

Smart Farm



Objetivos do documento

Adicionar a proposta de arquitetura do sistema Smart Farm com: diagrama de hardware, blocos funcionais, fluxograma do software, lógica de atuação e recomendações de implementação.

Descrição do problema

A agricultura enfrenta desafios cada vez maiores para manter a produtividade, eficiência e sustentabilidade frente às variações climáticas, escassez de recursos hídricos, aumento de custos operacionais e necessidade de decisões rápidas e precisas. A falta de monitoramento contínuo e em tempo real das condições ambientais no campo limita a capacidade de resposta imediata às mudanças que afetam diretamente a lavoura, como umidade do solo, temperatura, luminosidade e pressão atmosférica. Isso pode resultar em desperdício de recursos, baixa produtividade e prejuízos econômicos.

Requisitos Funcionais

- **Coleta de Dados Ambientais:** Monitorar variáveis como temperatura, umidade, luminosidade, cor das plantas e radiação UV.
- **Atuação automática:** Acionamento automático de atuadores (como bombas de irrigação e ventiladores) com base nos parâmetros coletados.
- **Conectividade remota:** Transmissão dos dados coletados para um sistema central via redes sem fio (LoRa, Zigbee ou BLE).
- **Processamento local:** Capacidade de pré-processar os dados na borda (Raspberry Pi ou ESP32), reduzindo o volume de dados enviados.
- **Armazenamento híbrido:** Armazenamento local (backup) e em nuvem para segurança e análise posterior.
- **Interface de visualização:** Acesso remoto aos dados em tempo real por meio de interface Web ou Mobile.
- **Modularidade de hardware:** Suporte a diferentes sensores, com detecção automática de novos dispositivos.
- **Análise preditiva:** Aplicação de algoritmos para detecção de padrões e apoio na tomada de decisões agrícolas.

Requisitos Não Funcionais

- **Baixo Consumo de Energia:** Fundamental para operação em áreas remotas e custo total de produção.
- **Alta Confiabilidade:** O sistema deve funcionar de forma robusta mesmo em ambientes adversos.
- **Escalabilidade:** Permitir a expansão do número de sensores e atuadores conforme necessário.
- **Usabilidade:** Interface intuitiva para facilitar o uso por operadores agrícolas.
- **Resiliência à falhas:** Armazenamento local em caso de perda de conexão com a nuvem.

Lista de componentes

Sensores

- GY-33 – Sensor de cor (não vamos mais usar/sensor muito pontual)
- AHT10 – Sensor de temperatura e umidade
- BH1750 – Sensor de iluminação

Microcontroladores e Processadores

- RP2040 – Microcontrolador principal para aquisição e controle
- ESP32 – Alternativa como elemento de borda e conectividade

