

Sistema RetroSense: Monitoramento de Distância Frontal e Traseira com ESP32 e Blynk

Disciplina: Internet das Coisas e Redes Veiculares

Autores: Raul M. Pereira, João P. de Souza

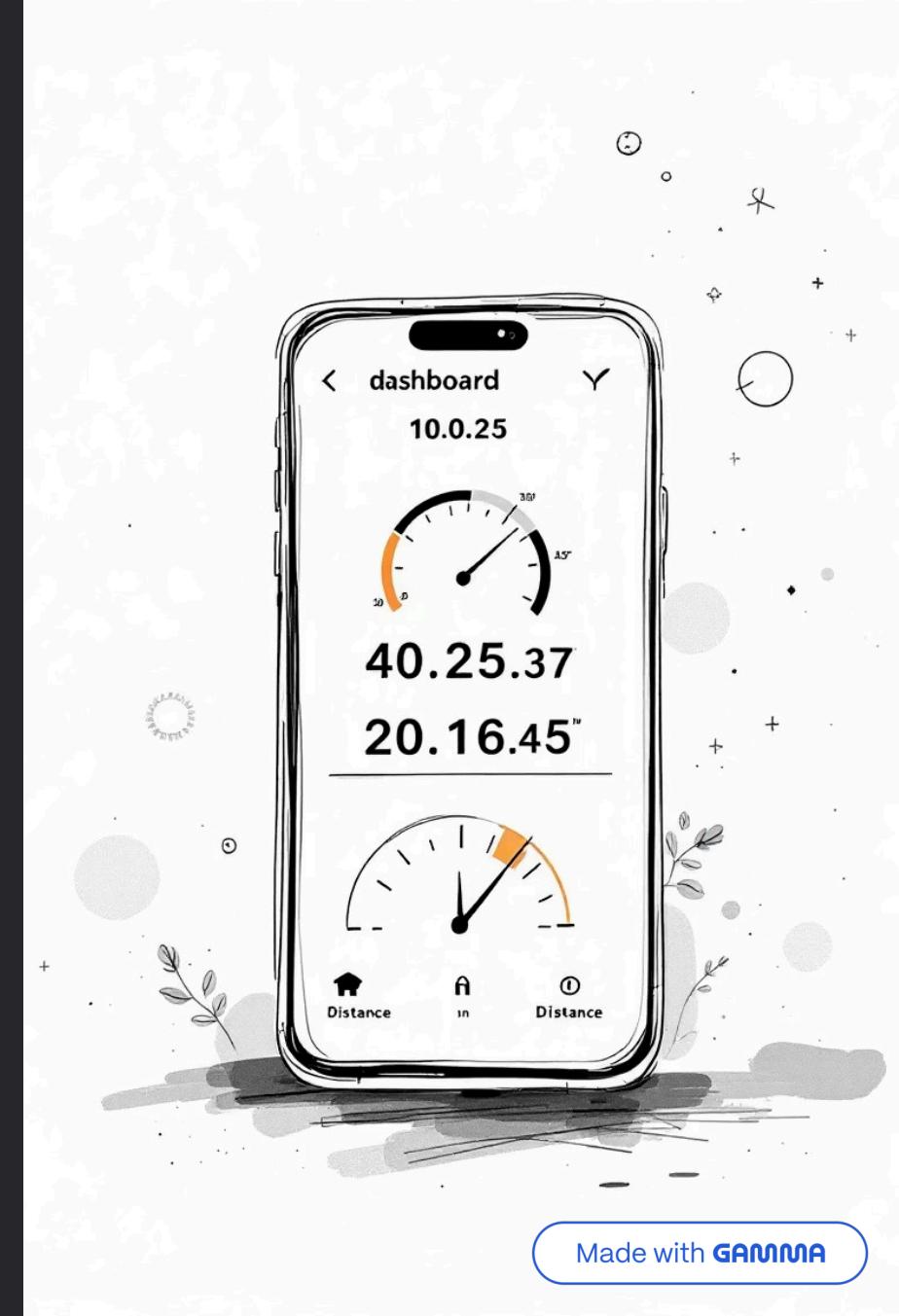
Prof.: Samuel B. Mafra



