

**CENTRO UNIVERSITÁRIO *FAMETRO***  
**ENGENHARIA DE SOFTWARE**

**Equipe DEV:** Eric Silva, Fábio Brasileiro, Guilherme Fernandes, Heloísa Cativo, João Sousa Pereira, José Pinto Filho, Luis Oliveira, Mateus Miranda, Victor Fernandes, Victor Abreu.

Documento de análise de requisitos: **ÓCULOS INTELIGENTE DE BAIXO CUSTO  
PARA AUXÍLIO À LOCOMOÇÃO DE PESSOAS COM DEFICIÊNCIA VISUAL**

Manaus/AM  
2025

**Equipe DEV:** Eric Silva, Fábio Brasileiro, Guilherme Fernandes, Heloísa Cativo, João Sousa Pereira, José Pinto Filho, Luis Oliveira, Mateus Miranda, Victor Fernandes, Victor Abreu.

**Documento de análise de requisitos: ÓCULOS INTELIGENTE DE BAIXO CUSTO  
PARA AUXÍLIO À LOCOMOÇÃO DE PESSOAS COM DEFICIÊNCIA VISUAL**

Trabalho do 6º período do curso de engenharia de software do Centro Universitário *FAMETRO*, Câmpus Manaus, como parte dos requisitos para obtenção da nota de **INOVATECH FEIRA DE TECNOLOGIA**.

Orientador: Prof. JEAN CARLOS FIGUEIREDO.

Manaus/AM  
2025

# RESUMO

---

Projeto de prova de conceito (PoC) para um óculos inteligente de baixo custo voltado à auxílio na locomoção de pessoas com deficiência visual. Em 6 semanas será desenvolvido um protótipo integrado composto por: (1) Módulo tátil de distância e vibração (HC-SR04 + vibracall controlado por ESP32/Pico), (2) Módulo de visão para reconhecimento rápido de objetos prioritários (ESP32-CAM / ESP32-S3 com modelo TFLite leve) e (3) App Android mínimo para TTS, calibração e logs. O sistema visa detectar perigos ambientais (fios/obstáculos suspensos, degraus, cadeiras, pessoas, portas) e transmitir alertas táteis ou sonoros ao usuário com baixa latência. Critérios de aceitação mínimos do PoC: acurácia  $\geq 70\%$  nas 5 classes prioritárias em ambiente controlado, latência média de alerta  $\leq 400$  ms, autonomia de prova  $\geq 2$  horas e testes com 1–5 usuários. O projeto prioriza componentes já disponíveis (ESP32, HC-SR04, vibracall, Raspberry Pi Pico, estrutura de óculos/boné) para reduzir tempo e custo.

**Palavras-chave:** ...

# ABSTRACT

---

This project proposes a six-week Proof of Concept (PoC) for the development of a low-cost smart glasses system focused on assisting the autonomous locomotion of visually impaired individuals. The system is designed to detect common environmental hazards (e.g., suspended obstacles, steps, people, doors) and provide feedback to the user with minimal latency. The prototype's architecture integrates three main modules: (1) a haptic distance detection module, comprising an HC-SR04 ultrasonic sensor and vibracall actuators, controlled by a microcontroller (ESP32/Pico); (2) an embedded computer vision module (ESP32-CAM/S3) running an optimized TFLite model for rapid recognition of priority objects; and (3) a minimal Android application for managing audio alerts (TTS), calibration, and log collection. Acceptance criteria for the PoC include a recognition accuracy of  $\geq 70\%$  for priority classes in a controlled environment, an average alert latency of  $\leq 400$  ms, a test battery life of  $\geq 2$  hours, and preliminary validation with 1-5 users. The project prioritizes the use of readily available hardware components to ensure rapid prototyping and low-cost viability..

**Keywords:** .

# SUMÁRIO

---

**Descrição Preliminar do Sistema (breve):.....7**

# INTRODUÇÃO

---

A mobilidade autônoma de pessoas com deficiência visual ainda depende majoritariamente de ferramentas táteis (bengala) e orientação tátil/sonora humana. Embora eficazes, essas soluções apresentam limitações: alcance físico restrito, incapacidade de detectar riscos acima do solo (fios, ramos, sinalizações suspensas) e falta de informação contextual antecipada. Dispositivos comerciais mais avançados costumam ser caros ou não combinam adequadamente sensores de distância, visão computacional e feedback tátil em uma solução compacta e acessível.

Este projeto propõe desenvolver um **óculos inteligente de baixo custo** como solução complementar à bengala, combinando sensores ultrassônicos para detecção de distância, câmera para reconhecimento de objetos relevantes e mecanismos de entrega de alerta (vibração e áudio). A arquitetura modular — Módulo 1 (distância + vibração), Módulo 2 (câmera + reconhecimento), Módulo 3 (app móvel) — permite desenvolvimento incremental e validação contínua ao longo de 6 semanas. O objetivo do PoC é entregar um protótipo funcional que comprove viabilidade técnica, usabilidade e ganho de percepção espacial para o usuário, utilizando majoritariamente hardware já disponível na equipe para acelerar a execução e minimizar custos.

# 1. Plano de projeto

---

**Projeto:**

Óculos inteligente de baixo custo que detecta perigos ambientais e fornece alertas táteis e sonoros para melhorar a locomoção de pessoas com deficiência visual.

**Registro de Alterações**

Versão	Responsáveis	Data	Alterações
0.1	Equipe de Desenvolvimento	01/05/2025	Versão inicial do plano de projeto

**1.1 Introdução**

Projeto PoC de 6 semanas para desenvolve/r um wearable ou tecnologia vestível que combina sensores de distância, visão computacional e app móvel, aumentando a percepção espacial além da bengala tradicional.

