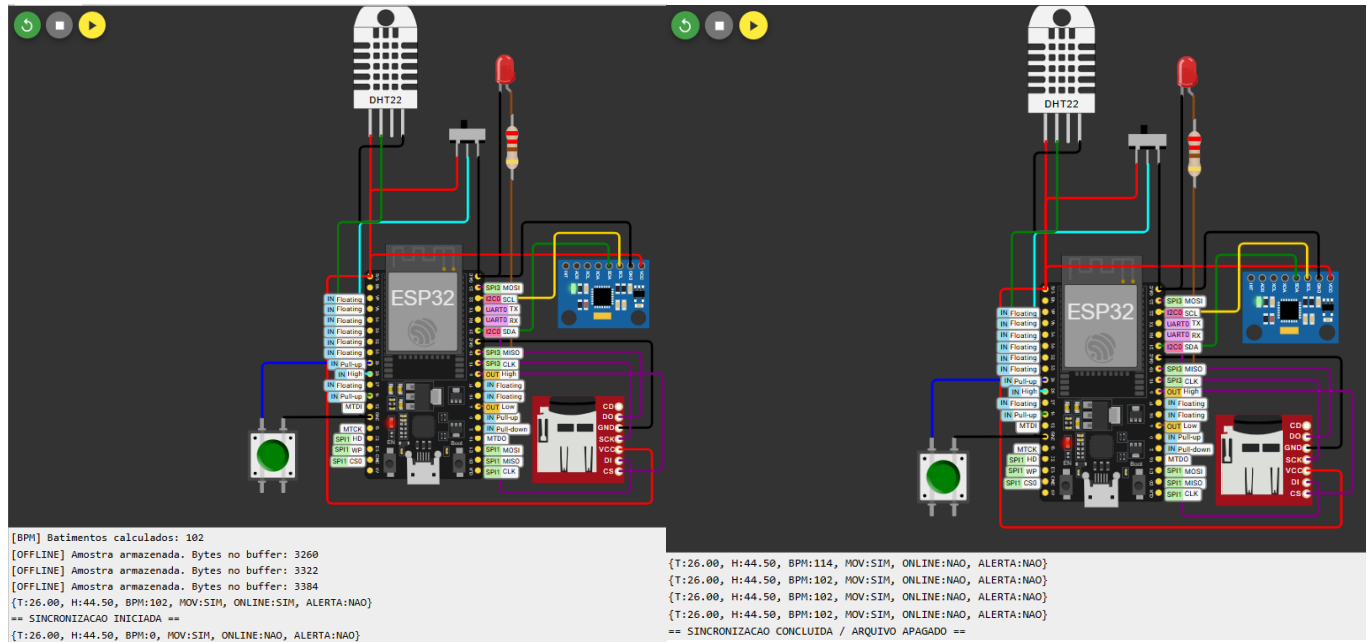
Foram realizados quatro testes principais para validar o funcionamento do sistema. As imagens e saídas de terminal de cada etapa estão descritas abaixo:



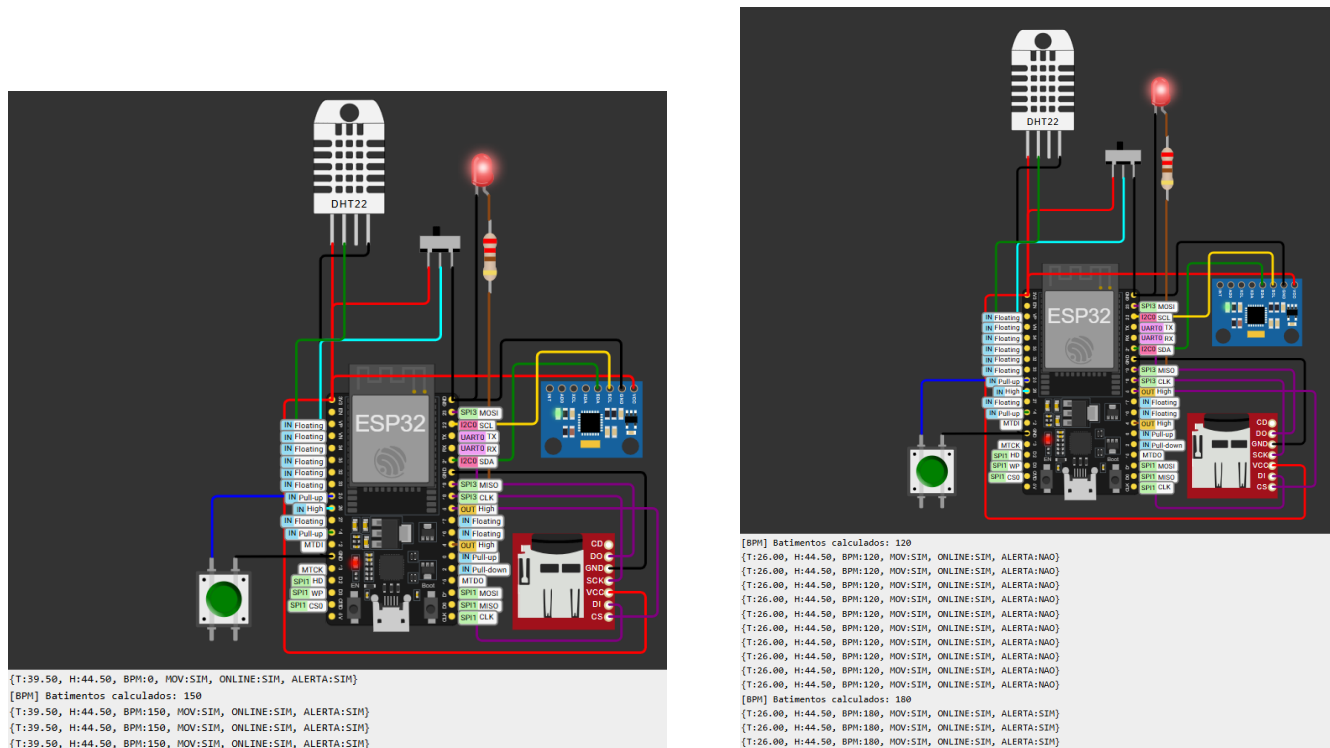
- Teste 1 – Inicialização do sistema: demonstra o funcionamento inicial, com a leitura dos sensores DHT22 e MPU6050 e a preparação do buffer local no cartão SD.



- Teste 2 – Operação Offline e Resiliência Local: ostras gravadas no SD com aumento do buffer.



Teste 3 – Detecção de movimento e sincronização Online: simula o retorno da conectividade, realizando a sincronização automática dos dados armazenados e apagando o buffer após o envio.



- Teste 4 – Sistema de alerta ativo: demonstra a resposta visual (LED aceso) e os alertas no terminal para condições de temperatura e batimentos cardíacos acima do limite definido.

Conclusão

O protótipo comprovou o objetivo da Parte 1: operar de forma contínua mesmo sem rede, armazenando dados localmente e sincronizando assim que possível. A troca do SPIFFS/LittleFS pelo SD foi uma decisão técnica motivada pelas limitações do simulador, sem prejuízo aos conceitos de Edge Computing e saúde digital. O aprendizado obtido reforça a importância de arquiteturas resilientes no ciclo IoT–Edge–Cloud.