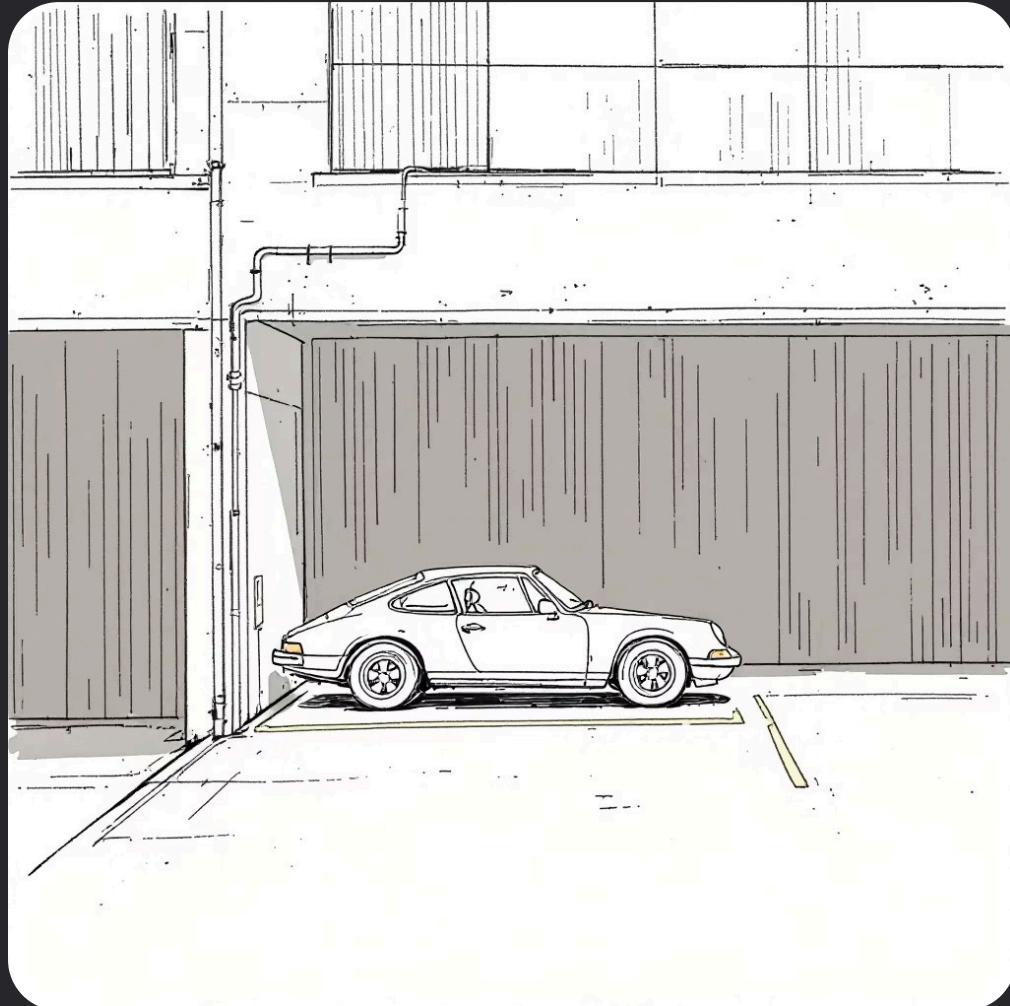
Introdução

RetroSense é um sistema IoT para monitorar, em tempo real, distâncias frontais e traseiras de veículos usando sensores ultrassônicos integrados a um ESP32 e à plataforma Blynk Cloud.