**1.2 Escopo do projeto**

**Projeto:** Inclui: desenvolvimento de protótipo (Módulos 1 e 2), app Android mínimo, integração e testes de campo; exclui: produção em escala e certificações regulatórias.

**Justificativa:**

A bengala não detecta riscos acima do solo nem fornece informação contextual — um óculos assistivo acessível reduz riscos e amplia autonomia do usuário.

**Produto:**

Protótipo funcional integrado: óculos com sensor ultrassônico + câmera, pulseira vibratória, app Android para TTS, calibração e status de sistema, e documentação técnica mínima.

**Descrição Preliminar do Sistema (breve):**



Sistema modular composto por: Módulo 1 — sensores de distância e atuador vibratório (detecção local e alerta tátil); Módulo 2 — câmera com modelo TFLite para reconhecimento rápido de classes prioritárias; Módulo 3 — app Android para receber eventos, executar TTS, calibrar e registrar logs. Comunicação por BLE/Wi-Fi com fallback para processamento no celular quando necessário

1.3 Descrição Preliminar do Sistema

- Hardware disponível: ESP32 (com/câmera), ESP32-S3/ESP32-CAM, Raspberry Pi Pico, HC-SR04, motores de vibração, estrutura wearable, smartphone Android.
- Performance: latência  $\leq 400$  ms do evento ao alerta; autonomia mínima de prova  $\geq 2$  horas.
- Precisão: acurácia  $\geq 70\%$  nas 5 classes prioritárias em ambiente controlado.
- Comunicação: BLE obrigatório; Wi-Fi opcional (offload/telemetria).
- Energia: bateria 3.7 V até 5 V com circuito de carga e proteção.
- Usabilidade: padrões de vibração diferenciáveis, TTS claro e opções de volume/tempo.
- Testes: 1-5 usuários para validação de campo dentro das 6 semanas.

1.4. Modelo de Ciclo de Vida

**Modelo:** Incremental com sprints semanais.

**Justificativa:** Permite validar hipóteses de negócio (planos) e incorporar feedback dos primeiros usuários/parceiros rapidamente.

Atividades e papéis:

Etapa	Atividades Principais	Insumos	Produtos	Papéis Envolvidos
Levantamento	Pesquisas de mercado, entrevistas com salões	Questionários, benchmarks de concorrentes	Documento de requisitos e personas	Analista de Requisitos
Projeto	Modelagem de dados e arquitetura, UX/UI	Requisitos, personas	Protótipos, diagramas UML, wireframes	Arquiteto de Software, Designer

Implementação	Desenvolvimento de módulos( front-end, backend e APIs)	Backlog de funcionalidades	Módulos entregues por sprint	Desenvolvedores Full-stack
Testes	Testes funcionais, de usabilidade e end-to-end	Builds de sprint	Relatórios de bugs e recomendações	QA/Testador
Implantação	Deploy em staging e produção, configuração CI/CD	Versão candidata (RC)	Sistema em produção	DevOps, Gerente de Projeto
Manutenção	Correções, Otimizações e novos incrementos	Feedback de usuários, métricas	Atualizações e patches	Suporte técnico, Desenvolvedores

## 1.5. Estrutura da Equipe do Projeto

### Organograma:

- Product Owner: Fabio Brasileiro
- Scrum Master: Heloisa Cativo
- Tech Lead / Arquiteto: Jose Filho
- Devs Backend (2–3): Victor Fernandes, Victor Abreu
- Dev Frontend: Luis Oliveira, Eric Silva, Joao Pereira
- DBA: Fabio Brasileiro
- QA: Mateus Miranda
- Config Manager: Jose Filho
- CCB: Victor Abreu

### Matriz de Responsabilidades (exemplo):

Atividade	GP	AR	DEV	QA	UX/UI	DEVOPS
Levantamento	R	R				
Projeto	C	R	C		R	
Implementação	A	C	R	C	C	
Testes	C		C	R		
Implantação	A		C	C		R
Manutenção	A	C	R	R		C

(R: Responsável, A: Aprovador, C: Consultado)

## 1.6. Definição de Medidas (KPIs)

### Medida 1: Taxa de Entrega de Funcionalidades

- Nome: Funcionalidades entregues por sprint
- Definição: Número de funcionalidades concluídas em cada sprint
- Tipo: Derivada
- Entidade Medida: Sprint

- Atributo Medido: Entregas realizadas
- Escala: Quantitativa absoluta
- Unidade de Medida: Funcionalidades
- Fórmula:  $\text{Total de funcionalidades concluídas} / \text{Total previstas na sprint}$
- Procedimento: Verificação dos tickets finalizados no Github
- Momento: Ao fim de cada sprint
- Responsável pela Medição: Scrum Master
- Procedimento de Análise: Comparação com metas da sprint
- Responsável pela Análise: Gerente de Projeto

**Justificativa:** Ajuda a avaliar o ritmo de desenvolvimento e a adequação ao planejamento.

## 2. Documento de requisitos

---

### 2.1. Introdução

**Mini-mundo:** ambiente real e delimitado onde o PoC será validado — calçadas urbanas, áreas internas (salas, corredores), e trechos curtos de rua com fluxo moderado de pedestres. Cenários prioritários: obstáculos baixos (cadeiras, degraus), obstáculos suspensos (fios, galhos), pessoas em movimento, portas/aberturas. Condições: iluminação diurna e noturna urbana típica (iluminação pública), piso irregular e ruído sonoro ambiente moderado. Testes serão feitos com usuários com deficiência visual e/ou cegueira total, usando bengala como baseline.

