

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 ≤ 400 ms.
- Acurácia de reconhecimento $\geq 70\%$ (ambiente controlado).
- Autonomia mínima de 2 horas.
- Peso máximo do conjunto ≤ 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.

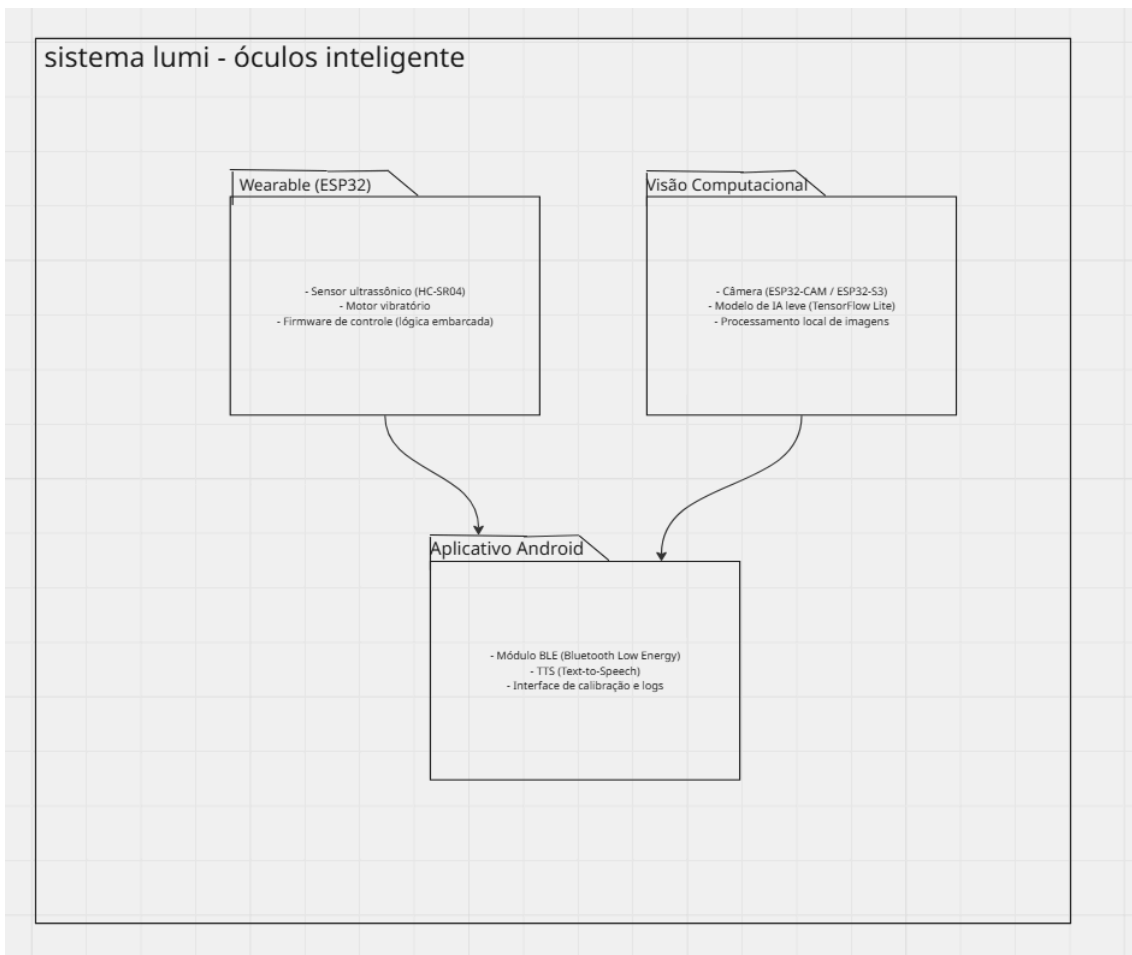
3. Modelagem de Subsistemas

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
Sensoriamento Tátil	Realiza medições de distância e aciona vibração conforme proximidade.	Sensor HC-SR04, motor vibratório, ESP32.
Visão Computacional	Reconhece objetos e classes prioritárias (fio, degrau, pessoa, porta, obstáculo).	Câmera ESP32-CAM / modelo TFLite.
Aplicativo Android	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.

