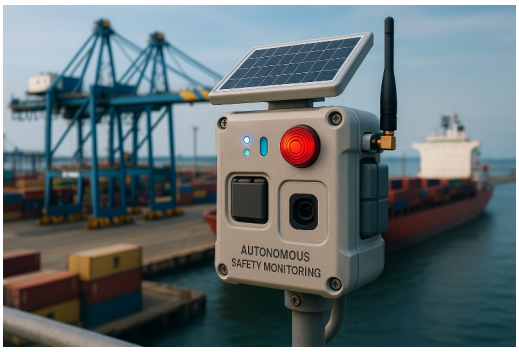


Smart EPI: Robô Autônomo para Monitoramento Inteligente de EPIs

A segurança no ambiente de trabalho, especialmente em setores industriais, é uma prioridade inegociável. O uso correto e consistente de Equipamentos de Proteção Individual (EPIs) é fundamental para mitigar riscos de acidentes e proteger a saúde dos trabalhadores. No entanto, a supervisão manual e a garantia da utilização adequada dos EPIs podem ser desafiadoras, demandando recursos significativos e estando sujeitas a falhas humanas.

A presente proposta aborda essa problemática através da implementação de um Sistema de Monitoramento de EPI Embarcado e Autônomo, utilizando a plataforma Raspberry Pi como unidade central de processamento. A justificativa para esta solução reside na necessidade de automatizar e otimizar o processo de verificação do uso de EPIs, proporcionando um ambiente de trabalho mais seguro, eficiente e em conformidade com as normas regulamentadoras.

A implementação de um sistema autônomo de detecção de EPIs oferece diversos benefícios, incluindo:

- **Aumento da Segurança:** Detecção imediata de não conformidades, permitindo ações corretivas rápidas e prevenindo acidentes.
- **Otimização de Recursos:** Redução da necessidade de supervisão manual dedicada ao controle de EPIs, liberando recursos humanos para outras tarefas.
- **Melhora da Conformidade:** Garantia de que os trabalhadores estejam utilizando os EPIs necessários, auxiliando no cumprimento de normas e regulamentações.
- **Registro e Análise de Dados:** Possibilidade de coletar dados sobre o uso de EPIs ao longo do tempo, permitindo identificar padrões, áreas de maior risco e a eficácia das políticas de segurança.
- **Alertas em Tempo Real:** Notificação imediata de não conformidades para supervisores, possibilitando intervenções proativas.

1.1. Objetivos

O objetivo deste sistema é automatizar a detecção do uso adequado de EPIs em ambientes industriais, visando aumentar a segurança, otimizar recursos e garantir a conformidade com as normas regulamentadoras.

- Desenvolver um sistema capaz de detectar, em tempo real, a presença ou ausência de EPIs específicos em imagens capturadas por câmeras.
- Implementar um sistema de alertas local (visual e/ou sonoro) em caso de não conformidade.
- Estabelecer um sistema de notificação remota para supervisores através do protocolo MQTT.
- Criar um mecanismo para registro local de eventos de detecção.
- Projetar um sistema robusto e adequado para operação em ambientes industriais.
-

1. ARQUITETURA

1.1 HARDWARE ESSENCIAL

Computador Embarcado: Raspberry Pi 4B (8GB)

Escolhemos a Raspberry Pi 4B (8GB) como unidade de processamento central ela oferece um desempenho com base na necessidade de desempenho, para bibliotecas de IA e comunicação sem sobrecarregar o hardware, com consumo energético otimizado — exatamente o que o projeto exige.

Câmera Raspberry Pi HQ + lente 6mm

Incluímos uma câmera Pi Camera pois oferece integração direta e alta qualidade de imagem, compatível com Raspberry Pi, ideal para detecção precisa em ambientes industriais.

Sensor de Movimento PIR HC-SR501

Escolhemos o sensor PIR HC-SR501 porque ele é versátil, de baixo custo e excelente para detectar ativar a câmera por movimento, economiza energia e ativa a detecção apenas quando há movimento humano detectado.

Sensor de Iluminação BH1750

Inserimos o sensor de Iluminação BH1750 para ajuste dinâmico de exposição da câmera. Possui baixo consumo, fácil integração via I2C, ajuda a adaptar a câmera à luz ambiente.

☀ Painel Solar 20W + Controlador PWM

Adotaremos um painel solar de 20W para fornecer energia autônoma suficiente ao sistema, permitindo operação 24 horas por dia, 7 dias por semana, sem a necessidade de uma fonte de energia externa. O controlador PWM regula o carregamento da bateria para prolongar sua vida útil.