Resumo do mini-mundo:

- Espaço físico: calçada pública e ambiente interno controlado (sala/corredor).
- Tipos de obstáculos: degraus, cadeiras, postes, fios, portas, pedestres/animais.
- Condições ambientais: iluminação variada, ruído ambiente, trânsito moderado.
- Uso esperado: caminhada normal (1–5 km/h), paradas comuns para inspeção.

### 2.2. Descrição do Propósito do Sistema

O sistema tem por finalidade **aumentar a percepção espacial** de pessoas com deficiência visual através de um wearable (óculos + pulseira vibratória + app), detectando perigos e comunicando alertas táteis e/ou sonoros em tempo real para prevenir colisões e incidentes. No PoC de 6 semanas o objetivo é demonstrar viabilidade técnica, usabilidade e métricas mínimas de desempenho (acurácia, latência, autonomia).

Objetivos práticos:

- Detectar obstáculos próximos e emitir vibração proporcional.
- Reconhecer 5 classes prioritárias com modelo leve (TFLite) e emitir TTS via app.
- Validar integração entre hardware e app com latência aceitável para navegação.

## 2.3. Descrição do Minimundo

### Entidades do mini-mundo

- **Usuário** (pessoa com deficiência visual)
- **Óculos inteligente** (dispositivo embarcado)
- **Pulseira vibratória** (atuador tátil)
- **Smartphone Android** (app + TTS)
- **Obstáculo** (objetos físicos no ambiente)
- **Operador / Testador** (configura e observa testes)

## 2.4. Requisitos de Usuário

### Requisitos Funcionais

Identificador	Descrição	Prioridade	Depende de
RF01	<b>Medição de Distância:</b> O sistema deve medir distância em frente ao usuário usando HC-SR04 e disponibilizar leitura atual a cada 200–500 ms.	Alta	-
RF02	<b>Alerta Tátil Proporcional:</b> Ao detectar distância abaixo de thresholds configuráveis, o dispositivo deve acionar o vibracall com intensidade proporcional (fraco/médio/alto) em $\leq 100$ ms após leitura.	Alta	RF01

RF03	<b>Captura de Imagem:</b> O sistema deve capturar imagens a cada X segundos (configurável) e disponibilizar para inferência local ou envio ao app quando solicitado.	Alta	RF02
RF04	<b>Reconhecimento de Objetos (5 classes prioritárias):</b> O sistema deve inferir localmente (ou via app) as classes: fio/ramo, degrau, cadeira/obstáculo baixo, pessoa, porta. Para o PoC, acurácia mínima alvo: 70% em ambiente controlado.	Alta	RF02
RF05	<b>Comunicação com App (BLE):</b> Enviar eventos (tipo, distância, timestamp, confidence) via BLE para app Android e receber comandos de calibração.	Alta	RF04
RF06	<b>TTS no App:</b> O app deve reproduzir mensagem curta correspondente ao evento (ex.: “fio à frente, 1,2 metros”) em $\leq 400$ ms após recebimento do evento	Média	RF05
RF07	<b>Registro/Logs:</b> Registrar localmente eventos com campos: timestamp, tipo, distância, confidence, bateria.	Alta	RF03
RF08	<b>Gestos/Botões de Controle:</b> Permitir ligar/desligar alertas via botão físico no wearable.	Médio	
RF09	<b>Fallback de processamento:</b> Caso a inferência local não alcance metas de latência ou acurácia, permitir offload da imagem para o smartphone para inferência.	Médio	
RF10	<b>Atualização de Parâmetros via App:</b> Permitir ajustes em thresholds de distância, sensibilidade de vibração, taxa de captura de imagem e volume de TTS pelo app.	Médio	
RF11	<b>Verificar dispositivos pelo App:</b> O aplicativo deve exibir o status da conexão BLE (Conectado/Desconectado), o nível de bateria restante do wearable (em porcentagem) e o status operacional dos sensores principais.	Médio	RF05

RF12	<b>Alerta de Bateria Baixa:</b> O sistema deve notificar ativamente o usuário (via vibração específica no wearable e alerta sonoro/TTS no app) quando o nível de bateria do wearable atingir um nível crítico (ex: 20%).	Alta	RF05, RF07
RF13	<b>Confirmação de Inicialização (Auto-Teste):</b> Ao ser ligado, o wearable deve executar um auto-teste (verificar conexão do HC-SR04 e da câmera ) e emitir um sinal claro de "Sistema Pronto" (ex: uma vibração longa e duas curtas, ou um TTS "Sistema Ativo" ).	Alta	RF01, RF03, RF06
RF14	<b>Modo de Silêncio Temporário:</b> O usuário deve poder pausar (silenciar) temporariamente todos os alertas (táteis e sonoros) através de um comando no wearable (ex: um clique duplo no botão RF08 ) ou no App (via RF10 ).	Média	RF08, RF10
RF15	<b>Exportação de Logs de Teste:</b> O App deve possuir uma função (acessível pelo Operador/Testador) para exportar os logs de eventos (RF07 ) em um formato padrão (ex: CSV ou JSON) para análise de desempenho offline.	Média	RF05, RF07

#### Regras de Negócio

Identificador	Descrição	Prioridade	Depende de
RN01	Se a distância medida for $\leq 0.8$ m → vibração alta + alerta TTS prioritário.	Alta	RF02

RN02	Se a distância estiver entre 0.8 m e 1.5 m → vibração média.	Alta	RF05
RN03	Se a distância > 1.5 m e detector de objeto identificar classe prioritária (fio/ramo) dentro do campo de visão → vibração fraca + TTS.	Média	RF05
RN04	Em caso de conflito (múltiplos eventos simultâneos), priorizar evento de menor distância; se distância similar, priorizar evento classificado como “perigo suspenso” (fio/ramo).		
RN05	odos os eventos registrados devem ter timestamp e tipo salvo localmente para posterior exportação.		
RN06	Se a bateria < 15% → notificar no app e reduzir frequência de captura de imagem para economizar energia.		
RN07	Comunicação BLE é padrão; se BLE indisponível e Wi-Fi disponível, usar Wi-Fi. Se ambos indisponíveis, armazenar eventos localmente (fallback).		