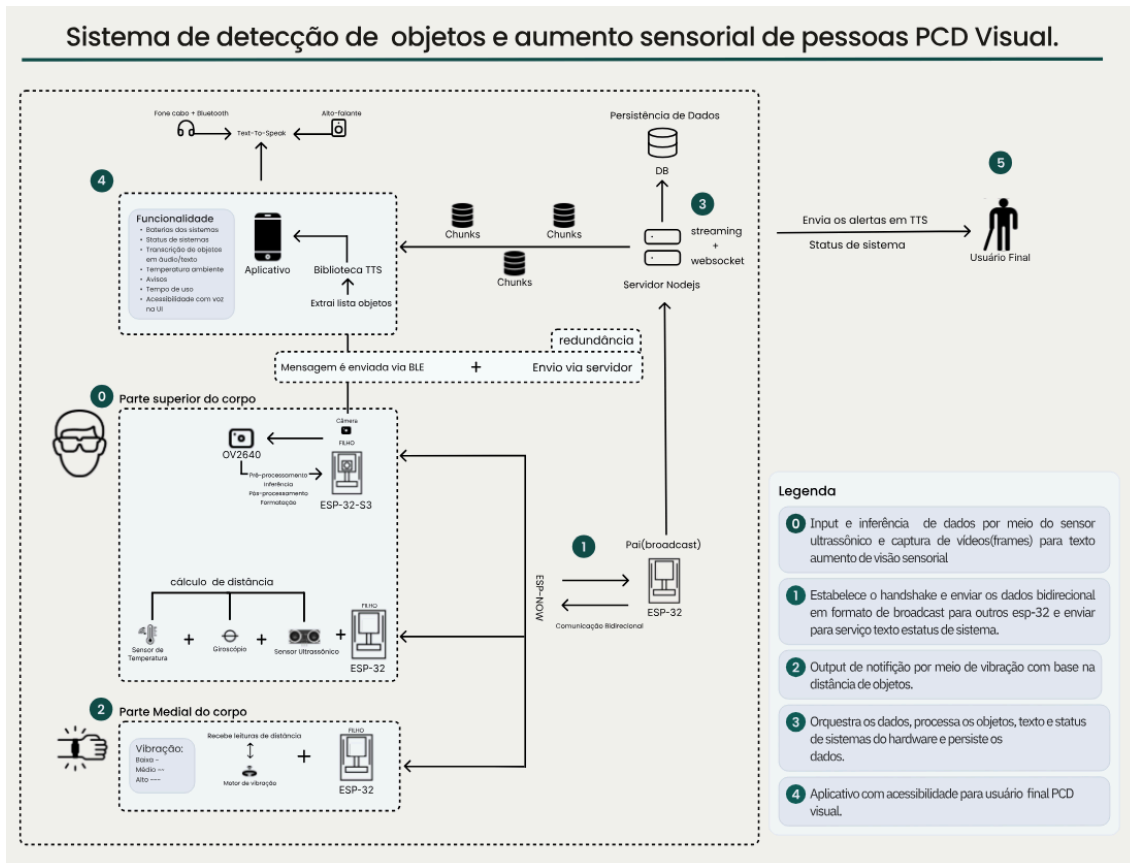
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio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