🔋 Bateria: LiFePO4 12V 6Ah (com BMS)

Optamos por utilizar uma bateria $LiFePO_4$ (fosfato de ferro-lítio) é indicado para o uso em diversos aparelhos como fitas de led, câmeras, placas solares, entre outros. Diferente de outras químicas de lítio, a $LiFePO_4$ tem altíssima resistência térmica e química — ela não entra em combustão espontânea, mesmo sob aquecimento, perfuração ou impacto, o que é essencial em áreas portuárias com ambientes de riscos .

🔁 Modulo de comunicação ESP32 + LoRa SX1276

Implementaremos um sistema comunicação usando LoRa que utiliza Biblioteca usada no ESP32 para enviar mensagens MQTT-like para o gateway remoto. A tecnologia tem seu papel estratégico no funcionamento do sistema, comunicação de longo alcance (alertas), comunicação redundante mesmo em locais sem Wi-Fi; baixo consumo de energia e boa cobertura.

⚡ Conversor de Tensão Buck 12V -> 5V

Para redução a tensão da bateria (12V) para a tensão necessária para alimentar o Raspberry Pi 5 (5V).

🔘 Alerta Visual Buzzer + LED RGB 12V (vermelho ou azul)

Incorporaremos Buzzer + LED RGB 12V para Alerta visual e sonoro para Feedback direto ao(s) operador(es) local. É ativado o buzzer e LED quando um EPI é detectado incorretamente ou na ausência de EPI. É visível mesmo sob neblina, fumaça ou ambientes com baixa luminosidade.

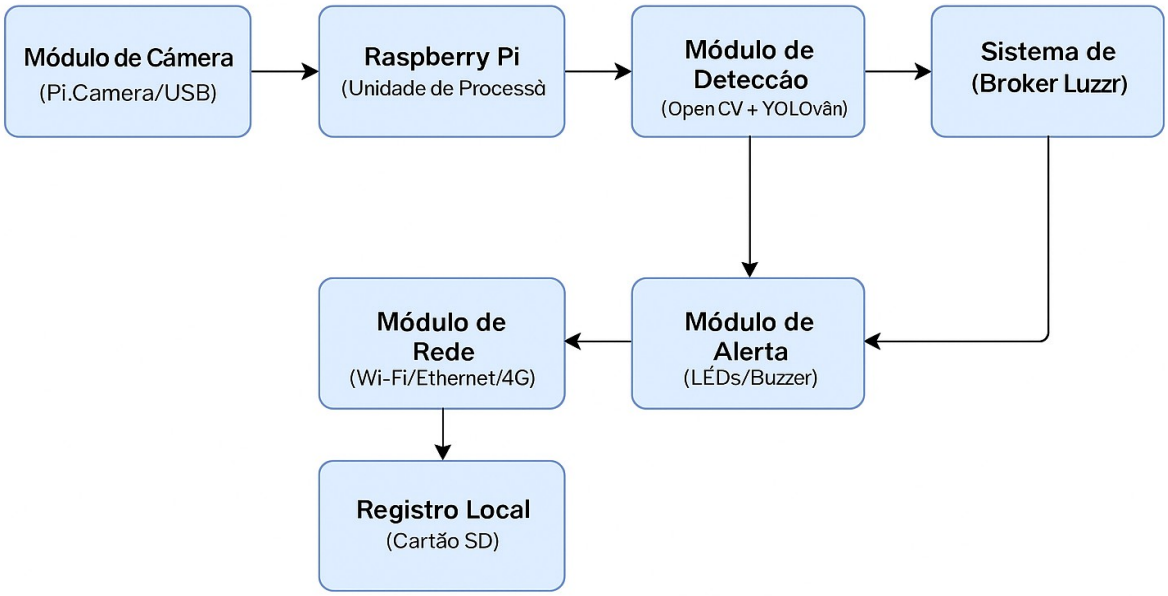
Cartão microSD 32GB Classe 10

Usado para armazenar o sistema operacional, o software, os modelos de IA e os dados coletados. A Classe 10 garante uma velocidade de leitura/escrita adequada para o desempenho do sistema.

Caixa Hermética de Policarbonato IP65

Escolhemos uma caixa de policarbonato IP65 do tipo compacta, para proteger os componentes eletrônicos contra as intempéries (chuva, poeira, umidade, maresia), garantindo a durabilidade do sistema em ambientes externos.