#### Requisitos Não Funcionais

Identificador	Descrição	Categoria	Escopo	Prioridade	Depende de
RNF01	Latência ponta-a-ponta (detecção → vibração/TTS) média $\leq 400$ ms; vibração acionada em $\leq 100$ ms a partir da leitura do microcontrolador.	Desempenho	Global	Alta	-
RNF02	Acurácia mínima por	Precisão	Global	Alta	-



	classe no PoC $\geq$ 70% em ambiente controlado; taxa de falso positivo $\leq$ 20%.				
RNF03	Autonomia mínima de prova $\geq$ 2 horas com ciclo de uso típico (captura periódica de imagem + vibração esporádica).	Energia	Autenticação	Alta	RF02
RNF04	Interface de calibração acessível no app com opções de volume, intensidade de vibração, e modo silencioso.	Usabilidade	Global	Médio	
	Disponibilidade do sistema durante sessão de teste $\geq$ 95% (tempo sem falhas).	Confiabilidade	Global		
	Dados de usuários e logs armazenados localmente devem ser acessíveis somente por aplicativo com autenticação local (senha/pin simples).	Segurança	Global		

	O firmware deve poder rodar em ESP32/ESP32-S3 e ter alternativa de execução parcial no smartphone.	Portabilidade	Global		
	Peso do conjunto (óculos + eletrônica) deve ser confortável para uso contínuo curto (meta $\lesssim$ 150–200 g, dependendo da estrutura).	Peso/Ergonomia	Global		

## 2.5. Casos de Teste

Id	Requisitos relacionados	Descrição	Pré-condições	Entradas	Fluxo	Resultados esperados	Pós-condições
CT01	RF01, NFR2	Medição HC-SR04 (Bench)	Sensor conectado ao ESP32, bancada com régua.	Leitura $\pm 10$ cm do valor real em 90% das tentativas.	Posicionar objeto a 0.5 m, 1.0 m, 1.8 m; ler valores.	Validar leitura correta do HC-SR04	
CT02	RF02	PWM e Intensidade de Vibração	Vibracall conectado e instrumentado (medir vibração qualitativa)	Três níveis perceptíveis e estáveis.	Enviar PWM para níveis fraco/médio/alto; observar motor.	Verificar intensidade proporcional ao PWM.	
CT03	RF3, RF5	Captura de Imagem (ESP32-CAM)		Imagem legível transferida com sucesso	Capturar imagem, salvar local, enviar via BLE.	Garantir snapshots e envio ao buffer.	
CT05		Testes de Integração					

# 3. Especificação de Requisitos

---

## 1. Introdução

### 1.1 Propósito do Documento

Este documento tem como objetivo especificar detalhadamente os requisitos funcionais e não funcionais, os modelos de caso de uso, os subsistemas e os diagramas de apoio do projeto **Óculos Inteligente de Baixo Custo para Auxílio à Locomoção de Pessoas com Deficiência Visual (LUMI)**.

O documento visa fornecer uma visão clara e completa para desenvolvedores, testadores e stakeholders, permitindo o correto entendimento, desenvolvimento e validação do sistema.

### 1.2 Escopo do Sistema

O sistema consiste em um **dispositivo wearable** (óculos com sensores ultrassônicos e câmera integrada) acoplado a uma **pulseira vibratória** e um **aplicativo móvel Android**.

O objetivo é auxiliar pessoas com deficiência visual a identificar obstáculos e perigos ambientais, emitindo **alertas táteis e sonoros em tempo real**.

Componentes principais:

- Módulo 1 – Sensoriamento de distância e vibração (HC-SR04 + motor vibratório).
- Módulo 2 – Câmera embarcada com IA leve (ESP32-CAM / ESP32-S3 + TensorFlow Lite).
- Módulo 3 – Aplicativo Android para calibração, TTS (text-to-speech), logs e controle.

## 2. Descrição Geral

### 2.1 Perspectiva do Produto

O produto é um **sistema embarcado integrado** que combina hardware e software. O dispositivo será controlado por um microcontrolador ESP32, comunicando-se via BLE com um aplicativo Android.

O sistema deve operar de forma autônoma, com feedback vibratório proporcional à proximidade de obstáculos e alertas sonoros gerados no aplicativo.

### 2.2 Usuários e Stakeholders

Tipo de Usuário	Descrição	Nível de Acesso
Usuário Principal	Pessoa com deficiência visual que utiliza o dispositivo.	Uso direto do wearable e app.
Operador/Testador	Pessoa que auxilia em testes e calibrações.	Acesso técnico e coleta de logs.
Equipe de Desenvolvimento	Engenheiros e programadores.	Manutenção e atualização do sistema.
Orientador / Stakeholder Acadêmico	Avaliador do projeto.	Acesso a relatórios e métricas.