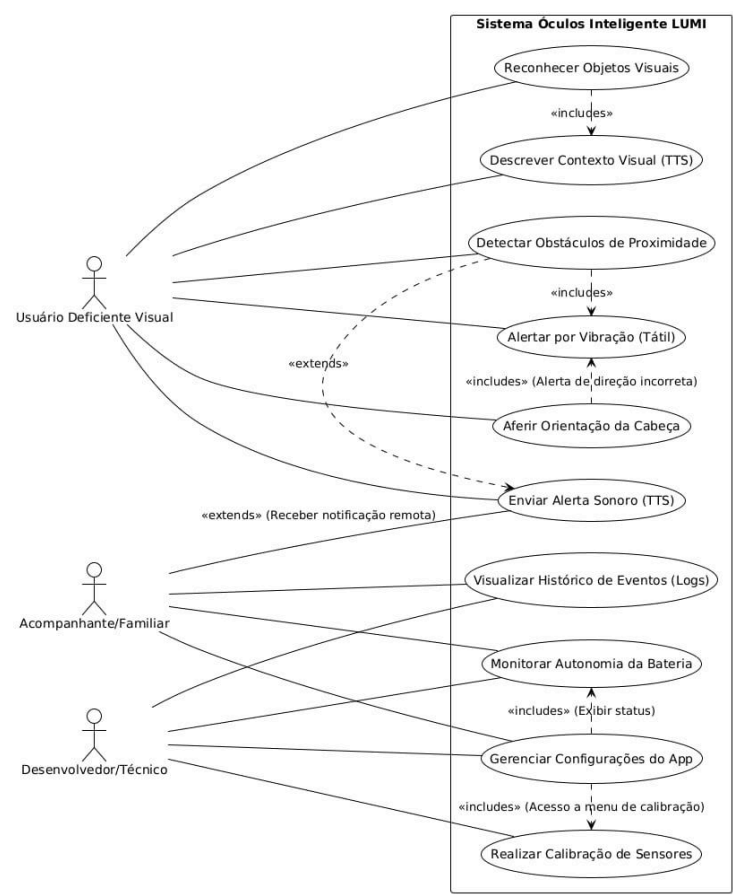
(Imagem – Diagrama de Visual)

4. Modelagem de Casos de Uso

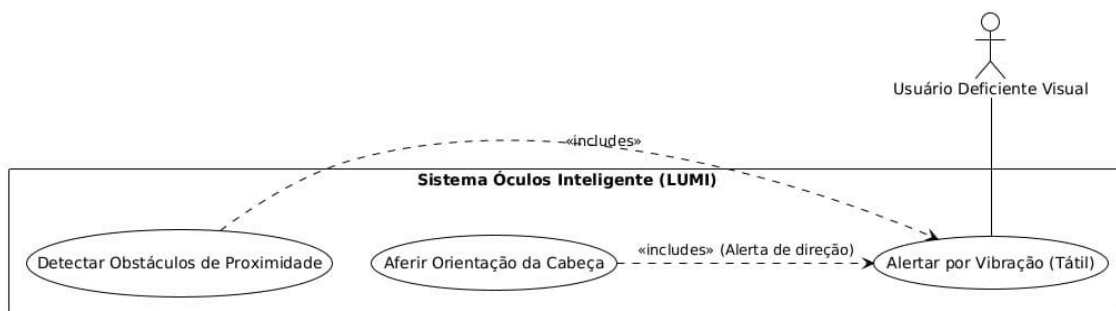
4.1 Atores

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)



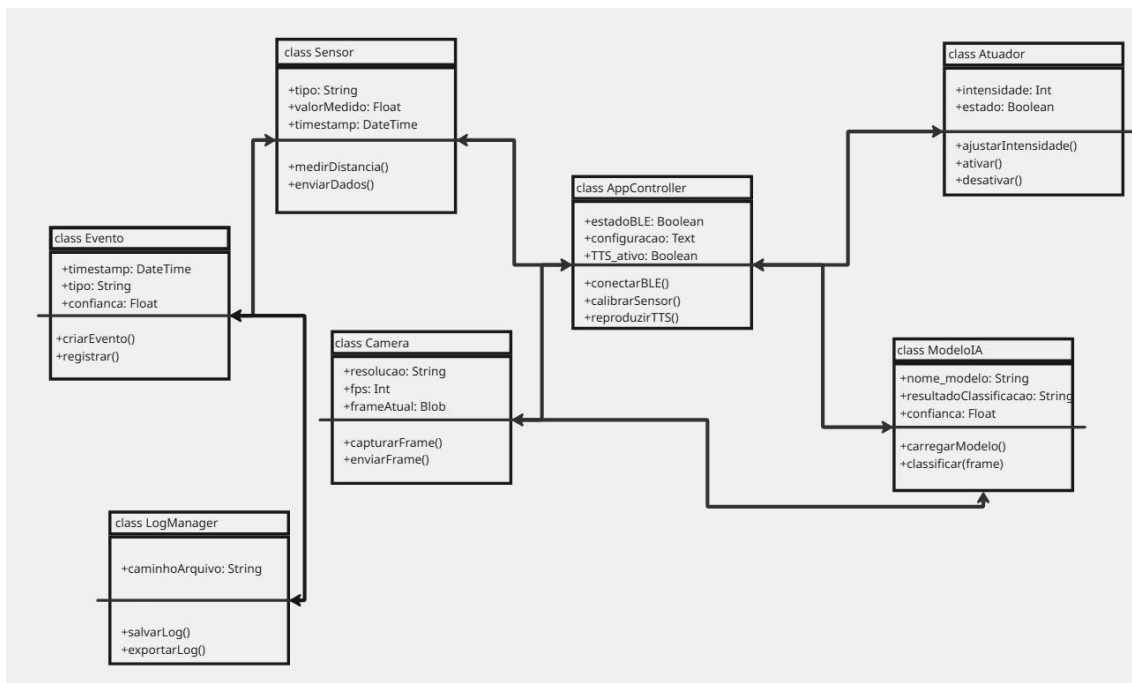
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