O sistema proposto adota uma arquitetura distribuída e modular, composta pelos seguintes elementos principais:



2. FUNCIONALIDADES PRINCIPAIS

2.1 DETECÇÃO DE EPI

- **Detecção em Tempo Real de Ausência/Presença de EPI:** A funcionalidade central é analisar continuamente as imagens da câmera para identificar se os trabalhadores estão utilizando os Equipamentos de Proteção Individual obrigatórios para a área monitorada.
- **Processamento de Visão Computacional Embarcado:** Utiliza o poder de processamento do Raspberry Pi e bibliotecas como OpenCV, juntamente com um modelo de detecção de objetos (YOLOv8n), para realizar a análise das imagens diretamente no dispositivo, sem depender de processamento na nuvem para a detecção.
- **Alertas Locais Imediatos:** Em caso de detecção de não conformidade (ausência de EPI), o sistema aciona alertas visuais (LEDs) e/ou sonoros (buzzer) no local para alertar o próprio trabalhador sobre a necessidade de utilizar o equipamento.
- **Notificação Remota para Supervisão:** Através da conexão de rede (Wi-Fi, Ethernet ou 4G), o sistema envia alertas para um sistema de supervisão (via protocolo MQTT), permitindo que os responsáveis monitorem as ocorrências em tempo real e tomem as ações necessárias.
- **Registro Local de Eventos:** O sistema armazena um histórico das detecções (com timestamp e informações sobre os EPIs detectados ou ausentes) localmente no cartão SD do Raspberry Pi para fins de auditoria, análise de tendências e rastreamento de conformidade.
- **Configuração Flexível de EPIs:** O sistema pode ser configurado para detectar diferentes tipos de EPIs, dependendo das necessidades específicas do ambiente industrial e das classes treinadas no modelo de detecção.
- **Operação Autônoma:** Uma vez configurado e implantado, o sistema opera de forma contínua e autônoma, minimizando a necessidade de intervenção manual para o monitoramento do uso de EPIs.
- **Adaptabilidade a Diferentes Ambientes:** Com o case industrial adequado e as devidas configurações de hardware e software, o sistema pode ser adaptado para operar em diversas condições ambientais encontradas em diferentes setores da indústria.

3. MODELO CAD E MONTAGEM

- Visualizável no software *FreeCAD* em *3D*
- Organização por camadas/grupos:

4. Descrição Detalhada dos Softwares Embarcados

A seguir estão descritos os principais softwares embarcados no robô, todos selecionados com base em compatibilidade com o sistema embarcado (Raspberry Pi), eficiência em tempo real e suporte à integração com YOLOv8n:

Categoria	Nome / Tecnologia	Função	Descrição / Justificativa	Tipo / Observação
Sistema	Raspberry Pi OS	SO embarcado	Versão leve e estável para o	Imagem

Categoria	Nome / Tecnologia	Função	Descrição / Justificativa	Tipo / Observação
Operacional	(Lite)		Raspberry Pi. Suporta Python, bibliotecas de IA e comunicação sem sobrecarregar o hardware.	minimalista baseada em Debian
IA / Visão Computacional	YOLOv8n (Ultralytics)	Detecção de EPIs em tempo real	Modelo leve, rápido e com boa acurácia para uso em edge devices. Otimizado para Raspberry Pi 4 com aceleração por NPU ou CPU.	Script Python com OpenCV
Framework IA	Ultralytics + PyTorch	Treinamento e inferência do modelo	PyTorch é a base para treinar o YOLOv8; Ultralytics fornece ferramentas e API de uso simples.	Python (>=3.8)
Visão Computacional	OpenCV	Captura e manipulação de imagens	Amplamente usado para pré-processamento, análise e renderização de imagens na detecção.	Integrado ao script do YOLO
Leitura de Sensores	Adafruit_CircuitPython_BH1750	Leitura de sensor de luz	Biblioteca oficial da Adafruit para o sensor BH1750; garante ajuste automático de exposição da câmera.	Script Python
Comunicação LoRa	pyLoRa / RadioHead (via Arduino IDE)	Comunicação LoRa	Biblioteca usada no ESP32 para enviar mensagens MQTT-like para o gateway remoto.	Código em Arduino (ESP32)
Broker MQTT (opcional)	Mosquitto	Transmissão dos dados para nuvem/servidor	Caso haja conexão com gateway, o Mosquitto pode ser usado para envio remoto de eventos de EPI detectados.	Executado local ou remoto
Script de Alerta	`alert_handler.py`	Geração de alerta local (buzzer, LED)	Script Python que ativa o buzzer e LED quando um EPI é detectado incorretamente ou ausência de EPI.	Python + GPIO
Gerenciador de Energia	`battery_monitor.py`	Verifica status da bateria e envia aviso	Script que monitora carga da bateria via ADC e alerta via LoRa se nível crítico for atingido.	Python (opcional, com ADC)