### 2.3 Restrições

- Tempo de resposta do sistema  $\leq 400$  ms.
- Acurácia de reconhecimento  $\geq 70\%$  (ambiente controlado).
- Autonomia mínima de 2 horas.
- Peso máximo do conjunto  $\leq 200$  g.

### 2.4 Suposições e Dependências

- Testes realizados em ambientes controlados (interno e externo leve).
- Smartphone Android com Bluetooth e TTS compatível.
- Alimentação via bateria recarregável de 3.7–5 V.

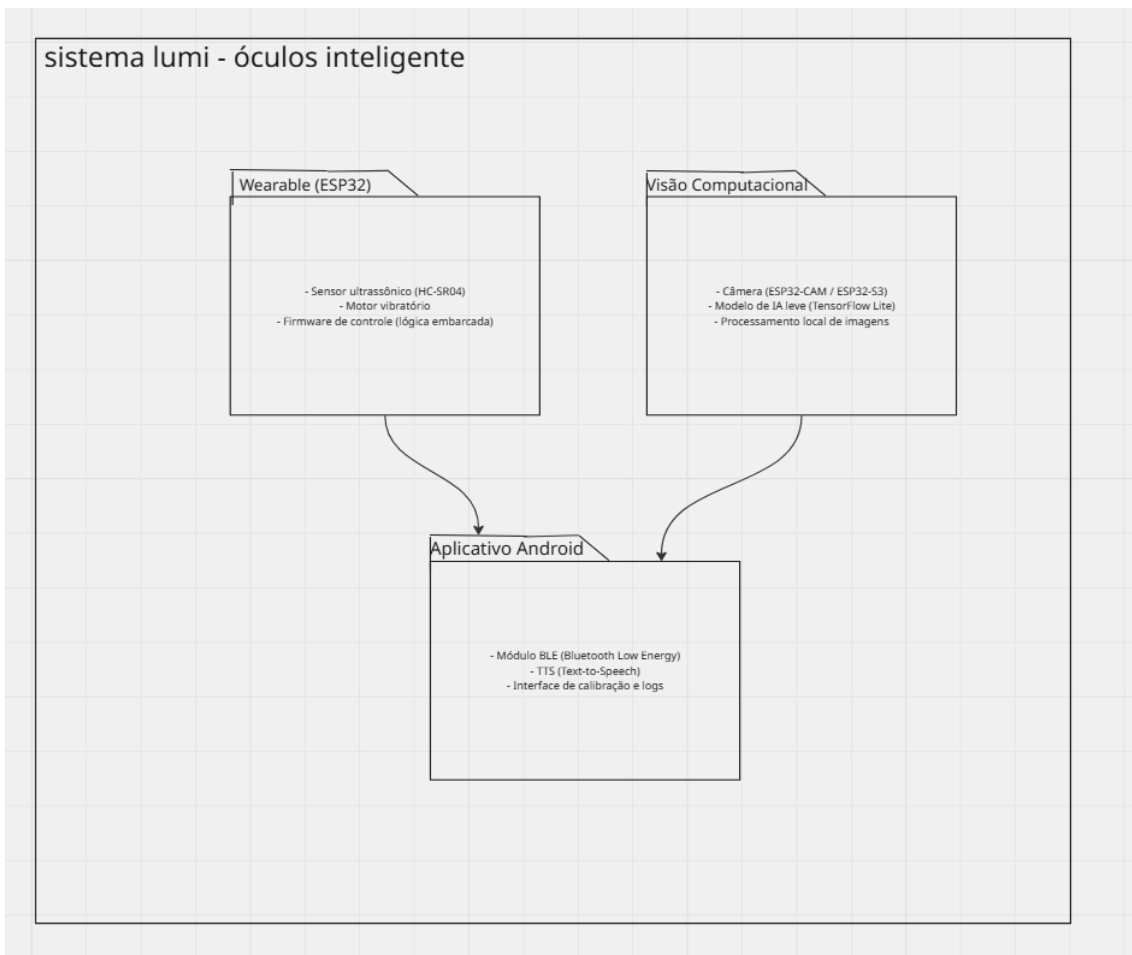
## 3. Modelagem de Subsistemas

### 3.1 Diagrama de Pacotes

#### Descrição:

O sistema é dividido em três subsistemas principais:

1. **Wearable** – Controle de sensores e atuadores (ESP32).
2. **Visão Computacional** – Captura e inferência de imagens.
3. **Aplicativo Android** – Gerenciamento de configuração, TTS e registro de logs.



(Imagem – Diagrama de Pacotes do Sistema)

### 3.2 Descrição dos Subsistemas

Subsistema	Descrição	Interfaces
<b>Sensoriamento Tátil</b>	Realiza medições de distância e aciona vibração conforme proximidade.	Sensor HC-SR04, motor vibratório, ESP32.
<b>Visão Computacional</b>	Reconhece objetos e classes prioritárias (fio, degrau, pessoa, porta, obstáculo).	Câmera ESP32-CAM / modelo TFLite.
<b>Aplicativo Android</b>	Exibe status, calibra sensores, reproduz TTS e armazena logs.	BLE, tela do usuário, TTS.

### 3.3 Diagrama Visual do Sistema

Esta seção apresenta a representação visual consolidada do fluxo de dados e comunicação entre os principais componentes do sistema “**Óculos Inteligente de Baixo Custo para Auxílio à Locomoção de Pessoas com Deficiência Visual**”.

O Diagrama Visual do Sistema (Figura X) ilustra a interação entre os módulos embarcados (ESP32, ESP32-S3, sensores e atuadores), o aplicativo Android e o servidor intermediário. Ele demonstra o ciclo completo de detecção, processamento, transmissão e resposta tátil e sonora para o usuário final.

