

CHECKLIST ERS – DOCUMENTO COMPLETO

Cada item contém um emoji indicando que a seção está presente e validada.

1. Identificação do Documento

2. Resumo e Abstract

3. Introdução

4. Escopo do Sistema

5. Minimundo

6. Requisitos Funcionais

- Medição de distância
- Alerta tátil
- Captura de imagem
- Reconhecimento de objetos
- Comunicação BLE
- TTS no app
- Logs, parâmetros, modo silêncio, exportação, auto-teste

7. Regras de Negócio

- Priorização por distância
- Vibração por faixa
- Economia de energia- Fallback e armazenamento

8. Requisitos Não Funcionais

- Latência
- Acurácia
- Autonomia
- Usabilidade
- Segurança
- Portabilidade
- Peso

9. Hardware e Arquitetura

- Sensores ultrassônicos HC-SR04
- Atuador vibratório (vibracall)
- ESP32 / ESP32-CAM / ESP32-S3
- Bateria 3.7–5V com proteção
- Estrutura wearable (óculos/boné)
- Comunicação BLE e Wi-Fi opcional

- Módulos independentes (distância, visão, app)
- Requisitos de consumo, peso e autonomia

10. Casos de Uso 

- Detectar obstáculo
- Identificar objeto
- Emitir TTS
- Calibrar sistema- Exportar logs

11. Modelo Estrutural (Classes) 

12. Modelo de Interação (Sequência) 

13. Modelagem de Subsistemas 

- Sensoriamento tátil
- Visão computacional
- Aplicativo mobile

14. Glossário 

15. Referências 

Documento validado e organizado.

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

RF03	Captura de Imagem: O sistema deve capturar imagens a cada X segundos (configurável) e disponibilizar para inferência local ou envio ao app quando solicitado.	Alta	RF02
RF04	Reconhecimento de Objetos (5 classes prioritárias): 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	Comunicação com App (BLE): Enviar eventos (tipo, distância, timestamp, confidence) via BLE para app Android e receber comandos de calibração.	Alta	RF04
RF06	TTS no App: O app deve reproduzir mensagem curta correspondente ao evento (ex.: “fio à frente, 1,2 metros”) em ≤ 400 ms após recebimento do evento	Média	RF05
RF07	Registro/Logs: Registrar localmente eventos com campos: timestamp, tipo, distância, confidence, bateria.	Alta	RF03
RF08	Gestos/Botões de Controle: Permitir ligar/desligar alertas via botão físico no wearable.	Médio	
RF09	Fallback de processamento: Caso a inferência local não alcance metas de latência ou acurácia, permitir offload da imagem para o smartphone para inferência.	Médio	
RF10	Atualização de Parâmetros via App: Permitir ajustes em thresholds de distância, sensibilidade de vibração, taxa de captura de imagem e volume de TTS pelo app.	Médio	
RF11	Verificar dispositivos pelo App: 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	Alerta de Bateria Baixa: 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	Confirmação de Inicialização (Auto-Teste): 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	Modo de Silêncio Temporário: O usuário deve poder pausar (silenciar) temporariamente todos os alertas (tátteis e sonoros) através de um comando no wearable (ex: um clique duplo no botão RF08) ou no App (via RF10).	Média	RF08, RF10
RF15	Exportação de Logs de Teste: 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 ≤ 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 ≤ 400 ms; vibração acionada em ≤ 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 \leq 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 ± 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)		Imagen 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.

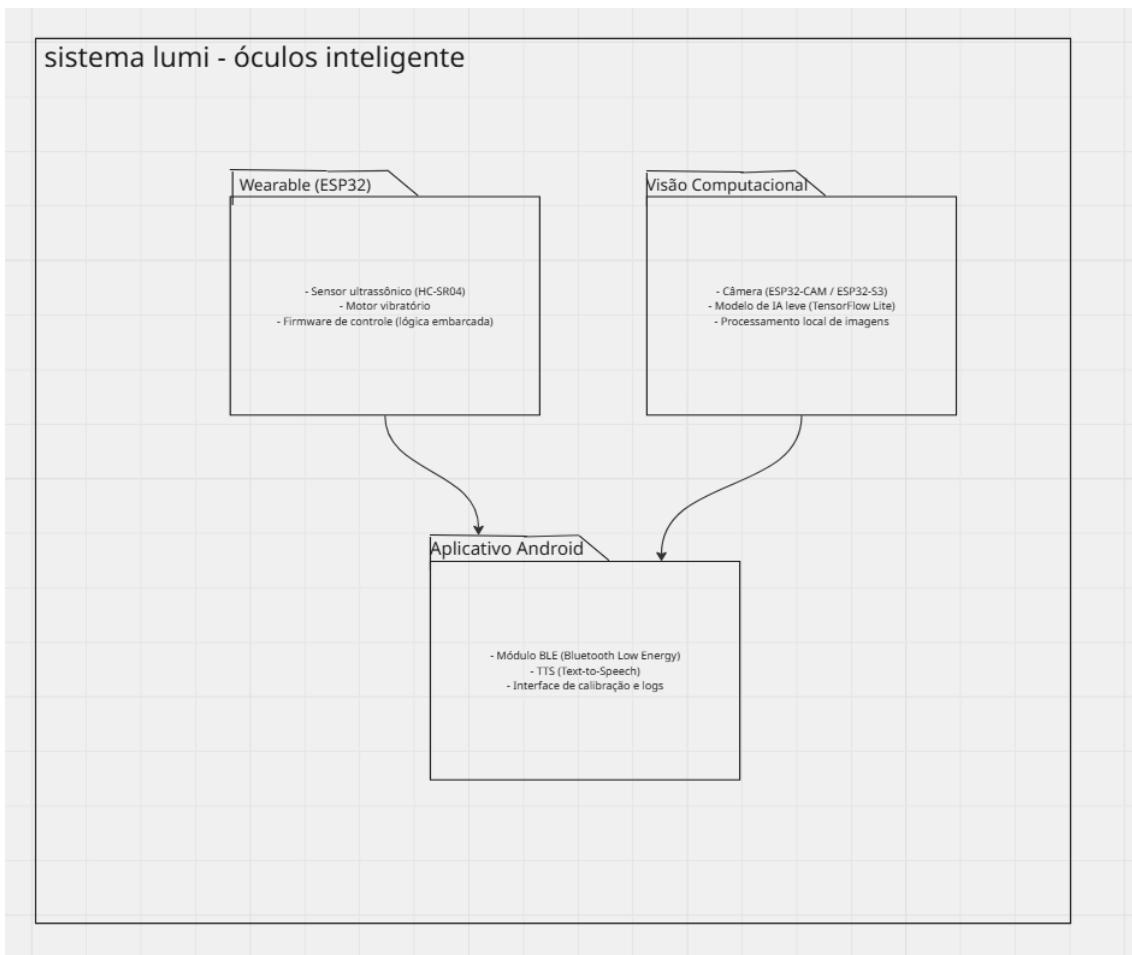
3. Modelagem de Subsistemas

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
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.

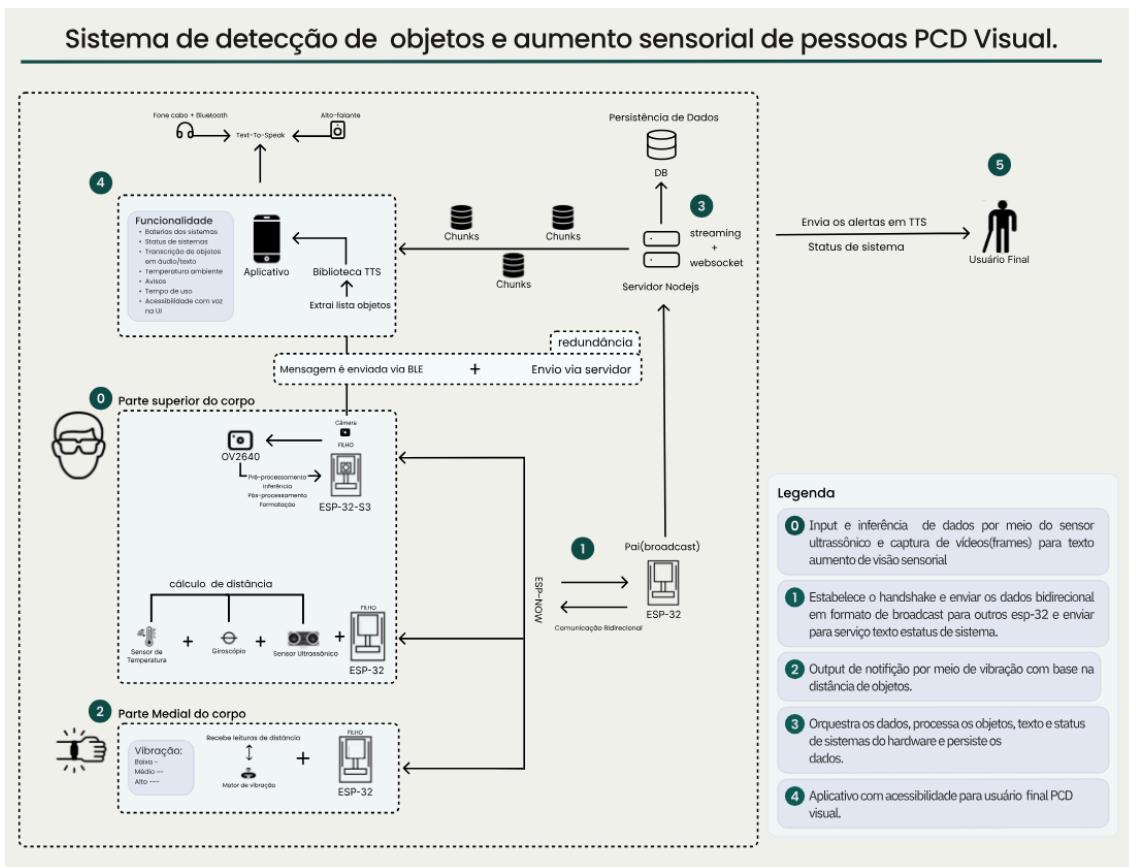
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio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