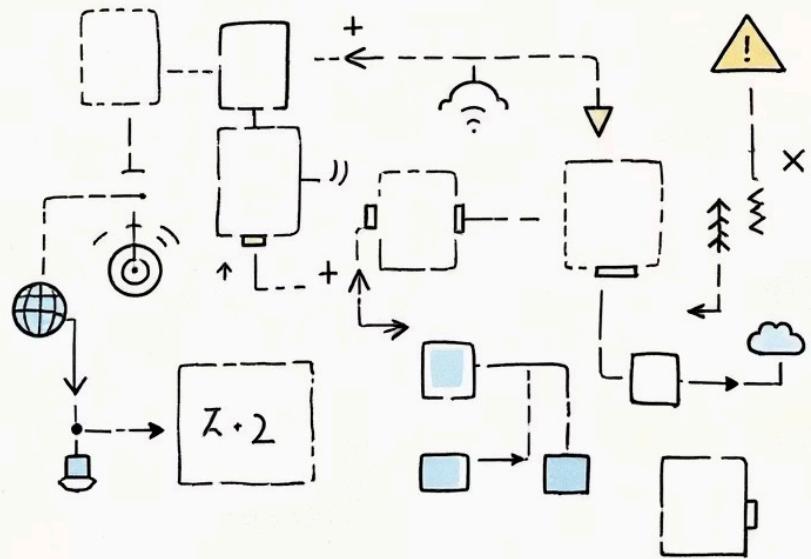
Objetivo: aumentar segurança em manobras com solução de baixo custo e fácil integração.



Motivação



- A maioria dos veículos populares não possui sensores integrados.
- Alternativa de **baixo custo** para retrofit e protótipos autônomos.
- Alertas locais (LED/buzzer) e remotos (dashboard).
- Aplicável a ensino, P&D e soluções comerciais simples.