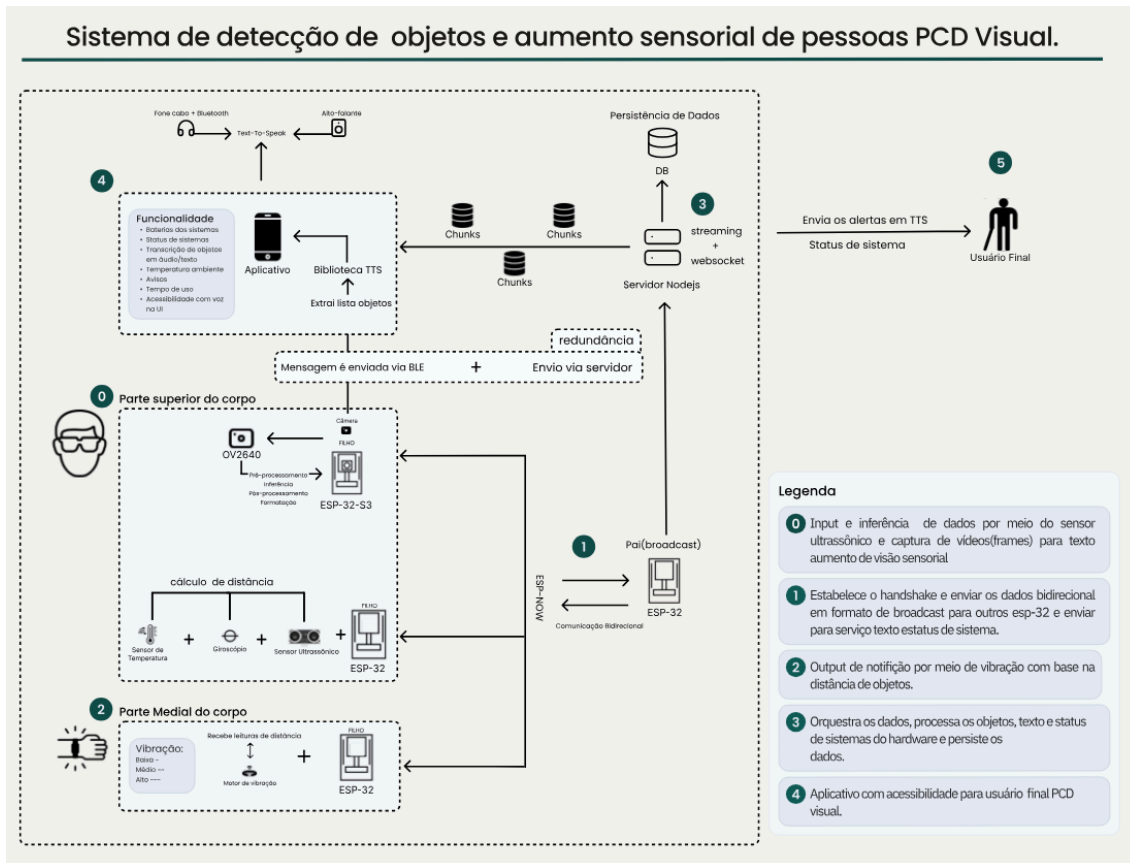
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio de alertas TTS ao usuário.

Principais fluxos representados:

- **Entrada de dados:** sensores ultrassônicos, câmera OV2640, giroscópio e sensores de temperatura;
- **Processamento:** pré-processamento, inferência e formatação dos dados;
- **Comunicação:** transmissão de informações entre ESPs e o servidor via WebSocket;

- Saída: feedback tátil (vibração) e sonoro (TTS);
- Persistência: armazenamento de logs e status de sistema no banco de dados;

Interface: aplicativo Android com acessibilidade por voz e controle de calibração.



(Imagem – Diagrama de Visual)