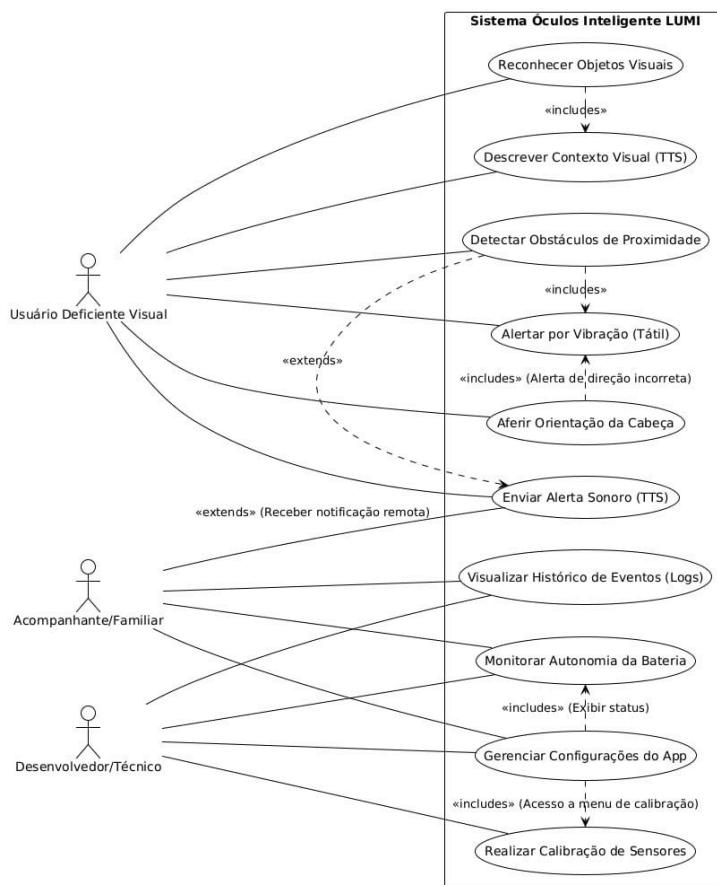
(Imagen – Diagrama de Visual)

4. Modelagem de Casos de Uso

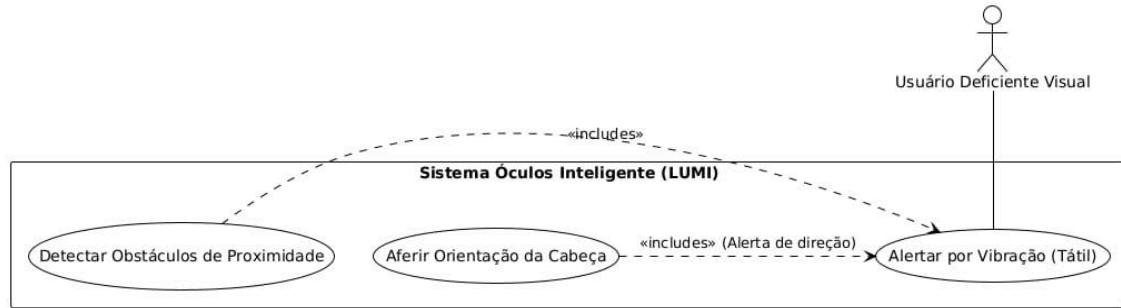
4.1 Atores

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)



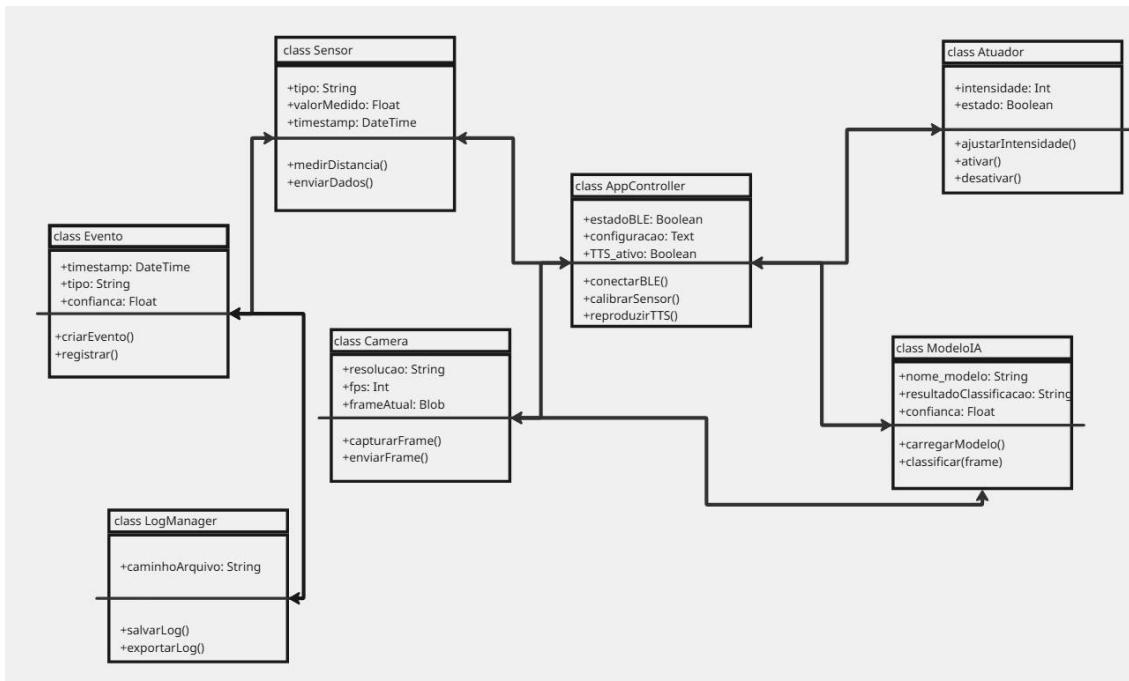
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

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)

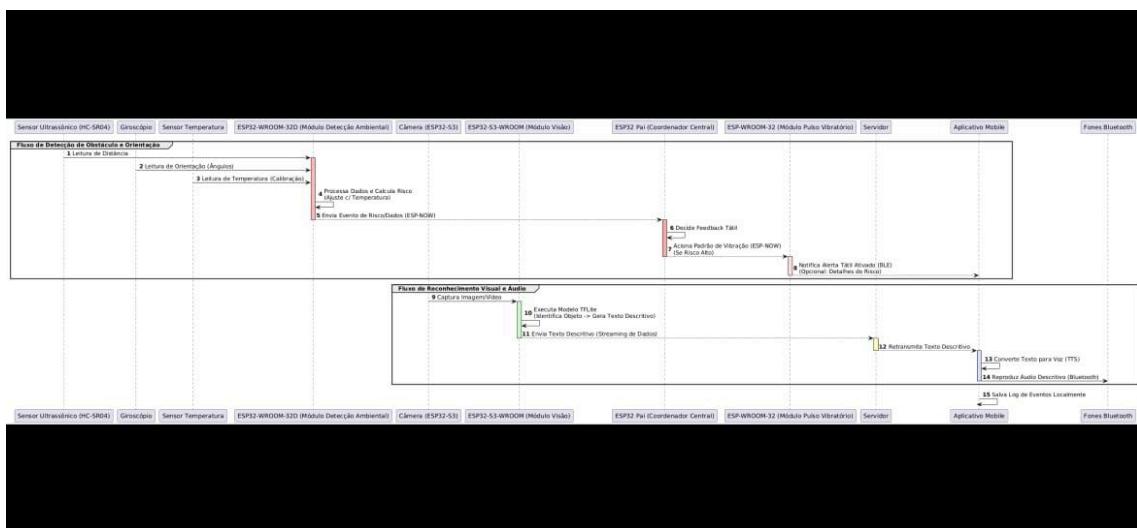
Descrição textual das principais classes:

- **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).

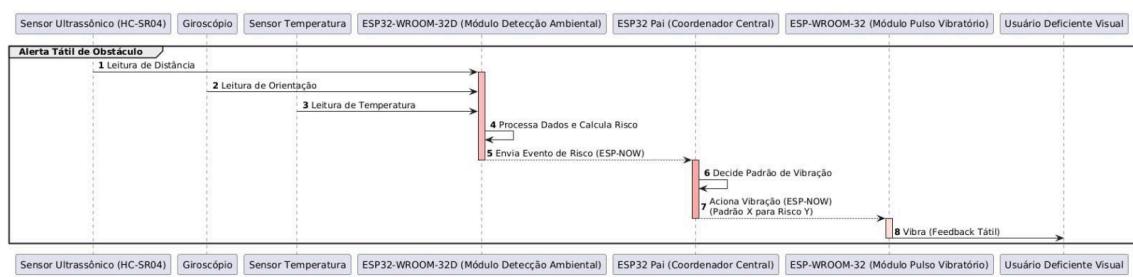
- **LogManager:** salva e exporta logs.

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:** ≥ 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.

MANUAL DA EQUIPE – PROJETO ÓCULOS INTELIGENTE

1. Objetivo do Manual

Este manual define **funções, responsabilidades, regras internas, fluxos de trabalho e procedimentos padrão** para a equipe do projeto “Óculos Inteligente de Baixo Custo”.

Serve como referência para manter **organização, alinhamento e consistência** durante todo o ciclo de desenvolvimento.

2. Estrutura da Equipe e Papéis

Product Owner – Fabio Brasileiro

- Define prioridades do backlog.
- Aprova critérios de aceitação.
- Alinha requisitos com stakeholders.

Scrum Master – Heloísa Cativo

- Garante que o time siga Scrum.
- Remove impedimentos.
- Facilita reuniões e boa comunicação.

Tech Lead / Arquiteto – José Filho

- Define arquitetura do sistema.
- Aprova decisões técnicas.
- Orienta devs sobre boas práticas.

Desenvolvedores Backend – Victor Fernandes, Victor Abreu, Guilherme Reis

- Implementam lógica embarcada e APIs.
- Integram sensores, IA e BLE.
- Criam testes de unidade.

Desenvolvedores Frontend/Android – Luis Oliveira, Eric Silva, João Pereira

- Desenvolvem app Android.
- Criam telas, acessibilidade e TTS.
- Integram BLE e calibradores do sistema.

QA / Testador – Mateus Miranda

- Cria planos e casos de testes.
- Valida módulos e sprint releases.
- Registra bugs e acompanha correções.

Config Manager – José Filho

- Cuida de versionamento (Git).
- Organiza branches e releases.
- Garante integridade do repositório.

CCB – Victor Abreu

- Aprova solicitações de mudança.
- Avalia impacto técnico e de cronograma.

3. Regras Internas da Equipe

3.1 Comunicação

- Canal principal: **WhatsApp + Discord**
- Reuniões:
 - **Daily:** 15 min
 - **Revisão de Sprint:** ao final de cada sprint
 - **Retrospectiva:** após review
- Chamadas urgentes: marcar @todos no Discord.

3.2 Regras de Commit (Git)

Sempre seguir o padrão:

`type: descrição curta`

- `o que foi feito`
- `motivo da mudança`

Types permitidos:

`feat, fix, docs, test, refactor, perf, style`

Branch pattern:

- `main` → versão estável
- `develop` → versão de integração
- `feature/nome` → novas funções
- `hotfix/nome` → correções urgentes

3.3 Boas Práticas de Código

- Nome de variáveis claro e objetivo.
- Um commit por tarefa.
- Evitar código duplicado.
- Sempre documentar funções principais.
- Criar testes unitários onde possível.