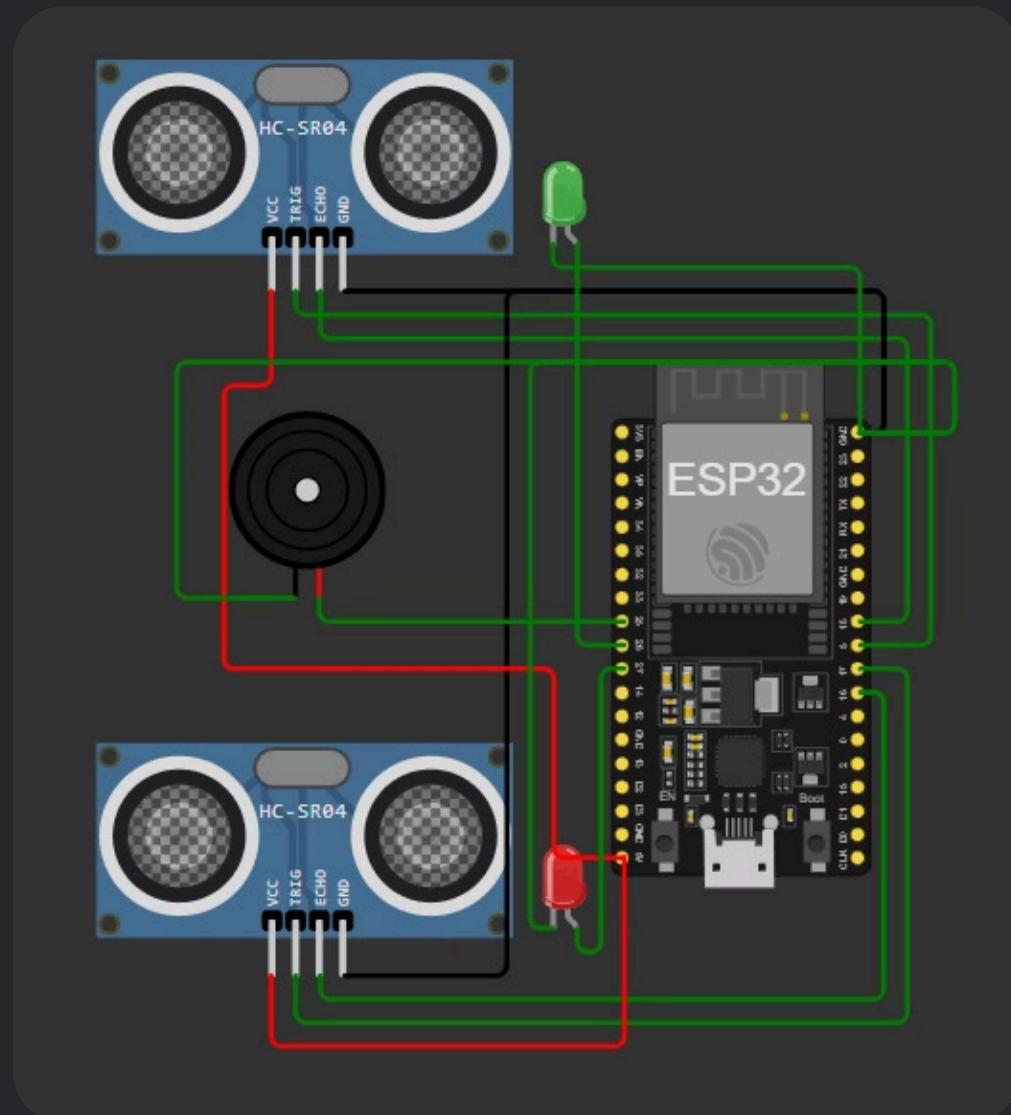


Arquitetura da Solução

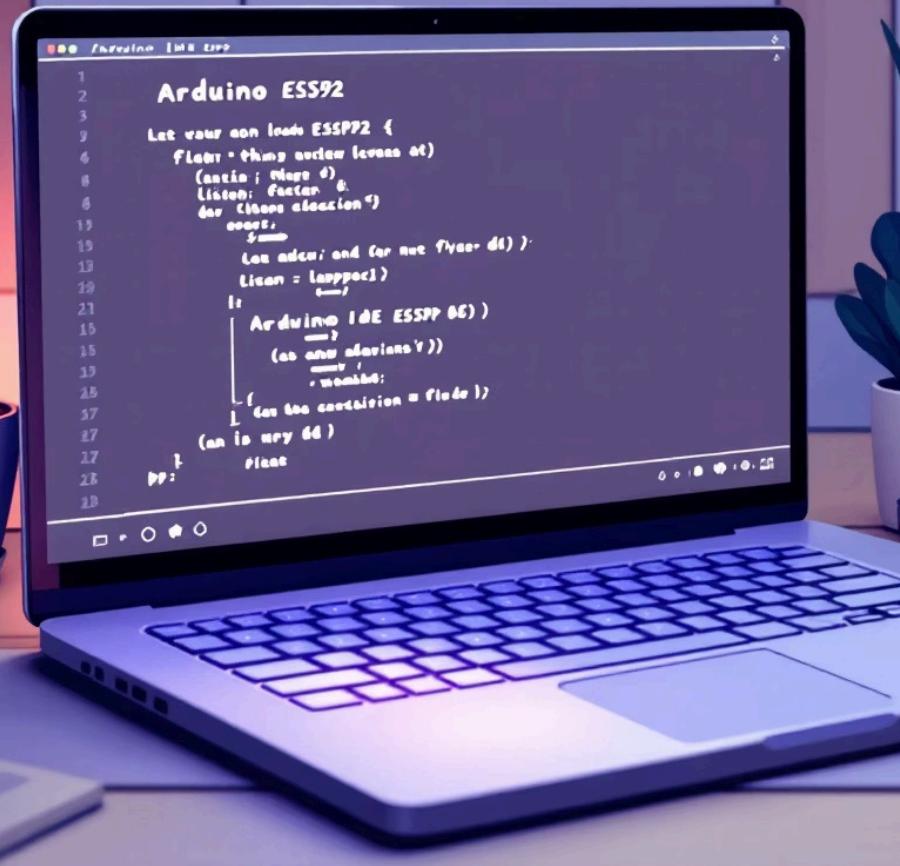
Componentes principais: **ESP32**, sensores ultrassônicos (HC-SR04) frontais e traseiros, LEDs, buzzer e Blynk Cloud.

Fluxo de dados: Sensores → ESP32 → Blynk Cloud → Dashboard (visualização e alertas remotos).

Implementação Técnica — Hardware



- ESP32: coleta de dados e conexão Wi-Fi.
 - HC-SR04: módulos duplicados para frente e traseira.
 - LEDs e buzzer: sinalização imediata de risco.
 - Fonte 5V/3.3V com proteção e filtro para ruído.