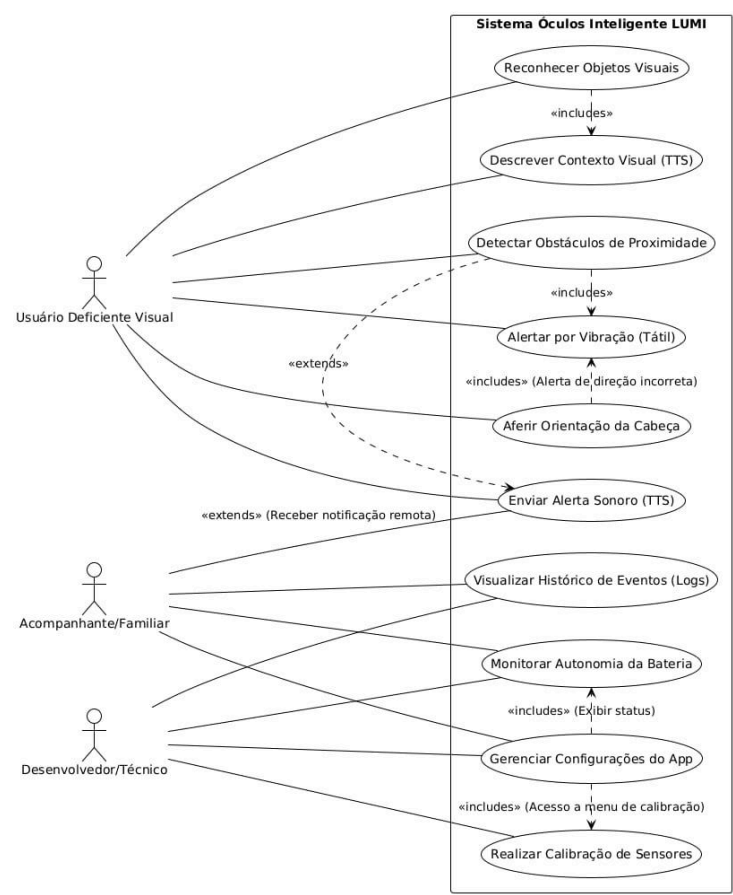


# 4. Modelagem de Casos de Uso

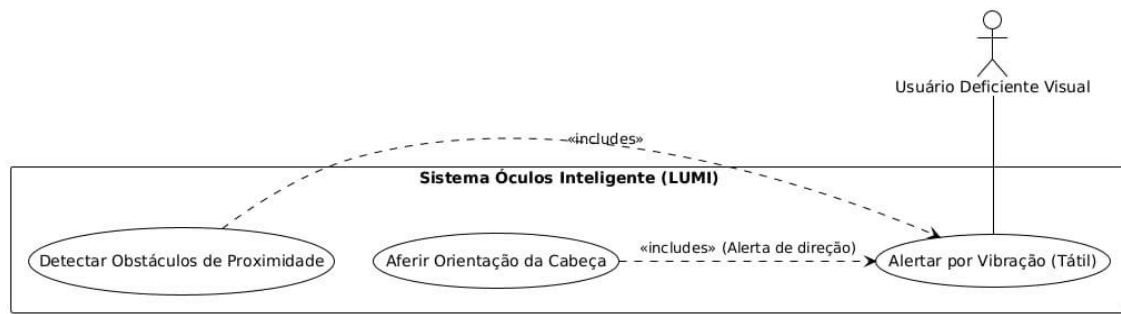
## 4.1 Atores

Ator	Descrição
Usuário	Utiliza o dispositivo durante a locomoção.
Aplicativo Android	Sistema auxiliar que fornece interface, TTS e calibração.
Operador/Testador	Auxilia no teste, calibração e coleta de dados.

## 4.2 Diagrama de Casos de Uso



(Imagem – Diagrama de Casos de Uso do Sistema)



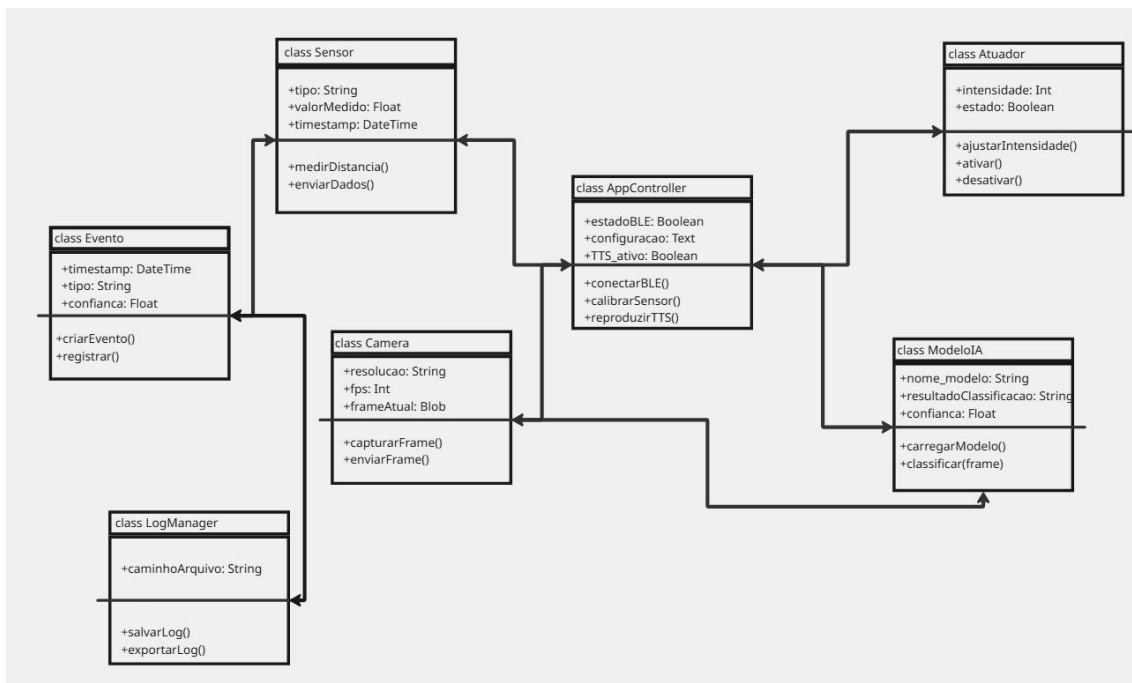
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

### 4.3 Descrição dos Casos de Uso Principais

ID	Nome do Caso de Uso	Descrição	Atores	Requisitos Relacionados
CU0 1	Detectar Obstáculo	Mede a distância à frente e aciona vibração conforme proximidade.	Usuário	RF01, RF02
CU0 2	Reconhecer Objeto	Captura imagem e identifica classes prioritárias.	Usuário, App	RF03, RF04
CU0 3	Emitir Alerta TTS	O app anuncia o tipo e distância do obstáculo.	App	RF06
CU0 4	Calibrar Sistema	Ajusta thresholds de distância, vibração e volume.	Operador	RF10
CU0 5	Exportar Logs	Exporta registros de eventos para análise.	Operador	RF15

## 5. Modelo Estrutural

### 5.1 Diagrama de Classes (ou Entidade-Relacionamento)



(Imagem – Diagrama de Classes do Sistema)