4. Procedimentos Operacionais

4.1 Desenvolvimento de Hardware

- Validar sensores antes de integrar.
- Testar HC-SR04 em bancada com régua.
- Testar vibracall com PWM em três níveis.
- Testar câmera com snapshot antes da IA.
- Registrar consumo da bateria a cada iteração.

4.2 Desenvolvimento do Firmware (ESP32)

Fluxo padrão:

1. Implementar leitura de sensores.
2. Garantir tratamento de erros.
3. Enviar eventos via BLE.
4. Testar latência e estabilidade.

5. Integrar com app Android.

Checklist interno:

- Distância
- Vibração proporcional
- Captura de imagem
- Inferência TFLite
- BLE

4.3 Desenvolvimento do App Android

- Criar telas simples e acessíveis.
- Implementar TTS responsivo.
- Garantir sincronização BLE.
- Incluir calibração (thresholds e vibração).
- Feedback por voz e botões grandes.

4.4 Testes

QA deve validar:

- Funcionalidade dos sensores
- Reconhecimento de objetos
- Latência ponta-a-ponta
- Uso com 1–5 usuários reais
- Exportação de logs
- Estabilidade da conexão BLE

5. Organização do Repositório

```
/docs
  ERS/
  diagramas/

/hardware
  esp32/
  esquemáticos/

/app
  android app/

/firmware
  modules/
  ble/
  vision/

/tests
  unit/
  integration/
```

6. Solicitações de Mudança (CCB)

Qualquer alteração no sistema deve seguir:

1. Abrir issue: “Solicitação de Mudança”.
2. CCB avalia impacto.
3. Decisão registrada no GitHub.
4. Implementação inicia somente após aprovação.

7. Fluxo de Sprint (Scrum)

Duração: 7 dias

Dia 1 – Planejamento

Definição do backlog da sprint.

Dias 2–6 – Execução

Desenvolvimento, testes e integração.

Dia 7 – Review + Retrospectiva

8. Regras de Qualidade

- Latência \leq 400 ms
- Acurácia \geq 70%
- Autonomia \geq 2 horas
- BLE estável
- Vibração perceptível
- Logs funcionando

9. Checklist Final da Sprint (interno)

- Código testado
- Testes feitos pelo QA
- Documentação atualizada

- Bugs corrigidos
- Branch mergeada corretamente
- Versão registrada



10. Contatos e Papéis Rápidos

- **PO:** decisões de produto
- **SM:** processos ágeis
- **Tech Lead:** decisões técnicas
- **QA:** testes
- **Config Manager:** Git
- **CCB:** aprova mudanças

Guia Passo a Passo para o Desenvolvimento do Projeto "Óculos Inteligente de Baixo Custo"

Este documento apresenta um guia passo a passo detalhado para a execução do projeto "Óculos Inteligente de Baixo Custo para Auxílio à Locomoção de Pessoas com Deficiência Visual", conforme o **Modelo de Ciclo de Vida Incremental com Sprints Semanais** definido no Documento de Análise de Requisitos. O projeto tem uma duração total de 6 semanas e visa a entrega de um Protótipo de Prova de Conceito (PoC) funcional.

1. Estrutura de Governança e Papéis

A execução do projeto é baseada em uma estrutura de equipe ágil, com papéis e responsabilidades claramente definidos, conforme o organograma e a matriz de responsabilidades do documento de origem.

Papel	Responsável	Foco Principal
Product Owner (PO)	Fabio Brasileiro	Maximizar o valor do produto, gerenciar e priorizar o <i>Backlog</i> de funcionalidades.
Scrum Master (SM)	Heloisa Cativo	Garantir a adesão ao processo <i>Scrum</i> , remover impedimentos e facilitar as reuniões.
Tech Lead / Arquiteto	Jose Filho	Definir a arquitetura do sistema (Módulos 1, 2 e 3), garantir a integridade técnica e orientar os desenvolvedores.
Desenvolvedores (Devs)	Luis Oliveira, Eric Silva, João Pereira, Victor Fernandes, Victor Abreu, Guilherme Reis	Implementação do <i>firmware</i> embarcado, desenvolvimento do aplicativo Android e integração dos módulos.
QA/Testador	Mateus Miranda	Planejar e executar testes funcionais, de usabilidade e de desempenho (latência e acurácia).

2. Plano de Execução Incremental (6 Sprints)

O desenvolvimento é dividido em seis *sprints* semanais, cada uma focada na entrega de um incremento funcional do sistema. O objetivo é validar hipóteses de negócio e incorporar *feedback* de forma contínua.

Sprint 1: Módulo Tátil e Base de Hardware

Foco: Estabelecer a base de *hardware* e a funcionalidade primária de detecção de distância e *feedback* tátil.

Atividade	Requisitos Chave	Produto de Entrega
1.1. Implementação do Módulo 1	RF01 (Medição de Distância), RF02 (Alerta Tátil Proporcional)	Protótipo funcional do Módulo 1 (ESP32/Pico + HC-SR04 + Vibracall) capaz de medir distância e acionar vibração proporcional em ≤ 100 ms.
1.2. Auto-Teste de Inicialização	RF13 (Confirmação de Inicialização)	Rotina de <i>firmware</i> para auto-teste de sensores (HC-SR04) e emissão de sinal de "Sistema Pronto" (vibração longa/curta).
1.3. Monitoramento de Energia	RF12 (Alerta de Bateria Baixa)	Implementação do circuito de monitoramento de bateria e lógica de alerta crítico.

Sprint 2: Módulo de Visão e Coleta de Dados

Foco: Integrar o subsistema de visão computacional e a capacidade de registro de eventos (*logging*).

Atividade	Requisitos Chave	Produto de Entrega
2.1. Implementação do Módulo 2	RF03 (Captura de Imagem)	Módulo 2 (ESP32-CAM/S3) configurado para captura periódica de imagens.
2.2. Estrutura de Logs	RF07 (Registro/Logs)	Implementação da estrutura de dados para registro local de eventos (<i>timestamp</i> , tipo, distância, <i>confidence</i>).
2.3. Treinamento do Modelo	RF04 (Reconhecimento de Objetos)	Início do treinamento e otimização do modelo TFLite para as 5 classes prioritárias (fio/ramo, degrau, cadeira/obstáculo baixo, pessoa, porta).

Sprint 3: Comunicação e Aplicativo Mínimo

Foco: Estabelecer a comunicação sem fio entre o *wearable* e o *smartphone* e desenvolver a interface básica do usuário.

Atividade	Requisitos Chave	Produto de Entrega
3.1. Conexão BLE	RF05 (Comunicação com App - BLE)	Conexão BLE estável para envio de eventos e recebimento de comandos de calibração.

Atividade	Requisitos Chave	Produto de Entrega
3.2. Interface de Status	RF11 (Verificar dispositivos pelo App)	Aplicativo Android mínimo exibindo status de conexão, nível de bateria e status operacional dos sensores.
3.3. Testes de Latência	RNF01 (Latência)	Testes de ponta a ponta para garantir que a latência de comunicação esteja dentro do limite de 400 ms.

Sprint 4: Inteligência e Feedback Sonoro (TTS)

Foco: Integrar o modelo de reconhecimento e o sistema de alerta de voz, completando o ciclo de *feedback* multimodal.

Atividade	Requisitos Chave	Produto de Entrega
4.1. Integração do Reconhecimento	RF04 (Reconhecimento de Objetos)	Modelo TFLite integrado ao <i>firmware</i> (ou App, via <i>offload</i>) e funcional, com acurácia alvo de 70%.
4.2. Alerta de Voz (TTS)	RF06 (TTS no App)	Implementação da funcionalidade de <i>Text-to-Speech</i> (TTS) no App para reproduzir alertas verbais curtos.
4.3. Regras de Negócio	RN01, RN02, RN03, RN04	Implementação da lógica de priorização de alertas (distância vs. perigo suspenso) e intensidade de vibração.

Sprint 5: Usabilidade e Robustez

Foco: Refinar a experiência do usuário, adicionando controles de usabilidade e mecanismos de robustez.

Atividade	Requisitos Chave	Produto de Entrega
5.1. Controles de Usuário	RF08 (Gestos/Botões de Controle), RF14 (Modo de Silêncio Temporário)	Implementação de botão físico no <i>wearable</i> e funcionalidade no App para pausar temporariamente os alertas.
5.2. Interface de Calibração	RF10 (Atualização de Parâmetros via App), RNF04 (Interface de calibração)	Interface no App para ajustes de <i>thresholds</i> de distância, sensibilidade de vibração e volume de TTS.
5.3. Fallback de Processamento	RF09 (Fallback de processamento)	Lógica de <i>offload</i> da imagem para o <i>smartphone</i> para inferência, caso a latência local seja comprometida.

Sprint 6: Validação, Implantação e Entrega

Foco: Testes finais de campo, validação dos critérios de aceitação e preparação da entrega do PoC.

Atividade	Requisitos Chave	Produto de Entrega
6.1. Testes de Campo com Usuários	RNF03 (Autonomia), RNF02 (Acurácia)	Relatório de testes de campo com 1-5 usuários, validando autonomia (≥ 2 horas) e acurácia em ambiente real.
6.2. Exportação de Logs	RF15 (Exportação de Logs de Teste)	Funcionalidade no App para exportar logs de eventos em formato CSV ou JSON para análise <i>offline</i> .
6.3. Implantação e Documentação	-	Versão Candidata (RC) finalizada e documentação técnica mínima do PoC.

3. Critérios de Aceitação do PoC

O sucesso do projeto será medido pela aderência aos seguintes critérios de aceitação, a serem validados na Sprint 6:

Critério	Meta	Requisito Relacionado
Acurácia	$\geq 70\%$ nas 5 classes prioritárias em ambiente controlado.	RF04, RNF02
Latência	Média de alerta (detecção \rightarrow feedback) ≤ 400 ms.	RNF01
Autonomia	Duração mínima de prova ≥ 2 horas com ciclo de uso típico.	RNF03
Validação	Testes preliminares realizados com 1-5 usuários.	-
Funcionalidade	Integração estável e funcional dos Módulos 1, 2 e 3.	RF05, RF11

1. Plano de projeto

Projeto:

Óculos inteligente de baixo custo que detecta perigos ambientais e fornece alertas táticos 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 desenvolver 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, Guilherme Reis
- 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: Total de funcionalidades concluídas / 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.