Comunicação

- Módulo LoRa – Comunicação de longa distância e baixo consumo
- MQTT

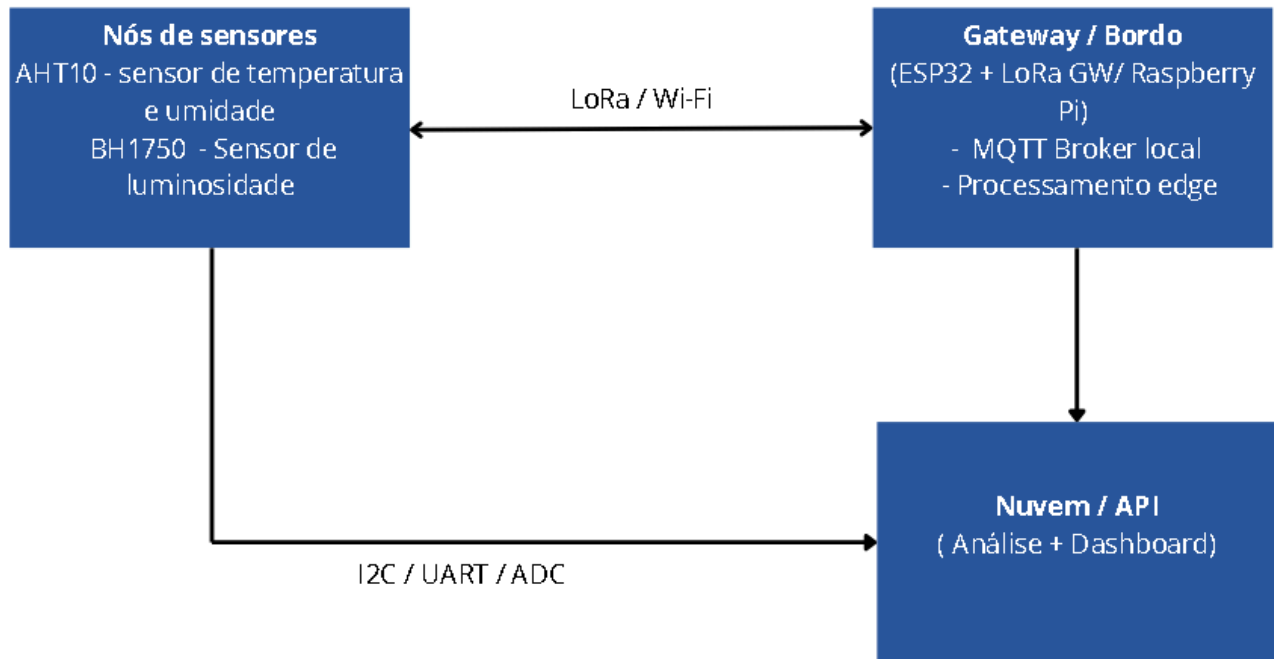
Outros Componentes

- Fonte de alimentação (bateria ou painel solar)
- Placas de desenvolvimento (BitDogLab)
- Cabos e conectores
- Extensor I2C
- Atuadores (bombas de irrigação, ventiladores)

Visão geral do sistema

O sistema Smart Farm realiza monitoramento contínuo de variáveis ambientais (temperatura, umidade e luminosidade), realiza pré-processamento na borda, aciona atuadores automaticamente (bombas, ventiladores) e envia dados para nuvem via redes sem fio de baixo consumo (LoRa / MQTT). Deve ser modular, escalável e com armazenamento híbrido (local + nuvem).

Diagrama de Hardware (blocos)



Atuadores locais (bombas, ventiladores) conectados ao controlador

Blocos funcionais (detalhe)

1. Camada de Sensoriamento (Edge Nodes)

- Função: leitura periódica de sensores (T, U, e Lux), filtragem e amostragem.
- Hardware típico: ESP32/ RP2040 + módulos sensores via I2C/ADC.
- Saída: pacotes JSON reduzidos com métricas e timestamp.

2. Comunicação de Campo

- Função: enlace de baixa potência e longo alcance (LoRa) ou curto alcance (BLE/Zigbee) dependendo da topologia.
- Protocolo: LoRaWAN simples ou LoRa P2P + MQTT sobre gateway.

3. Gateway / Unidade de Borda

- Função: receber dados de múltiplos nós, agregar, pré-processar (filtragem, compressão, detecção de anomalias), armazenar localmente e encaminhar à nuvem.
- Hardware: ESP32 + módulo LoRa ou Raspberry Pi (se processamento mais pesado/pacotes MQTT).
- Serviços: Broker MQTT local, buffer em disco, regras de atuação imediata.

4. Nuvem / Backend

- Função: persistência de dados, análises avançadas, dashboards, APIs para mobile/web.

5. Módulo de Atuação

- Função: lógica de controle local e remoto; acionamento de bombas/ventiladores via relé ou drivers sólidos.