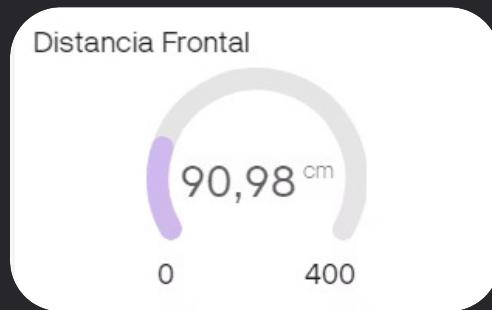
Implementação Técnica — Software

O ESP32 executa loop contínuo de leitura dos sensores.

Classificação local em três níveis: **Seguro** (>25 cm), **Atenção** (10–25 cm) e **Perigo** (<10 cm).

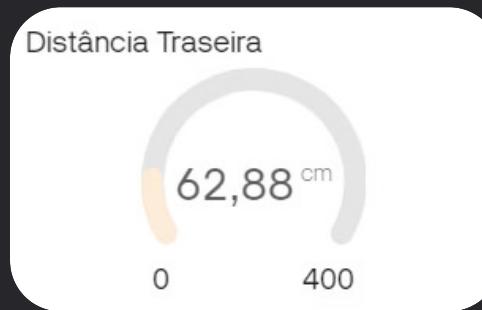
Aciona LEDs/buzzer e publica valores via
Blynk.virtualWrite(V0..V3).

Tela do Dashboard (Blynk) — Widgets



Gauge — Distância Frontal (V0)

Visualização em tempo real do sensor frontal.



Gauge — Distância Traseira (V1)

Indicação separada do sensor traseiro.



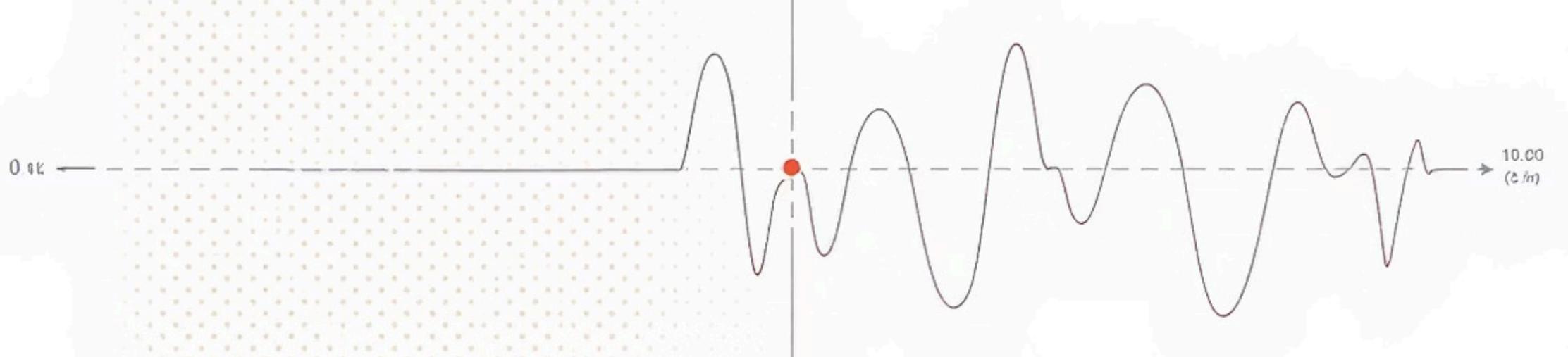
Label — Estado Geral (V2)

Resumo do nível de risco (Seguro/Atenção/Perigo)



Label — Origem do Alerta (V3)

Identifica sensor responsável: Frontal, Traseiro, Ambos, Nenhum.