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 tátteis 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 ≤ 400 ms, autonomia de prova ≥ 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 ≤ 400 ms, a test battery life of ≥ 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 tátteis (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 táticos 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 desenvolver 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, Guilherme Reis
- 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: Total de funcionalidades concluídas / 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	Medição de Distância: O sistema deve medir distância em frente ao usuário usando HC-SR04 e disponibilizar leitura atual a cada 200–500 ms.	Alta	-
RF02	Alerta Tátil Proporcional: Ao detectar distância abaixo de thresholds configuráveis, o dispositivo deve acionar o vibracall com intensidade proporcional (fraco/médio/alto) em ≤ 100 ms após leitura.	Alta	RF01

RF03	Captura de Imagem: O sistema deve capturar imagens a cada X segundos (configurável) e disponibilizar para inferência local ou envio ao app quando solicitado.	Alta	RF02
RF04	Reconhecimento de Objetos (5 classes prioritárias): 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	Comunicação com App (BLE): Enviar eventos (tipo, distância, timestamp, confidence) via BLE para app Android e receber comandos de calibração.	Alta	RF04
RF06	TTS no App: O app deve reproduzir mensagem curta correspondente ao evento (ex.: “fio à frente, 1,2 metros”) em ≤ 400 ms após recebimento do evento	Média	RF05
RF07	Registro/Logs: Registrar localmente eventos com campos: timestamp, tipo, distância, confidence, bateria.	Alta	RF03
RF08	Gestos/Botões de Controle: Permitir ligar/desligar alertas via botão físico no wearable.	Médio	
RF09	Fallback de processamento: Caso a inferência local não alcance metas de latência ou acurácia, permitir offload da imagem para o smartphone para inferência.	Médio	
RF10	Atualização de Parâmetros via App: Permitir ajustes em thresholds de distância, sensibilidade de vibração, taxa de captura de imagem e volume de TTS pelo app.	Médio	
RF11	Verificar dispositivos pelo App: 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	Alerta de Bateria Baixa: 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	Confirmação de Inicialização (Auto-Teste): 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	Modo de Silêncio Temporário: O usuário deve poder pausar (silenciar) temporariamente todos os alertas (tátteis e sonoros) através de um comando no wearable (ex: um clique duplo no botão RF08) ou no App (via RF10).	Média	RF08, RF10
RF15	Exportação de Logs de Teste: 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 ≤ 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 ≤ 400 ms; vibração acionada em ≤ 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 \leq 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 ± 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)		Imagen 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 ≤ 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.

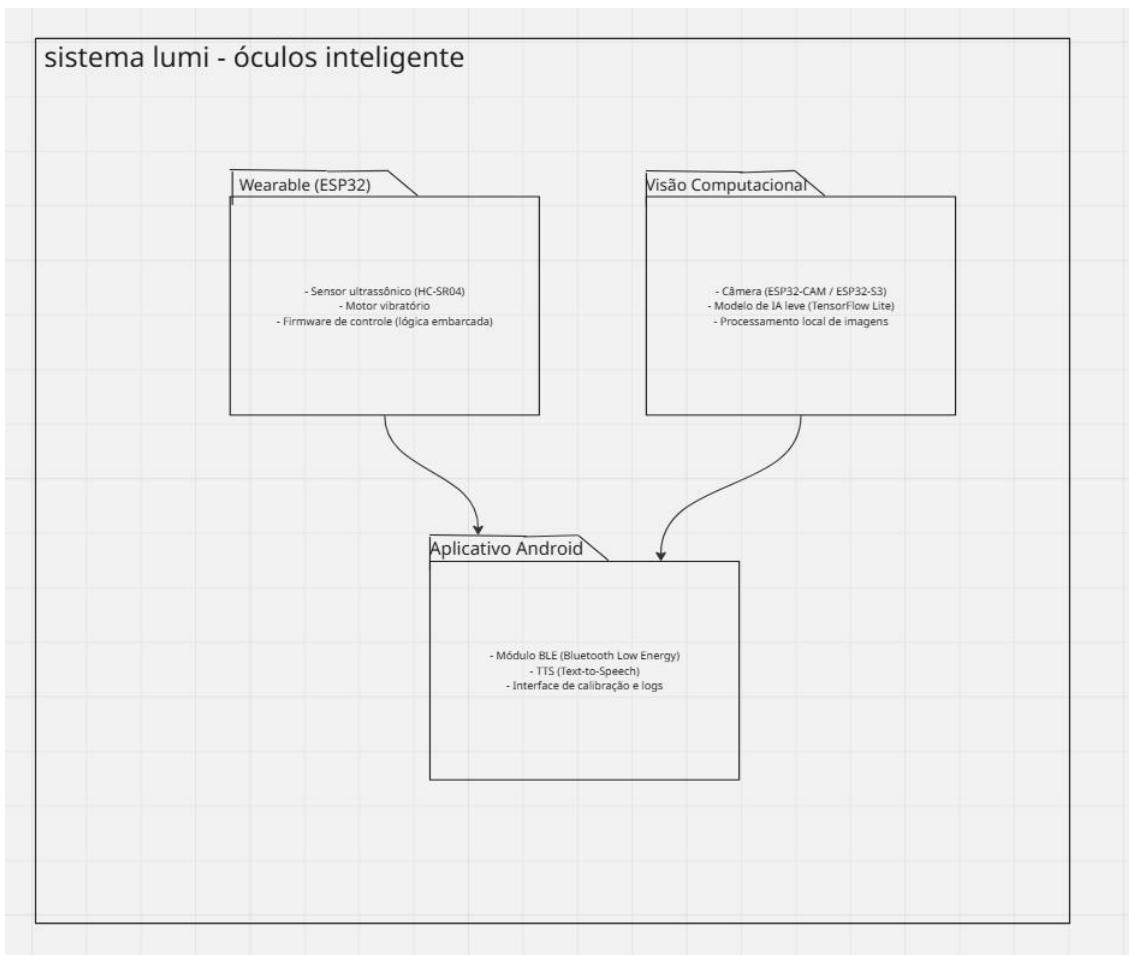
3. Modelagem de Subsistemas

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.

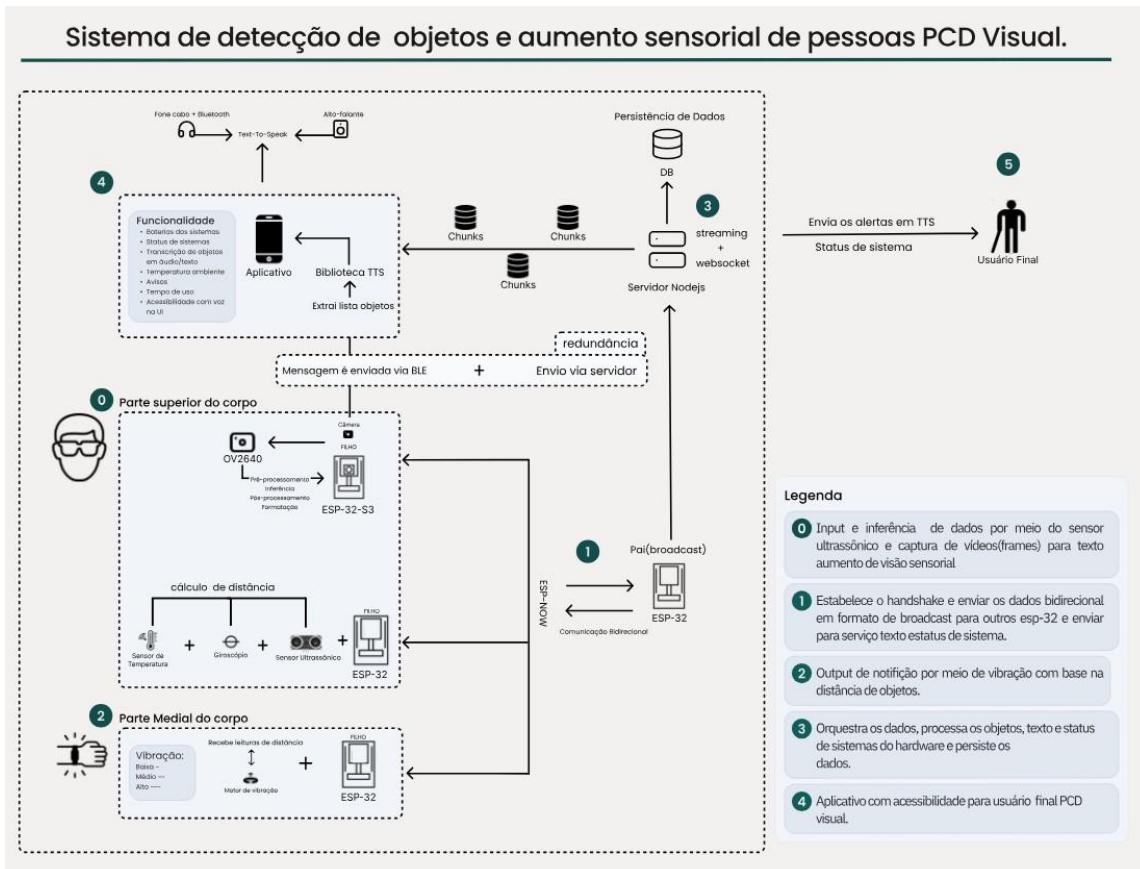
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio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



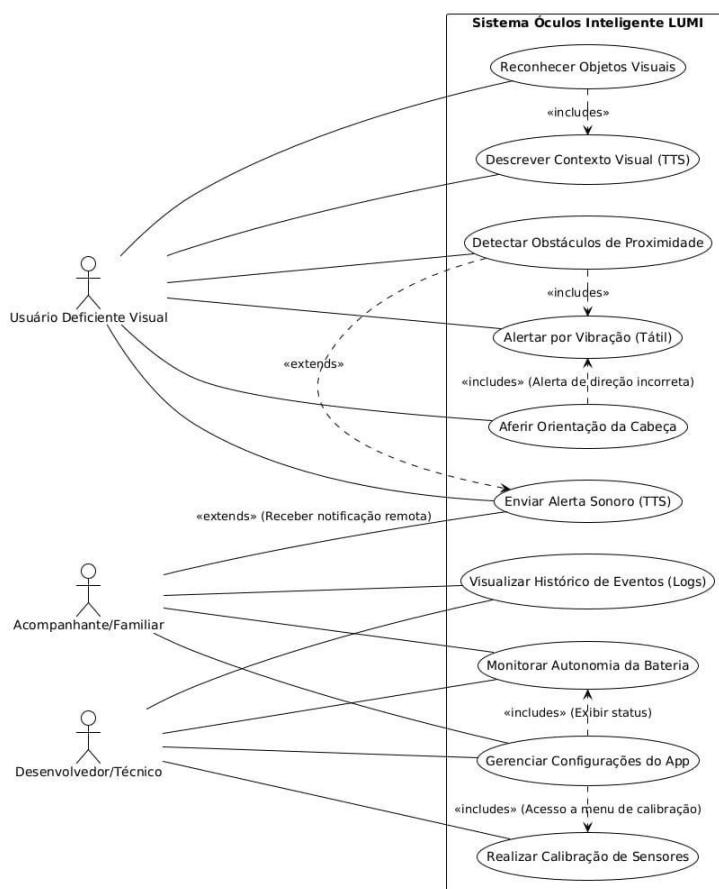
(Imagem – Diagrama de Visual)