Categoria	Nome / Tecnologia	Função	Descrição / Justificativa	Tipo / Observação
Monitoramento e Log	`log_events.py`	Registro de eventos locais	Grava eventos de detecção (data, hora, tipo de falha) em arquivo local (CSV ou SQLite).	Python
Interface Web (opcional)	Flask + Bootstrap	Visualização local por Wi-Fi (em hotspot)	Pequeno servidor web para exibir imagens detectadas, histórico de alertas e status do sistema.	Python (Flask), HTML/CSS
Ambiente de Desenvolvimento	VSCode / Thonny / Jupyter Notebook	Criação e teste de scripts	Ferramentas leves e práticas para desenvolver e debugar os scripts em Python e visualizar saídas de câmera e sensores.	No PC
Deploy Automatizado	rsync + crontab / Docker (opcional)	Atualização de scripts remotamente	Permite enviar atualizações via SSH, ou empacotar o sistema com Docker se for em versão mais robusta (PC embarcado ao invés de Raspberry Pi).	Linux Shell Script / Dockerfile

📝 Resumo dos Principais Scripts

Nome do Script	Função	Frequência
`detect_epi.py`	Carrega modelo YOLOv8, processa vídeo e detecta EPIs	Contínuo (loop principal)
`alert_handler.py`	Aciona LEDs e buzzer conforme o tipo de alerta	Quando há detecção
`log_events.py`	Registra eventos locais (CSV ou banco)	Toda nova detecção
`battery_monitor.py`	Verifica tensão da bateria e envia alerta	A cada X minutos
`camera_control.py`	Ajusta foco, exposição ou ativação via sensor PIR	Reativo

Todos os softwares foram escolhidos levando em consideração: performance embarcada, suporte à comunidade open source, documentação ativa e integração comprovada com a arquitetura embarcada.