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)

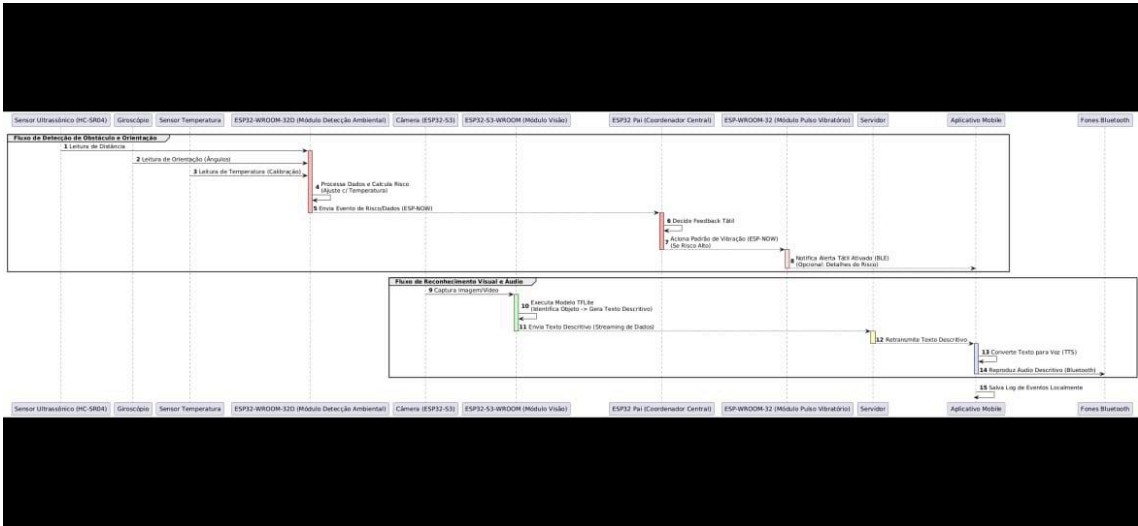
Descrição textual das principais classes:

- **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).

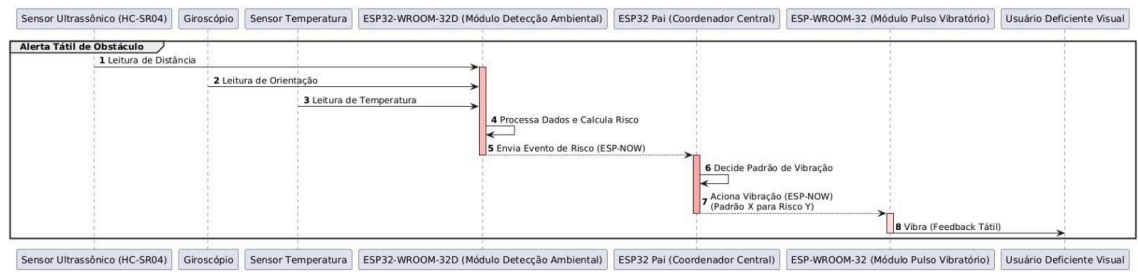
- **LogManager**: salva e exporta logs.

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

8. Glossário

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