4. Modelagem de Casos de Uso

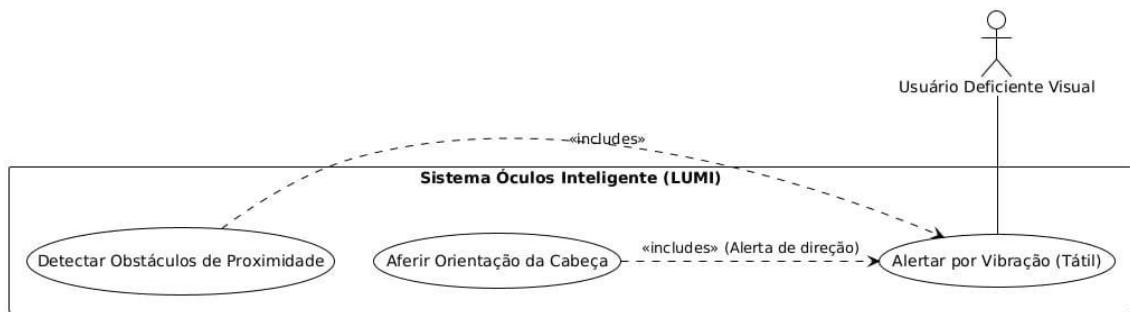
4.1 Atores

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)



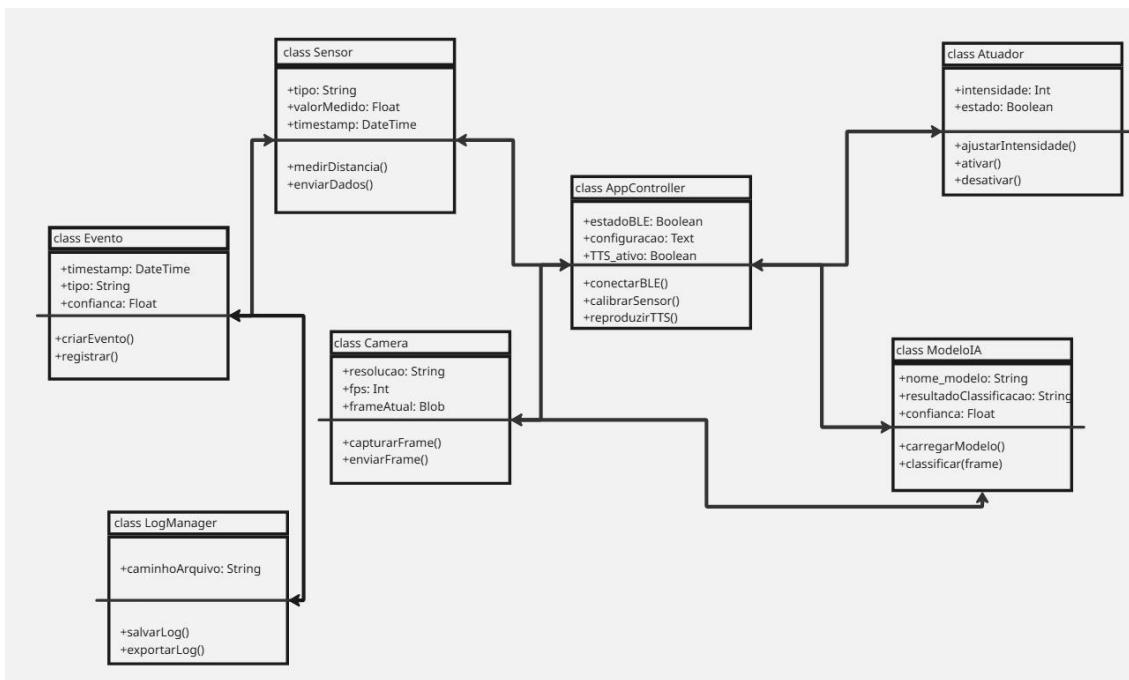
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

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)

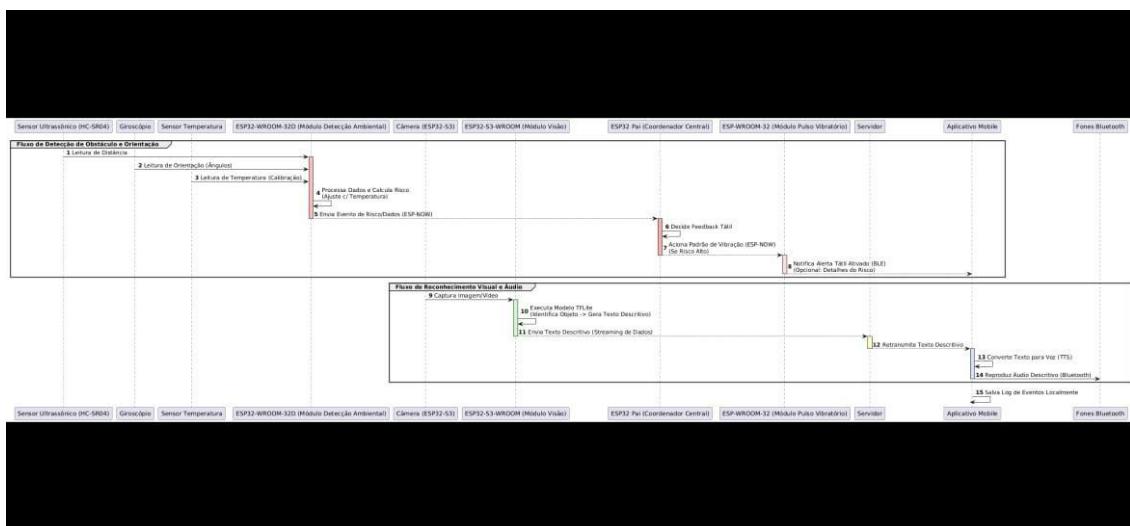
Descrição textual das principais classes:

- **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).

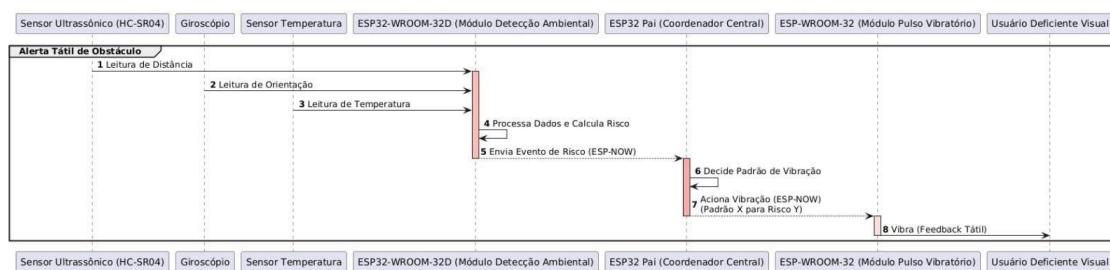
- **LogManager:** salva e exporta logs.

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:** ≥ 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.

ID	Tipo	Descrição do Requisito (Resumo)
RF01	Funcional	Medição de Distância (uso do HC-SR04 em 200 ms).
RF02	Funcional	Alerta Tátil Proporcional (vibração com intensidade proporcional à distância).
RF03	Funcional	Captura de Imagem (a cada 100 ms se a distância for crítica).
RF04	Funcional	Reconhecimento de Objetos (disponibilizar inferência local ou via App).
RF05	Funcional	Comunicação com App (transferência de dados, timestamp, confidence via BLE).
RF06	Funcional	TTS no App (mensagens de áudio correspondentes ao evento).
RF07	Funcional	Armazenamento Local (eventos/logs por 30 dias para análise offline).
RF08	Funcional	Controles por Hardware (botão liga/desliga, silêncio temporário, etc.).
RF09	Funcional	Falha de Processamento (notificar se a inferência local falhar).
RF10	Funcional	Atualização de Parâmetros via App (distância, sensibilidade, volume TTS).
RF11	Funcional	Verificar dispositivos pelo App (status de conexão/bateria).
RF12	Funcional	Alerta de Bateria (ativação de alerta sonoro/TTS quando atingir nível crítico).
RF13	Funcional	Confirmação de Inicialização (vibração + TTS "Sistema Ativo").
RF14	Funcional	Modo de Silêncio Temporário (silenciar temporariamente todos os alertas).
RF15	Funcional	Exportação de Logs de Teste (CSV/JSON para análise de desempenho offline).
RNF01	Não Funcional	Latência ponta-a-ponta (vibração < 400 ms).
RNF02	Não Funcional	Acurácia mínima de \$70\%\$ em ambiente controlado.
RNF03	Não Funcional	Autonomia mínima de \$\sim 4\$ horas de uso típico.

RNF04	Não Funcional	Interface de calibração acessível no App.
RNF Outros	Não Funcional	Disponibilidade: \$99.9\%\$ durante a sessão de uso.
RNF Outros	Não Funcional	Dados de usuários e logs armazenados localmente.