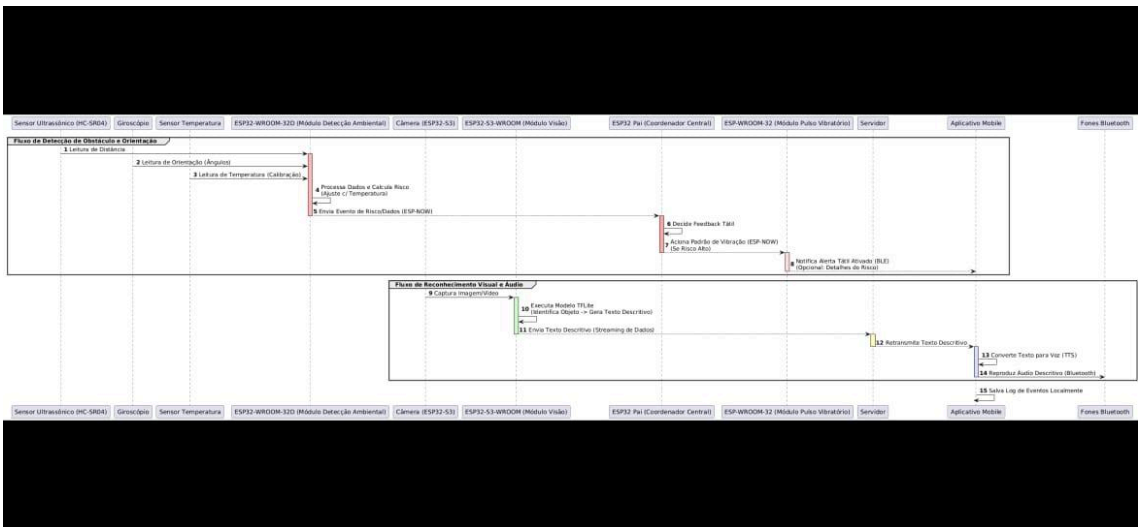
#### Descrição textual das principais classes:

- **Sensor**: armazena dados de medição de distância.
- **Atuador**: controla intensidade de vibração.
- **Câmera**: captura e envia frames para inferência.
- **ModeloIA**: executa classificação TFLite.
- **AppController**: gerencia BLE, calibração e TTS.
- **Eventos** registro de alertas (timestamp, tipo, confiança).

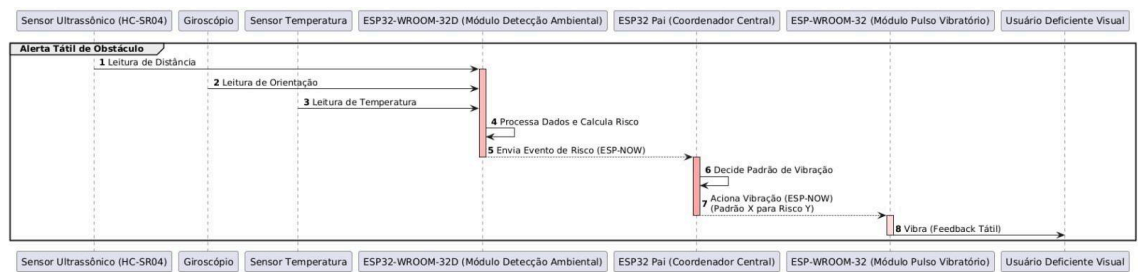
- **LogManager**: salva e exporta logs.

## 6. Modelo de Interação

### 6.1 Diagrama de Sequência – Fluxo de Detecção e Alerta



(Imagem – Diagrama de Sequência)



(Imagem – Diagrama de Sequência(específico))

### **Fluxo Descrito:**

1. Sensor detecta obstáculo.
2. ESP32 processa a distância.
3. Se limiar ultrapassado, aciona vibração.
4. Envia evento ao App via BLE.
5. App converte evento em mensagem TTS.
6. Log é salvo localmente.

## **7. Requisitos Funcionais e Não Funcionais (Resumo)**

### **7.1 Requisitos Funcionais**

Principais requisitos já definidos no DR:

- RF01 – Medição de Distância
- RF02 – Alerta Tátil
- RF03 – Captura de Imagem
- RF04 – Reconhecimento de Objetos
- RF05 – Comunicação BLE
- RF06 – Alerta TTS
- RF07 – Registro de Logs
- RF10 – Ajuste de Parâmetros
- RF15 – Exportação de Logs

### **7.2 Requisitos Não Funcionais**

- **Latência:**  $\leq 400$  ms
- **Acurácia:**  $\geq 70\%$
- **Autonomia:**  $\geq 2$  horas
- **Peso:**  $\leq 200$  g
- **Usabilidade:** padrões de vibração distinguíveis
- **Segurança:** logs acessíveis apenas no app autenticado

## **8. Glossário**

<b>Termo</b>	<b>Definição</b>
<b>TTS (Text-to-Speech)</b>	Recurso de voz do app que lê mensagens.
<b>BLE (Bluetooth Low Energy)</b>	Protocolo de comunicação entre o óculos e o app.
<b>ESP32 / ESP32-CAM</b>	Microcontroladores usados no protótipo.
<b>TFLite (TensorFlow Lite)</b>	Framework de IA leve para inferência embarcada.
<b>PoC (Proof of Concept)</b>	Prova de conceito – protótipo experimental de curta duração.
<b>Vibracall</b>	Atuador tátil de vibração usado no wearable.

## 10. Referências

- Documentação Técnica ESP32 (Espressif, 2024).
- TensorFlow Lite Micro Guide.
- Android BLE API Reference.
- Diretrizes de Acessibilidade W3C / WCAG 2.1.