Lógica de Medição e Alertas

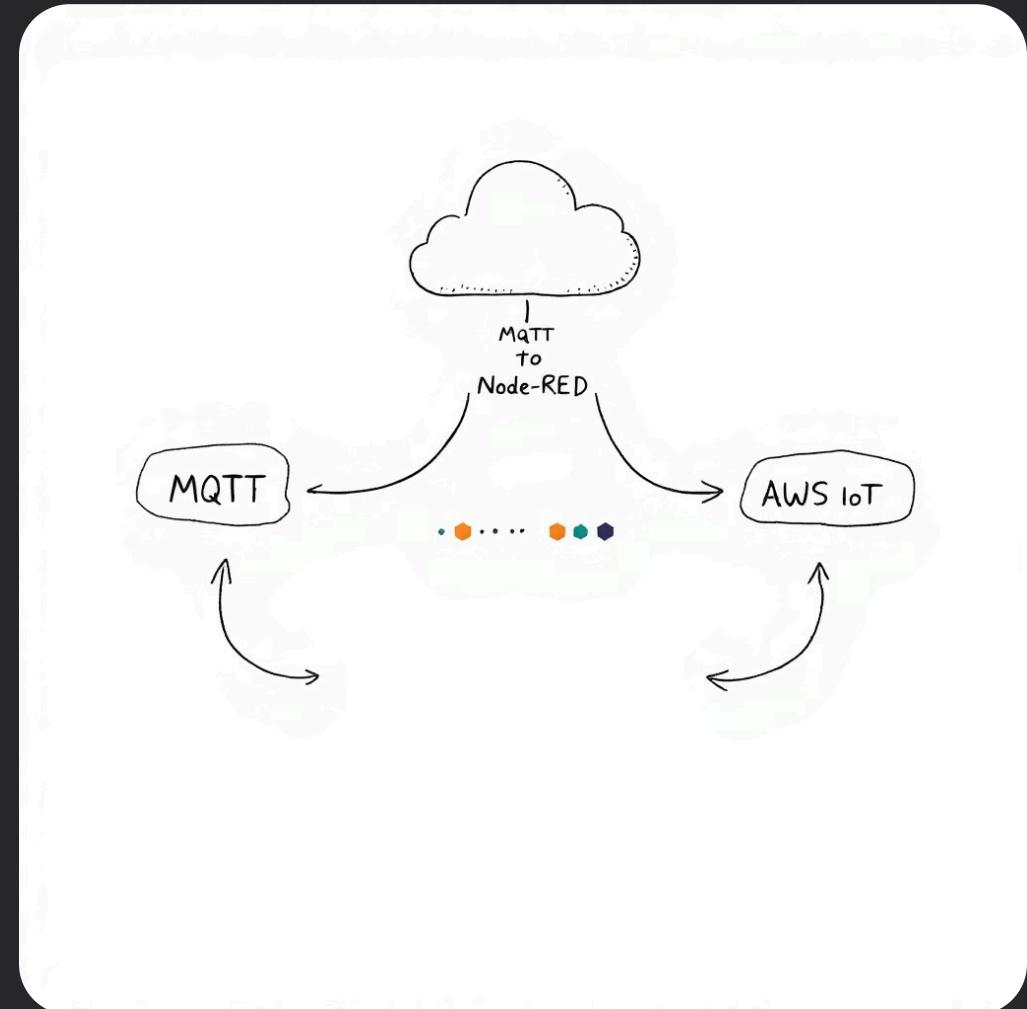
Função `medirDistancia()` converte tempo de eco em distância. A lógica compara ambas leituras e define:

1. Nível de risco (Seguro / Atenção / Perigo),
2. Sensor responsável (Frontal/Traseiro/Ambos),
3. Ativação local (LEDs/buzzer) e envio `Blynk.virtualWrite()`.