Caso de Uso Relacionado	Módulo / Componente
UC04 - Medir Distância	Módulo de Sensoriamento
UC02 - Emitir Alerta Tátil	Módulo de Feedback Tátil
UC07 - Capturar Imagem	Módulo de Captura
UC08 - Reconhecer Objeto	Módulo de Processamento
UC09 - Transmitir Dados	Módulo de Comunicação (BLE)
UC03 - Emitir Alerta Sonoro	Módulo de Áudio (App)
UC10 - Armazenar Log	Módulo de Armazenamento
UC11 - Controlar Dispositivo	Controlador Principal
UC12 - Notificar Falha	Módulo de Processamento
UC05 - Calibrar Sensores	Interface de Configuração (App)
UC13 - Monitorar Status	Módulo de Comunicação (App)
UC03 - Emitir Alerta Sonoro	Hardware / Energia
UC01 - Detectar Obstáculo	Módulo de Feedback Tátil
UC11 - Controlar Dispositivo	Controlador Principal
UC10 - Armazenar Log	Módulo de Comunicação (App)
—	Módulo de Sensoriamento
—	Módulo de Sensoriamento
—	Hardware / Energia

—	Interface de Configuração (App)
—	Confiabilidade
—	Segurança

Caso de Teste Relacionado	Prioridade	Status
CT01	Alta	Em Desenvolvimento
CT01, CT02, CT05	Alta	Em Desenvolvimento
CT03	Alta	Pendente
—	Alta	Pendente
CT03	Alta	Aprovado
—	Média	Pendente
—	Alta	Aprovado
—	Média	Em Desenvolvimento
—	Média	Pendente
—	Média	Pendente
—	Média	Aprovado
—	Alta	Em Análise
—	Alta	Em Desenvolvimento
—	Média	Aprovado
—	Média	Pendente
CT01 (Implícito)	Alta	Em Desenvolvimento
CT04 (Implícito)	Alta	Em Análise
CT07 (Implícito)	Alta	Em Análise

CT05 (Implícito)	Média	Pendente
—	Alta	Em Análise
—	Alta	Aprovado

RELATÓRIO DE CONTROLE E REGISTRO DE CONFIGURAÇÃO (STATUS ACCOUNTING REPORT)

Projeto: ÓCULOS INTELIGENTE DE BAIXO CUSTO (LUMI) **Versão do Documento:** 1.1 **Data:** 09/11/2025 **Aprovador:** Prof. Jean Carlos Figueiredo (CCB) **Revisão:** Equipe DEV – Engenharia de Software FAMETRO

1. Propósito

Este documento apresenta o Status Accounting (Controle e Registro de Configuração) do projeto LUMI, com o objetivo de garantir a rastreabilidade e visibilidade do progresso de artefatos de software e hardware durante o ciclo de vida do produto.

Inclui informações sobre:

- Itens de configuração (CIs) identificados;
- Status atual de cada item;
- Mudanças aprovadas pela Change Control Board (CCB);
- Versões liberadas e *baseline* vigente.

2. Itens de Configuração (Configuration Items – CIs)

ID	Tipo	Descrição	Versão	Repositório / Local	Status	Responsável
CI-SW-001	Documento	Documento de Requisitos (DR)	1.3	GitHub / docs/requirements	Aprovado	Mateus M. Miranda
CI-SW-002	Código-fonte	Firmware ESP32 (Sensores + Vibração)	1.1	GitHub / src/esp32_main	Em Teste (QA)	Victor Abreu
CI-SW-003	Código-fonte	Módulo IA (TFLite ESP32-CAM)	1.0	GitHub / src/vision_ai	Em Teste (QA)	Victor Fernandes
CI-SW-004	Aplicativo Android	App “LUMI Assist” (TTS + BLE + Logs)	1.0	GitHub / app/android_lumi	Em Teste (QA)	Joao Pereira
CI-HW-001	Hardware	Montagem do protótipo (sensor HC-SR04 + ESP32)	Rev. B	Laboratório FAMETRO	Integrado	Luis Oliveira
CI-TEST-001	Documento	Plano de Teste Funcional	1.1	docs/testing/PT-LUMI.pdf	Aprovado	Mateus M. Miranda
CI-MGMT-001	Documento	Plano de Gerenciamento de Configuração	1.1	docs/config/CM_PLAN_LUMI.pdf	Aprovado	José Filho
CI-DOC-001	Documento	Documento de Análise de Requisitos	1.4	docs/analysis	Aprovado	Heloisa Cativo

3. Baseline de Configuração Vigente

Tipo de Baseline	Data	Itens Incluídos	Observações
Functional Baseline (FB)	01/10/2025	CI-SW-001, CI-DOC-001	Requisitos e arquitetura aprovados pelo CCB.
Allocated Baseline (AB)	15/10/2025	CI-SW-002, CI-HW-001, CI-TEST-001	Integração parcial dos módulos sensoriais e app.
Product Baseline (PB)	09/11/2025	CI-SW-003, CI-SW-004, CI-SW-002, CI-TEST-001	Todos os artefatos de software e teste em fase de QA final para submissão à INOVATECH.

4. Controle de Mudanças (Change Requests – CRs)

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-001	PO	Commit Inicial	06/09/2025	Aprovada	Implementada	37abf1d
CR-002	PO	Adiciona a estrutura inicial (scaffold) do projeto backend e seu esquema de banco de dados.	20/09/2025	Aprovada	Implementada	ed8cab6
CR-003	PO	Adicionar código do sensor ultrassônico HCSR04 em C++.	12/10/2025	Aprovada	Implementada	1f8f21
CR-004	PO	Atualizar título do projeto no README.md.	12/10/2025	Aprovada	Implementada	fd18058
CR-005	PO	Remover código do sensor ultrassônico HCSR04 e adicionar configuração do projeto	12/10/2025	Aprovada	Implementada	2e4b8b9
CR-006	PO	Merge na branch main	12/10/2025	Aprovada	Implementada	f34276c
CR-007	PO	Adiciona script reset-project (script para reiniciar o projeto para inicializar a estrutura do projeto.	21/10/2025	Aprovada	Implementada	2885dee
CR-008	PO	Atualiza o package.json para simplificar o script ‘start’ e adicionar o comando de build do EAS.	21/10/2025	Aprovada	Implementada	ad8db63
CR-009	PO	Atualização do README.md	23/10/2025	Aprovada	Implementada	2e68b5e
CR-010	PO	Adiciona a estrutura inicial do projeto e implementa a comunicação o ESP-NOW para detecção de distância e controle de vibração.	26/10/2025	Aprovada	Implementada	cbc669
CR-011	PO	Remove arquivos obsoletos (descontinuados) e atualiza a estrutura do projeto.	26/10/2025	Aprovada	Implementada	00f5113
CR-012	PO	Implementa o servidor de desenvolvimento com WebSocket para PCD Visual	28/10/2025	Aprovada	Implementada	60e720d
CR-013	PO	Adiciona o componente Bateria (Battery) e atualiza os mapeamentos de ícones no IconSymbol	28/10/2025	Aprovada	Implementada	428e4c7
CR-014	PO	Substituir IconSymbol por Imagem (Image) dos óculos da HomeScreen	28/10/2025	Aprovada	Implementada	4aa4aff
CR-015	PO	Atualiza a configuração do dependabot para o ecossistema de pacotes	28/10/2025	Aprovada	Implementada	6015feb
CR-016	DEV	Adicionar contexto do App	28/10/2025	Aprovada	Implementada	2a17d03
CR-017	DEV	Adicionar estrutura base	28/10/2025	Aprovada	Implementada	6e6b8d2
CR-018	PO	Adicionar servidor de visão para PCD com suporte a Websocket e rotas HTTP	29/10/2025	Aprovada	Implementada	f4ff1cc
CR-019	PO	Inicia a API de Visão (Vision API) com endpoints de extratos de extremidade (edge) de verificação de saúde (health check) e detecção	29/10/2025	Aprovada	Implementada	58048df
CR-020	PO	Adiciona o cliente Websocket para ESP32-CAM com manipulação de detecção e comandos	29/10/2025	Aprovada	Implementada	936b497
CR-021	PO	Adiciona arquivos via upload (inventario de itens de configuração)	04/11/2025	Aprovada	Implementada	2bf3c75

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-022	PO	Adiciona arquivos via upload	04/11/2025	Aprovada	Implementada	4de820e
CR-023	PO	Adiciona arquivos via upload	04/11/2025	Aprovada	Implementada	4fe048b
CR-024	PO	Rename status accounting report.pdf to delet	11/11/2025	Aprovada	Implementada	b84513e
CR-025	PO	Rename doc padrao.pdf to delet_2	11/11/2025	Aprovada	Implementada	b4b37fa
CR-026	PO	Rename inventario de itens de configuraçao.pdf to delet_3	11/11/2025	Aprovada	Implementada	c7cb6b2
CR-027	PO	atualizações locais	12/11/2025	Aprovada	Implementada	b215906
CR-028	PO	Adiciona novas pastas ao projeto	12/11/2025	Aprovada	Implementada	c1875d3
CR-029	PO	Adicionando novos documentos	12/11/2025	Aprovada	Implementada	dbeed15
CR-030	PO	Add kaz-image-captioning subproject	13/11/2025	Aprovada	Implementada	4dcbb58e
CR-031	PO	Fix: Adiciona kaz-image-captioning como diretório normal (não submodule)	13/11/2025	Aprovada	Implementada	5a15e29
CR-032	PO	Refactor: Replace argparse alias with direct import and add run script for testing.	14/11/2025	Aprovada	Implementada	a2e0a3b
CR-033	PO	Add MPU6050 testing documentation and isolated test code.	15/11/2025	Aprovada	Implementada	9d7d2aa
CR-034	PO	Merge branch ‘main’ of github.com:fabiobrasileiroo/sistema_de_pagamento_escolar_por_nfc	15/11/2025	Aprovada	Implementada	a355993
CR-035	PO	Merge branch ‘main’ of github.com:fabiobrasileiroo/consciencia-espacial-PCD-visual	16/11/2025	Aprovada	Implementada	7198f12
CR-036	PO	feat: Add SSE and WebSocket test scripts for monitoring and debugging	16/11/2025	Aprovada	Implementada	22c0219
CR-037	PO	feat: Adiciona status do servidor e dispositivos conectados na tela de configurações	16/11/2025	Aprovada	Implementada	606b208
CR-038	PO	feat: Adiciona exibição da distância do objeto detectado e configuração da URL da API	16/11/2025	Aprovada	Implementada	2e801b8
CR-039	PO	style: Ajusta formatação e espaçamento no componente SettingsScreen	16/11/2025	Aprovada	Implementada	16f4088
CR-040	PO	Refactor Settings Screen: Replace emoji icons with Lucide icons, enhance loading state with SkeletonLoader, and update old styles file and consolidate styles into a new shared styles...	16/11/2025	Aprovada	Implementada	a4f1ce2
CR-041	PO	style: Ajusta a formatação do código nos componentes SettingsScreen e SkeletonLoader	16/11/2025	Aprovada	Implementada	2fdafaa
CR-042	PO	Create README.md for object detection project	16/11/2025	Aprovada	Implementada	dd5ffc4

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-043	PO	feat: Adiciona instalação do googletrans e atualiza script de execução com comandos de instalação de pacotes para...	16/11/2025	Aprovada	Implementada	d6c30cd
CR-044	PO	feat: Adiciona novos scripts para captura e processamento de imagens com ESP32-CAM, incluindo suporte a streaming e...	16/11/2025	Aprovada	Implementada	3ef875b
CR-045	PO	Merge remote-tracking branch 'refs/remotes/origin/main'	17/11/2025	Aprovada	Implementada	6324c83
CR-046	PO	Add ESP32-CAM firmware with captive portal and camera configuration,	17/11/2025	Aprovada	Implementada	d057722
CR-047	PO	feat: Atualiza envio de detecções para servidor via HTTP POST e ajusta script de execução.	17/11/2025	Aprovada	Implementada	071250d
CR-048	PO	Merge remote-tracking branch 'refs/remotes/origin/main'	17/11/2025	Aprovada	Implementada	3ab29c0
CR-049	PO	feat: Atualiza o módulo de motor de vibração com melhorias na configuração WiFi e mensagens de status, além de...	17/11/2025	Aprovada	Implementada	d029742
CR-050	PO	Merge branch 'main' of github.com:fabriobrasileiroo/sistema_de_pagamento_escolar_por_nfc	17/11/2025	Aprovada	Implementada	f0637dd
CR-051	PO	feat: Implement webcam capture functionality with realtime/manual movement and...	17/11/2025	Aprovada	Implementada	1d182f9
CR-052	PO	Add final model image to README	18/11/2025	Aprovada	Implementada	

5. Resumo do Status Atual

Área	Percentual Concluído	Observações
Firmware ESP32	95%	Comunicação BLE estável, calibração finalizada. Aguardando testes de QA.
Módulo IA (TFLite)	90%	Modelo leve implementado e integrado. Aguardando testes de QA.
Aplicativo Android	90%	Tela de calibração e logs concluídas. Exportação CSV implementada. Aguardando testes de QA.
Hardware Integrado	90%	Protótipo funcional testado em bancada.
Testes e QA	75%	Casos CT01–CT03 executados e aprovados. Início dos testes de integração (CT04–CT06).

