## **Relatório – Projeto CardioIA**

### *(Fase 3 – Parte 1)*

### **1. Introdução e Objetivo**

### Este relatório apresenta o desenvolvimento da Parte 1 da Fase 3 do projeto CardioIA, cujo foco foi a criação de um protótipo IoT de monitoramento de saúde. O objetivo principal é **simular um sistema embarcado inteligente**, capaz de coletar e interpretar sinais vitais, garantindo a resiliência dos dados mesmo em situações de falha de conexão com a internet.

O projeto adota o conceito de Edge Computing, onde o ESP32 realiza o processamento e o armazenamento local dos dados coletados, evitando a perda de informações e assegurando a continuidade do monitoramento.

### **2. Componentes e Tecnologias Utilizadas**

A simulação foi desenvolvida na plataforma Wokwi, utilizando os seguintes componentes e recursos:

* **Microcontrolador ESP32**: núcleo central do sistema, responsável por processar leituras, armazenar dados e gerenciar conexões.
* **Sensor DHT22**: coleta dados de temperatura e umidade, simulando sinais vitais e condições ambientais.
* **Potenciômetro**: usado como simulador de batimentos cardíacos (BPM), permitindo variar o valor manualmente.
* **LED RGB (ânodo comum)**: atua como indicador de status do paciente, mostrando o estado de saúde em três cores (Normal, Atenção e Crítico).
* **SPIFFS (Serial Peripheral Interface Flash File System)**: sistema de arquivos utilizado para o armazenamento local dos dados, garantindo a persistência das informações mesmo sem conexão.
* **Monitor Serial**: representa a “nuvem” na simulação, exibindo os dados coletados e enviados.

### **3. Fluxo de Funcionamento do Sistema**

O sistema opera de forma cíclica e contínua, seguindo o seguinte fluxo lógico:

1. Coleta de Dados:A cada 5 segundos, o ESP32 lê os valores dos sensores:  
   * Temperatura (°C) e Umidade (%) via DHT22
   * Batimentos Cardíacos (BPM) via Potenciômetro
2. Análise e Classificação:Os dados são analisados pela função atualizarLED(), que classifica o estado de saúde do paciente conforme as faixas de referência definidas:

| **Estado** | **Cor do LED** | **Descrição** | **Faixa de Valores** |
| --- | --- | --- | --- |
| 🟢 **Normal** | Verde | Parâmetros dentro da faixa saudável; sem risco. | 36.5–37.5°C 60–100 BPM 30–60% Umidade |
| 🟡 **Atenção** | Amarelo | Valores fora da faixa ideal, mas ainda estáveis. | 37.6–38.5°C 101–120 BPM <30% ou >60% |
| 🔴 **Crítico** | Vermelho | Risco à saúde – requer intervenção imediata. | >38.5°C  >120 BPM |

Essa leitura contínua e visual permite detectar rapidamente variações anormais nos parâmetros vitais.

1. Verificação de Conectividade: O sistema verifica uma variável chamada conectado, que representa o estado da rede Wi-Fi (simulada).  
   * conectado = true → dispositivo online
   * conectado = false → dispositivo offline
2. Tomada de Decisão (Online vs Offline):
   * Modo Offline: os dados são armazenados localmente em um arquivo no SPIFFS (/dados.txt), garantindo que nenhuma leitura seja perdida.
   * Modo Online: os dados armazenados são “enviados” (impressos no Monitor Serial), e o arquivo local é apagado após a sincronização.
3. Repetição do Ciclo:Após cada ciclo de leitura e decisão, o sistema aguarda alguns segundos e recomeça o processo, garantindo monitoramento contínuo e autônomo.

### **4. Lógica de Resiliência e Edge Computing**

A resiliência do sistema é o ponto central deste protótipo.  
Em contextos de saúde, falhas de rede podem comprometer a segurança e a continuidade do monitoramento de pacientes.  
Por isso, o CardioIA utiliza uma arquitetura de Edge Computing, com duas camadas principais:

* Modo Offline (Edge Storage):  
  O ESP32 armazena localmente os dados coletados quando não há conexão, criando um histórico no arquivo /dados.txt.  
  Essa funcionalidade assegura que nenhuma medição seja perdida, mesmo que a transmissão para a nuvem esteja indisponível.
* Modo Online (Sync & Clean):  
  Assim que a conexão é restabelecida, o sistema envia os dados armazenados para o servidor (simulado via serial) e limpa o arquivo, evitando duplicidade e liberando espaço.  
  Esse processo de armazenar, transmitir e limpar reflete uma estrutura real de sincronização inteligente, amplamente utilizada em sistemas IoT de missão crítica.

### **5. Estratégia de Armazenamento e Continuidade**

O sistema de arquivos SPIFFS tem espaço limitado, portanto, em um cenário real, seria implementado um controle FIFO (First In, First Out), que:

* Apaga os registros mais antigos quando a memória atinge 90% da capacidade;
* Mantém apenas os dados mais recentes e relevantes para o diagnóstico.

Isso garante que o dispositivo nunca interrompa sua operação, mantendo a coleta contínua mesmo em longos períodos sem rede.

### **6. Resultados Obtidos**

Durante os testes no Wokwi, o sistema demonstrou:

* Correta identificação visual dos estados clínicos (LED RGB mudando de cor conforme as leituras).
* Gravação e leitura confiável dos dados no SPIFFS.
* Simulação bem-sucedida de resiliência, com alternância entre modo online e offline.

Esses resultados comprovam o funcionamento estável do protótipo e sua capacidade de processar e preservar dados localmente, característica essencial em aplicações médicas reais.

### **7. Conclusão**

O sistema desenvolvido ilustra como dispositivos IoT podem operar de forma autônoma, resiliente e segura, mesmo diante de falhas de rede.

A partir desta base, as próximas etapas do projeto poderão integrar o envio real dos dados para a nuvem, banco de dados e algoritmos de IA para análise preditiva de sinais vitais.