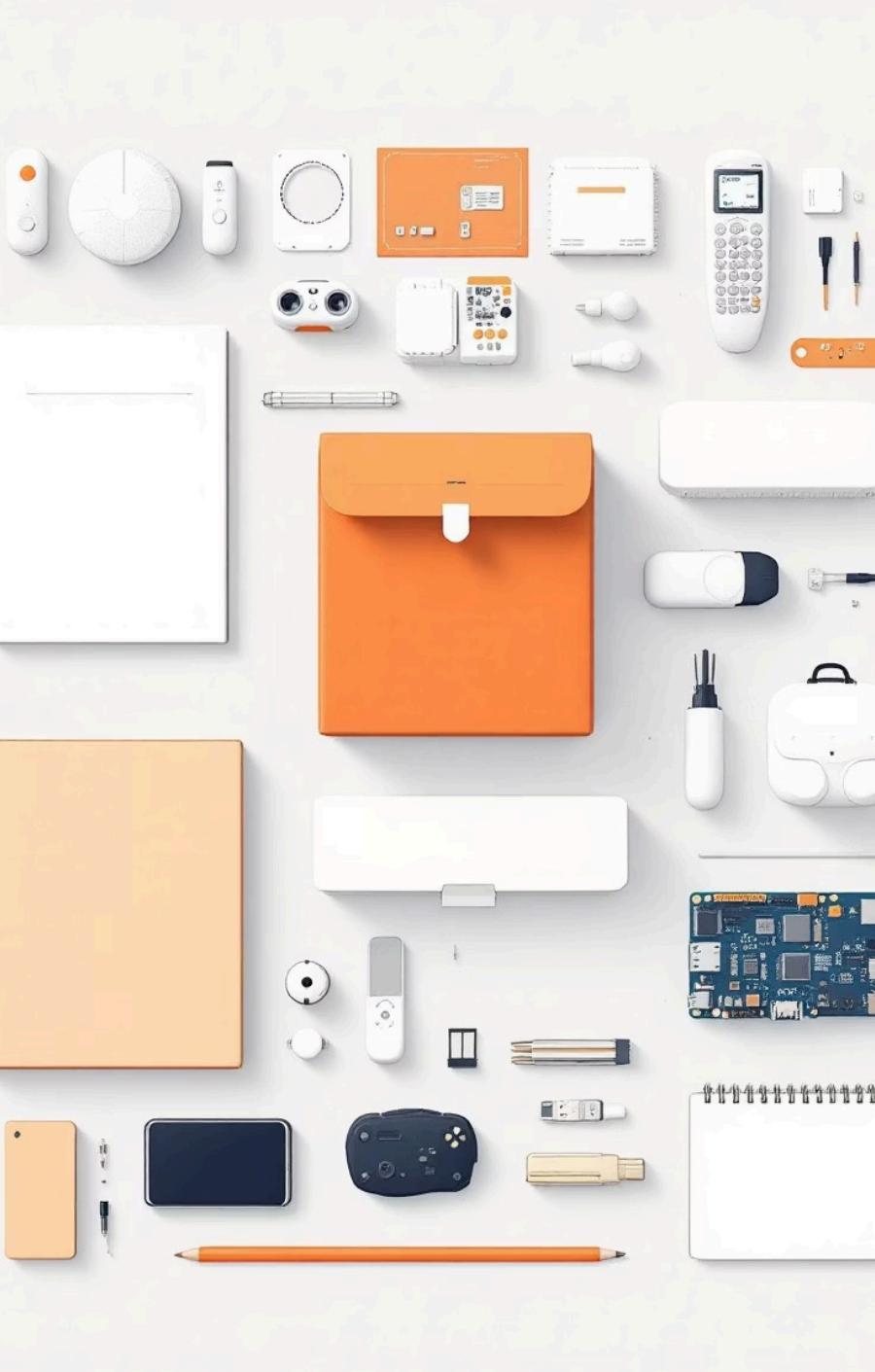
Comparação com Produtos Comerciais e Expansibilidade



Produtos comerciais (Bosch, Multilaser, Positron) oferecem sensores integrados, porém com custo mais alto e sem conectividade IoT nativa.



Diferenciais do RetroSense: integração direta com Blynk, monitoramento remoto, código aberto e possibilidade de expansão (MQTT, IA embarcada, análise na nuvem).



Conclusão, Aplicações e Próximos Passos

Benefícios

Melhora a segurança em manobras, reduz colisões leves e fornece acesso em tempo real via smartphone.

Aplicações

Retrofit automotivo, kits educacionais IoT, protótipos robóticos e sistemas de estacionamento inteligente.

Próximos passos

Adicionar MQTT, sensores ambientais (temperatura/luz), aplicar ML para predição de risco e produzir demonstrador físico.