6. Histórico de Versões e Auditorias

Data	Auditor	Escopo	Resultado	Ações Corretivas
05/10/2025	José Filho	Repositório Git (controle de branches)	Sem não-conformidades	—
18/10/2025	CCB	Requisitos vs Casos de Uso	100% rastreável	—
25/10/2025	QA	Plano de Testes	Corrigidos 2 desvios menores em CT03	—
02/11/2025	CCB	Status Accounting Review	Status OK	Nenhuma ação requerida
09/11/2025	QA	Product Baseline (PB) 1.0	Artefatos prontos para testes de aceitação.	Nenhuma ação requerida

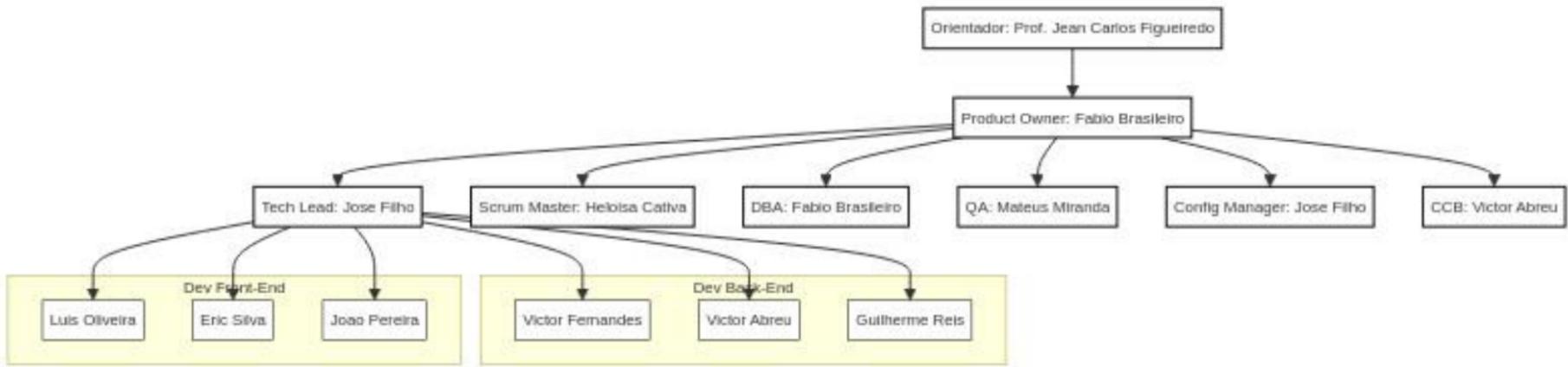
7. Conclusão e Próximos Passos

- O projeto LUMI encontra-se na fase de Qualidade e Testes (QA), com o **Product Baseline (PB) 1.0** estabelecido em 09/11/2025.
- Todos os Itens de Configuração de software (CI-SW-002, CI-SW-003, CI-SW-004) atingiram a Versão 1.0 e estão sob controle de configuração.
- A próxima atualização do Status Accounting (v1.2) incluirá:
 - Resultados de teste de campo e testes de aceitação;
 - Atualização final do Product Baseline (PB) após aprovação do CCB;
 - CRs implementados e revalidados.

Previsão de auditoria final interna: 18/11/2025. **Entrega PoC:** 22/11/2025.

8. Assinaturas

Nome	Cargo	Assinatura	Data
José Filho	Configuration Manager	_____	09/11/2025
Heloisa Cativo	Scrum Master	_____	09/11/2025
Jean Carlos Figueiredo	Orientador / CCB Chair	_____	09/11/2025



File: merge_docs_to_pdf.py

Type: .py

Size: 4117 bytes

Path: /home/fabionote/inovatech-2025/consciencia-espacial-PCD-visual/docs/merge_docs_to_pdf.py

pypdf>=3.0.0
Pillow>=10.0.0
pandas>=2.2.2
openpyxl>=3.1.2

CHECKLIST ERS – DOCUMENTO COMPLETO

Cada item contém um emoji indicando que a seção está presente e validada.

1. Identificação do Documento

2. Resumo e Abstract

3. Introdução

4. Escopo do Sistema

5. Minimundo

6. Requisitos Funcionais

- Medição de distância
- Alerta tátil
- Captura de imagem
- Reconhecimento de objetos
- Comunicação BLE
- TTS no app
- Logs, parâmetros, modo silêncio, exportação, auto-teste

7. Regras de Negócio

- Priorização por distância
- Vibração por faixa
- Economia de energia- Fallback e armazenamento

8. Requisitos Não Funcionais

- Latência
- Acurácia
- Autonomia
- Usabilidade
- Segurança
- Portabilidade
- Peso

9. Hardware e Arquitetura

- Sensores ultrassônicos HC-SR04
- Atuador vibratório (vibracall)
- ESP32 / ESP32-CAM / ESP32-S3
- Bateria 3.7–5V com proteção
- Estrutura wearable (óculos/boné)
- Comunicação BLE e Wi-Fi opcional

- Módulos independentes (distância, visão, app)
- Requisitos de consumo, peso e autonomia

10. Casos de Uso 

- Detectar obstáculo
- Identificar objeto
- Emitir TTS
- Calibrar sistema- Exportar logs

11. Modelo Estrutural (Classes) 

12. Modelo de Interação (Sequência) 

13. Modelagem de Subsistemas 

- Sensoriamento tátil
- Visão computacional
- Aplicativo mobile

14. Glossário 

15. Referências 

Documento validado e organizado.

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

RF03	Captura de Imagem: O sistema deve capturar imagens a cada X segundos (configurável) e disponibilizar para inferência local ou envio ao app quando solicitado.	Alta	RF02
RF04	Reconhecimento de Objetos (5 classes prioritárias): 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	Comunicação com App (BLE): Enviar eventos (tipo, distância, timestamp, confidence) via BLE para app Android e receber comandos de calibração.	Alta	RF04
RF06	TTS no App: O app deve reproduzir mensagem curta correspondente ao evento (ex.: “fio à frente, 1,2 metros”) em ≤ 400 ms após recebimento do evento	Média	RF05
RF07	Registro/Logs: Registrar localmente eventos com campos: timestamp, tipo, distância, confidence, bateria.	Alta	RF03
RF08	Gestos/Botões de Controle: Permitir ligar/desligar alertas via botão físico no wearable.	Médio	
RF09	Fallback de processamento: Caso a inferência local não alcance metas de latência ou acurácia, permitir offload da imagem para o smartphone para inferência.	Médio	
RF10	Atualização de Parâmetros via App: Permitir ajustes em thresholds de distância, sensibilidade de vibração, taxa de captura de imagem e volume de TTS pelo app.	Médio	
RF11	Verificar dispositivos pelo App: 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	Alerta de Bateria Baixa: 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	Confirmação de Inicialização (Auto-Teste): 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	Modo de Silêncio Temporário: O usuário deve poder pausar (silenciar) temporariamente todos os alertas (tátteis e sonoros) através de um comando no wearable (ex: um clique duplo no botão RF08) ou no App (via RF10).	Média	RF08, RF10
RF15	Exportação de Logs de Teste: 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 ≤ 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 ≤ 400 ms; vibração acionada em ≤ 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 \leq 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 ± 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)		Imagen 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 ≤ 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.