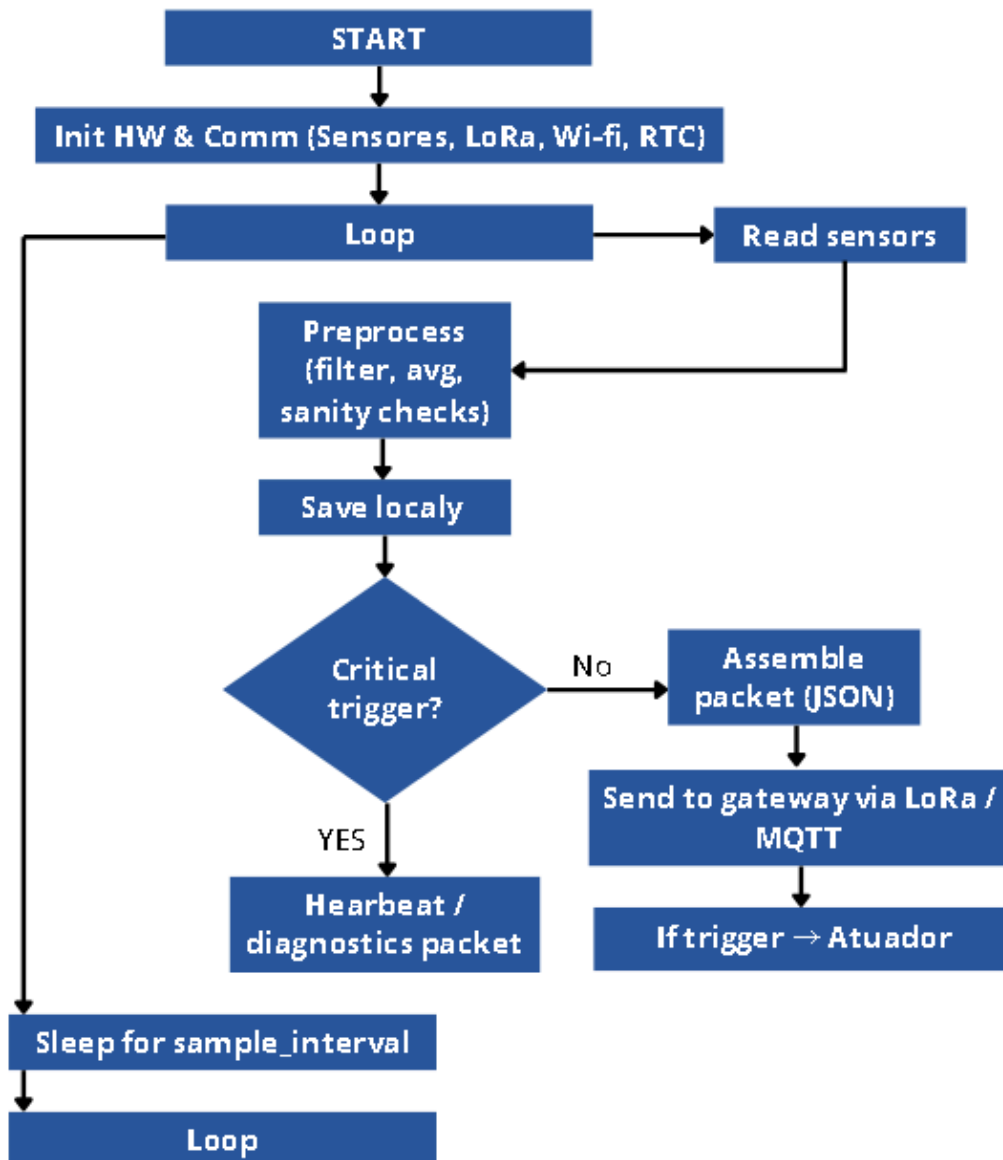
6. Interface de Visualização

- Web e Mobile: dashboards em tempo real, histórico, alarmes, configurações de thresholds.

7. Módulo Analítico (ML / Preditivo)

- Função: algoritmos de detecção de padrões (irrigação preditiva), rodando preferencialmente na nuvem; modelos leves podem rodar no gateway.

Fluxograma do Software (nível alto)



Notas de software:

- Amostragem adaptativa: intervalo de amostragem pode variar conforme hora do dia, nível de radiação, ou estado da bateria.
- Conteúdo dos pacotes: node_id, timestamp, temperature, humidity e lux.
- Segurança: TLS/DTLS sobre MQTT quando possível; chaves pré-provisionadas.

Lógica de atuação (exemplo)

- **Regra 1 (Irrigação automática):** Se $umidade_solo < U_min$ por t_consec minutos -> ativar bomba por $t_irrigacao$ ou até $umidade_solo \geq U_target$.
- **Regra 2 (Proteção elétrica):** Se $corrente > I_max$ ou $voltagem < V_min$ -> desligar atuadores e enviar alarme.

As regras podem ser implementadas localmente (gateway) para resposta imediata e na nuvem para decisões complexas e otimização.