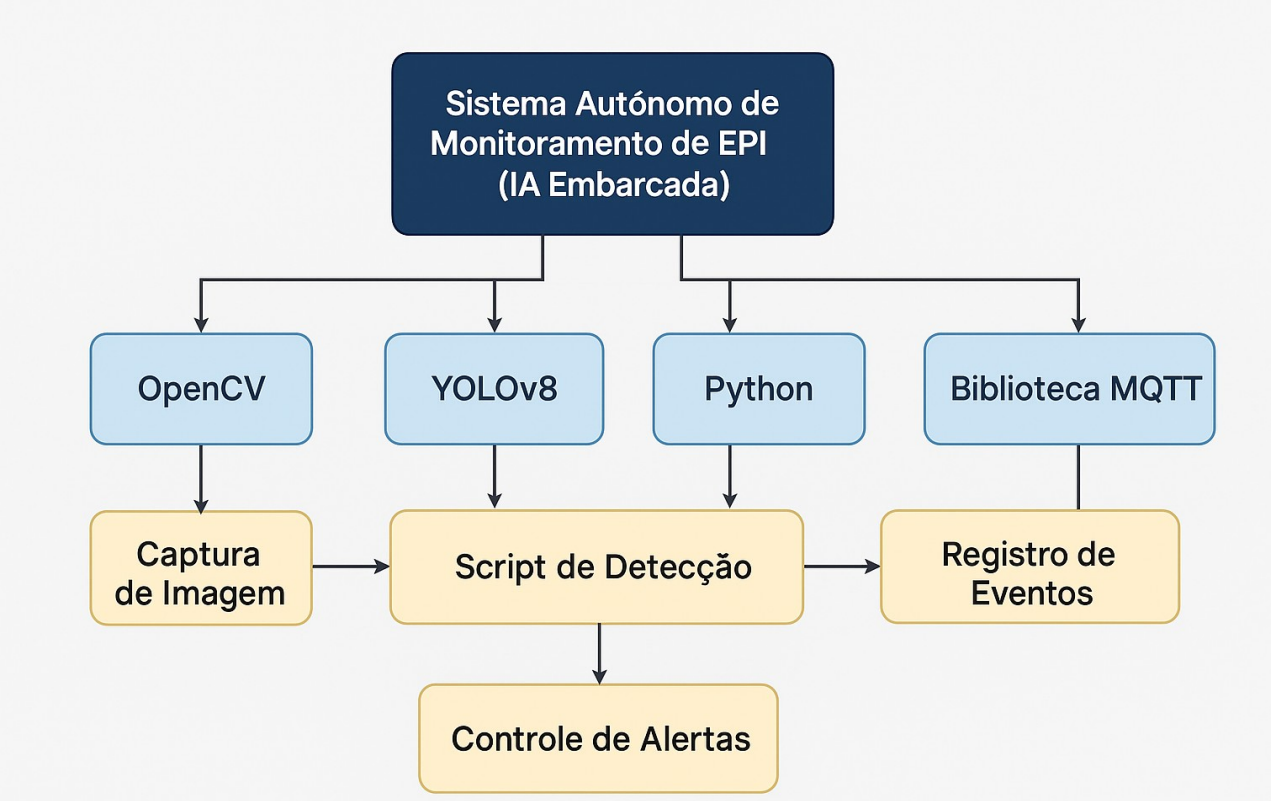


diagrama em camadas que mostra como os softwares se comunicam

5. TABELA DE CUSTOS ESTIMADOS (2025)

Componente	Marca / Modelo Sugerido	Função	Descrição / Justificativa da Escolha	Valor Estimado (R$)
Computador Embarcado	Raspberry Pi 4 Model B (4GB)	Processamento de IA local	Popular, compacto, eficiente em energia, com bom suporte à comunidade e compatível com YOLOv8.	R$ 360,00
Câmera	Raspberry Pi HQ Camera + lente 6mm	Captura de imagem para detecção de EPIs	Alta qualidade de imagem, compatível com Raspberry Pi, ideal para detecção precisa em ambientes industriais.	R$ 120,00
Sensor de	BH1750	Ajuste dinâmico de	Baixo consumo, fácil	R$ 30,00

Componente	Marca / Modelo Sugerido	Função	Descrição / Justificativa da Escolha	Valor Estimado (R$)
Iluminação		exposição da câmera	integração via I2C, ajuda a adaptar a câmera à luz ambiente.	
Alimentação Solar	Mini Painel Solar 20W + Controlador MPPT	Fornecimento de energia em locais abertos	Mantém o sistema autônomo em campo; o controlador MPPT aumenta a eficiência da carga em condições variáveis.	R$ 200,00
Bateria	LiFePO4 12V 6Ah (com BMS)	Armazenamento de energia solar	Segura, durável, recarregável e ideal para sistemas embarcados operando continuamente.	R$ 190,00
Conversor Buck	LM2596 DC-DC Buck Converter	Redução de 12V para 5V (para Raspberry Pi)	Regula eficientemente a tensão da bateria para alimentar o Raspberry Pi.	R$ 30,00
Módulo de Comunicação	ESP32 + LoRa SX1276	Comunicação de longo alcance (alertas)	Comunicação redundante mesmo em locais sem Wi-Fi; baixo consumo de energia e boa cobertura.	R$ 90,00
Alerta Local	Buzzer + LED RGB 12V	Alerta visual e sonoro	Feedback direto ao operador local em caso de falha ou detecção crítica.	R$ 30,00
Sensor de Movimento	PIR HC-SR501	Ativação da câmera por movimento	Economiza energia e ativa a detecção apenas quando há movimento humano detectado.	R$ 18,00
Caixa Hermética	IP67 20x15x10 cm (ABS ou Policarbonato)	Proteção dos componentes	Garante resistência a poeira, água e intempéries, essencial para o ambiente portuário.	R$ 100,00
Placa de Integração	Protoboard ou PCB customizada	Conexão dos componentes eletrônicos	Facilita a montagem e manutenção. Uma PCB otimizada reduz espaço e melhora confiabilidade.	R$ 60,00
Sistema de Montagem	Suportes 3D + Trilhos DIN	Fixação dos módulos	Compactação e organização dos dispositivos em ambiente portátil.	R$ 60,00

Total estimado: R$ 1.228,00

7. CONCLUSÃO

A proposta deste projeto visa a construção de Sistema de Monitoramento de EPI Embarcado e Autônomo. A solução central proposta envolve a utilização de um Raspberry Pi como o cérebro do sistema, conectado a uma câmera para aquisição de imagens. Essas imagens são processadas em tempo real utilizando o modelo de detecção de objetos YOLOv8n, previamente treinado para identificar as classes de EPIs relevantes para o ambiente industrial específico (e.g., capacetes, óculos de proteção, luvas, coletes, etc..).

Quando o sistema detecta a ausência de um ou mais EPIs obrigatórios, ele aciona um sistema de alertas local (LEDs e/ou buzzer) para alertar o trabalhador e, simultaneamente, envia uma notificação para um sistema de supervisão através do protocolo MQTT. Adicionalmente, registros das detecções (com timestamps e, opcionalmente, imagens) podem ser armazenados localmente para futuras análises.