

Projeto COMP0417 - Especificação

Título: Portable Inspection Edge Node. Nó Embarcado Portátil para Vistorias com Processamento de Borda e Integração com Backend

Integrantes: Dayvid Santana da Silva e Enzo Guilherme Dias Marques

Disciplina: Fundamentos de Sistemas Embarcados (COMP0417)

Data de submissão: 01/02/2026

Resumo executivo

Este projeto propõe o desenvolvimento de um nó embarcado portátil baseado em ESP32-CAM para apoio a vistorias técnicas. O dispositivo captura imagens e leituras de sensores (IMU, distância e luminosidade), executa processamento de borda leve no próprio dispositivo (edge computing) para extração de métricas visuais explicáveis (ex.: densidade de bordas associadas a fissuras), persiste dados localmente (microSD e flash) e sincroniza os resultados com o backend por meio de Wi-Fi (HTTPS) ou comunicação serial (USB) como mecanismo de contingência.

O processamento embarcado não realiza decisão pericial, mas sim pré-processamento e sumarização, reduzindo o volume de dados transmitidos e aumentando a rastreabilidade técnica. Cada captura gera um rascunho de laudo estruturado (JSON) acompanhado de thumbnail pronto para ingestão por endpoint REST do EngScan.

Justificativa

Vistorias técnicas em engenharia diagnóstica dependem, em geral, da coleta manual de imagens e posterior análise em escritório. A introdução de um nó embarcado com processamento de borda local permite:

- Redução do volume de dados transmitidos;
- Padronização da coleta;
- Inclusão automática de metadados técnicos;
- Geração de dados estruturados e auditáveis para sistemas de apoio à decisão.

Do ponto de vista acadêmico, o projeto explora de forma integrada sensores, processamento de sinais/imagens, comunicação e sistemas operacionais embarcados, alinhando-se diretamente aos objetivos da disciplina.

Requisitos do projeto

Requisitos funcionais

1. Captura de imagem via câmera OV2640.
2. Geração de thumbnail em resolução reduzida para envio.
3. Leitura de sensores:
 - IMU (aceleração e orientação);
 - Sensor de distância (VL53L0X ou HC-SR04);
 - Sensor de luminosidade (BH1750 ou LDR).
4. Processamento de borda embarcado, incluindo:
 - Conversão para escala de cinza;
 - Suavização leve (blur);
 - Detecção de bordas (Sobel ou Canny simplificado);
 - Extração de métricas (densidade de bordas, comprimento total, área estimada).
5. Geração de pacote de dados estruturado (JSON) com métricas e metadados.
6. Persistência de JSON + imagem/thumbnail em microSD.
7. Envio do pacote via Wi-Fi (HTTPS POST multipart) para a API do EngScan.
8. Manutenção de **fila local de envios pendentes** para operação offline.
9. Comunicação serial (USB) como fallback, com framing, CRC e confirmação.
10. Indicação de estado por OLED e LEDs; botão físico para captura manual.
11. Persistência de configurações (device_id, token) em flash interna.

Requisitos não funcionais

1. Tempo alvo de processamento local (pré-processamento + métricas) < 3 s por captura.
2. Robustez contra falhas de comunicação, com retry e backoff.
3. Segurança básica: autenticação por token e uso de HTTPS (modo de testes controlado permitido).
4. Eficiência energética: uso de delays, sleep parcial e controle de periféricos.
5. Código documentado e reproduzível, com repositório versionado.

Atendimento aos critérios da disciplina

- **Sensores e atuadores:** câmera, IMU, distância, luminosidade, LEDs, botões e display OLED.
- **Memória não volátil:** microSD para dados e flash para configurações.
- **Comunicação serial:** protocolo próprio com framing e CRC.
- **Processamento embarcado:** pipeline de detecção de bordas e métricas.
- **Arquitetura de software:** FreeRTOS com tarefas isoladas e filas.

- **Documentação:** especificação, pseudocódigo, testes e apresentação.
-

Arquitetura do sistema

```
[Operador]
| botões / OLED
▼
[ESP32-CAM Edge Node]
└ sensor_task
└ camera_task
└ process_task (borda + métricas)
└ storage_task
└ comms_task
└ watchdog
  |
  microSD — Wi-Fi / USB-Serial → EngScan Backend
```

Pipeline de processamento de borda

O processamento embarcado segue o fluxo:

```
Imagen capturada
↓
Conversão para grayscale
↓
Blur gaussiano leve (3x3)
↓
Detecção de bordas (Sobel ou Canny simplificado)
↓
Threshold
↓
Extração de métricas
```

Métricas calculadas:

- Densidade de bordas (edge_density);
- Comprimento total de bordas (pixels);
- Área estimada de regiões com alta concentração de bordas;
- Índice de confiança (heurístico).

Essas métricas **não substituem análise humana**, servindo apenas como pré-processamento e triagem.

Formato do pacote de dados

```
{  
    "device_id": "ES-0001",  
    "timestamp": "2026-02-10T14:35:20Z",  
    "metadata": {  
        "imu": {"ax": 0.01, "ay": -0.02, "az": 0.98},  
        "distance_mm": 325,  
        "light_lux": 120,  
        "thumbnail": "thumbnail.jpg"  
    },  
    "analysis": {  
        "edge_density": 0.23,  
        "estimated_area_cm2": 12.5,  
        "confidence": 0.74,  
        "algorithm": "sobel_v1"  
    }  
}
```

Protocolo serial

Framing byte-oriented com start byte, tamanho, payload e CRC-16-CCITT. Suporte a chunking e confirmação (ACK/NACK).

Backend mínimo

- **Endpoint:** POST /api/engscan/ingest-scan
- **Autenticação:** Bearer Token
- **Payload:** multipart/form-data (JSON + thumbnail)
- **Resposta:** 201 Created

Plano de testes e métricas

- Testes de sensores, SD, pipeline de processamento e comunicação.
- Métricas: latência, uso de heap, taxa de sucesso de upload e ocupação do SD.

Entregáveis

1. Documento de especificação.
 2. Código-fonte versionado.
 3. Protótipo montado e documentado.
 4. Vídeo demonstrativo.
 5. Relatório técnico e slides.
-

Observação final: esta especificação enfatiza o uso de processamento de borda embarcado como aplicação prática e didática de sistemas embarcados, mantendo a decisão técnica final no backend, em conformidade com princípios de rastreabilidade e explicabilidade.