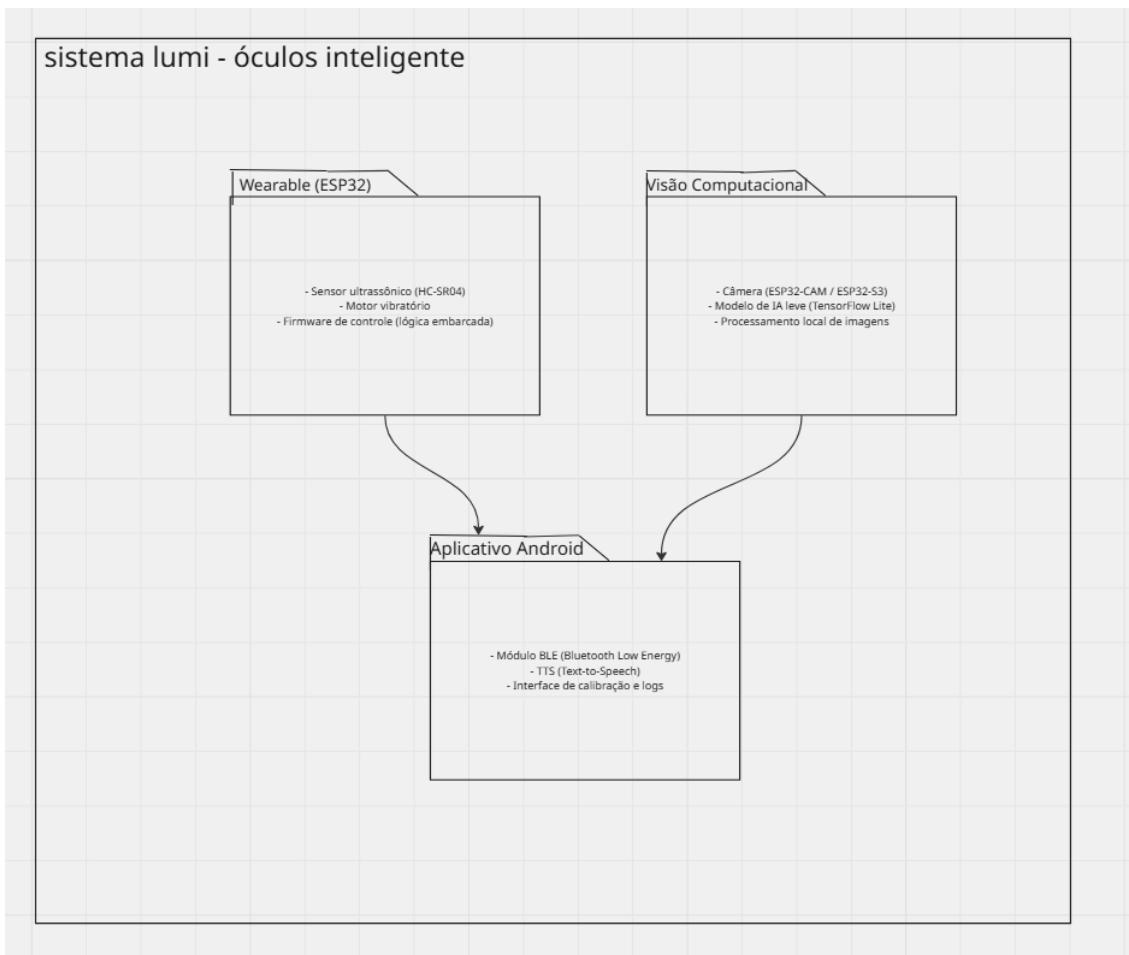
3. Modelagem de Subsistemas

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.

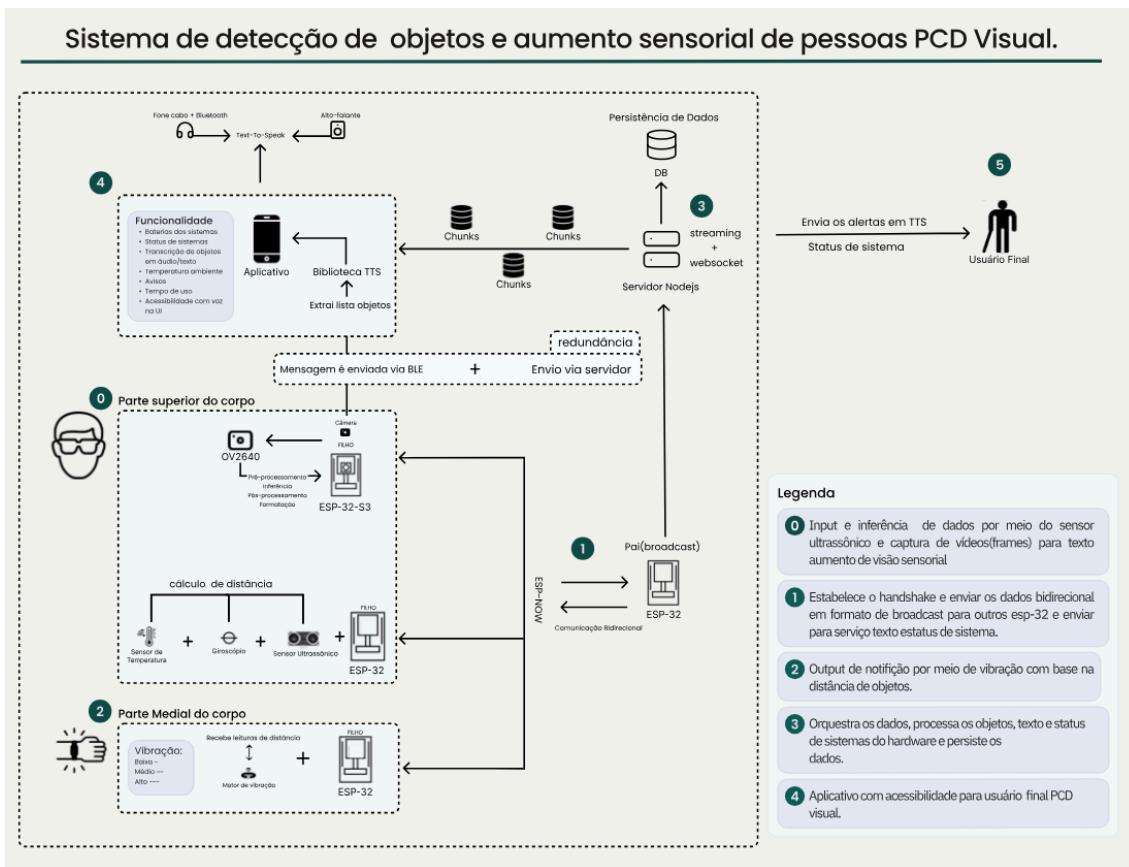
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio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



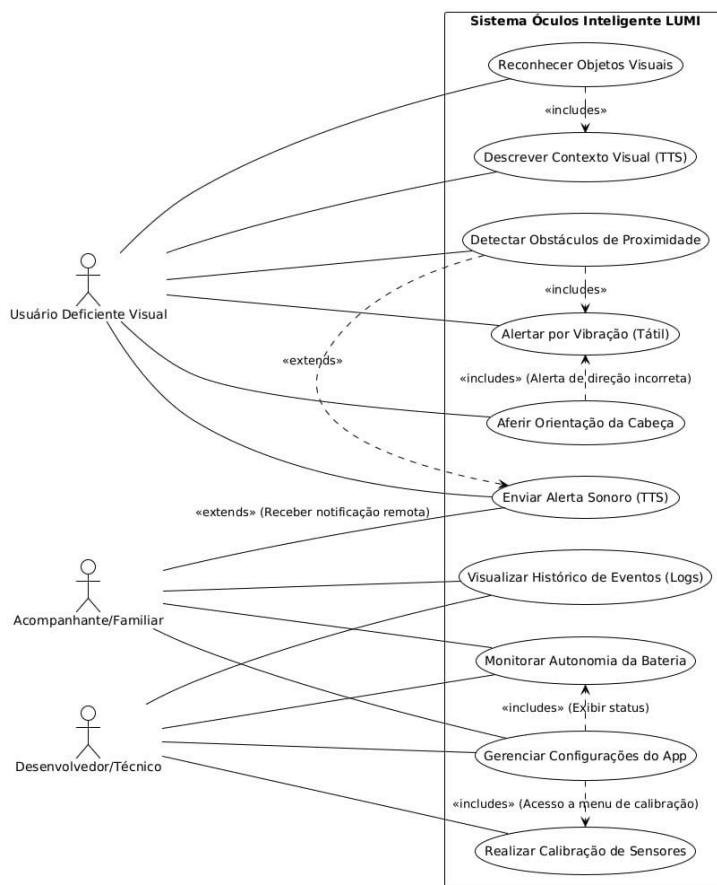
(Imagen – Diagrama de Visual)

4. Modelagem de Casos de Uso

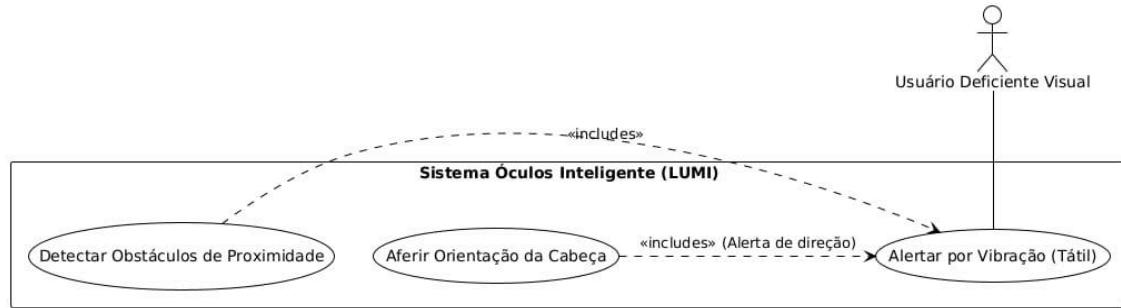
4.1 Atores

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)



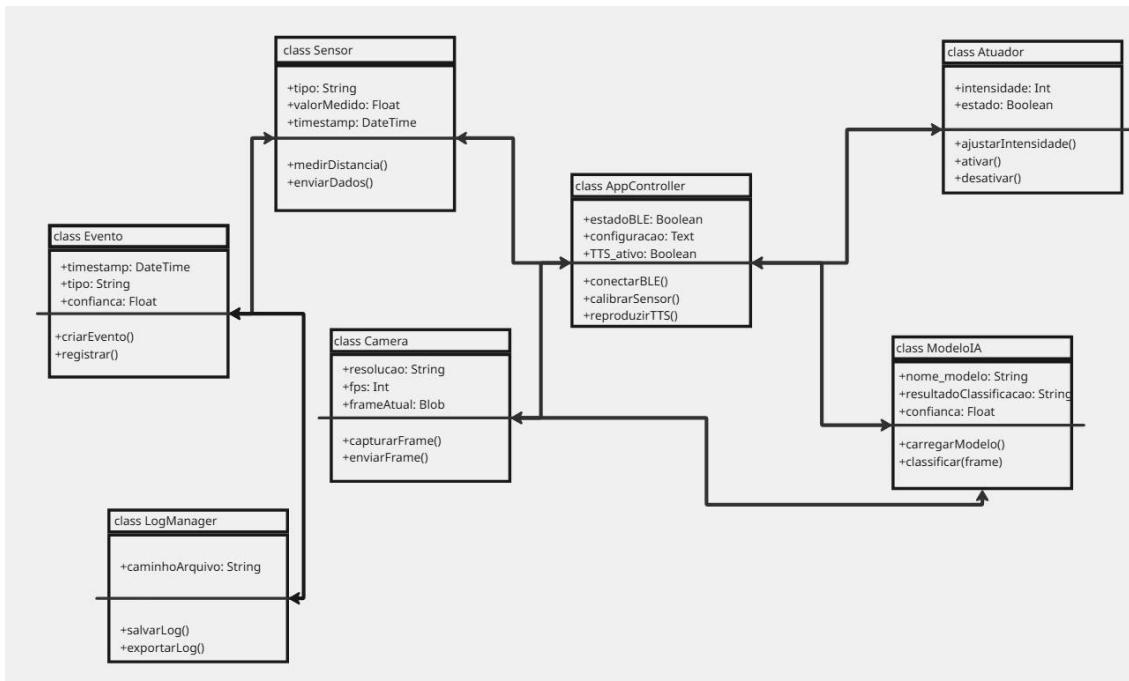
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

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)

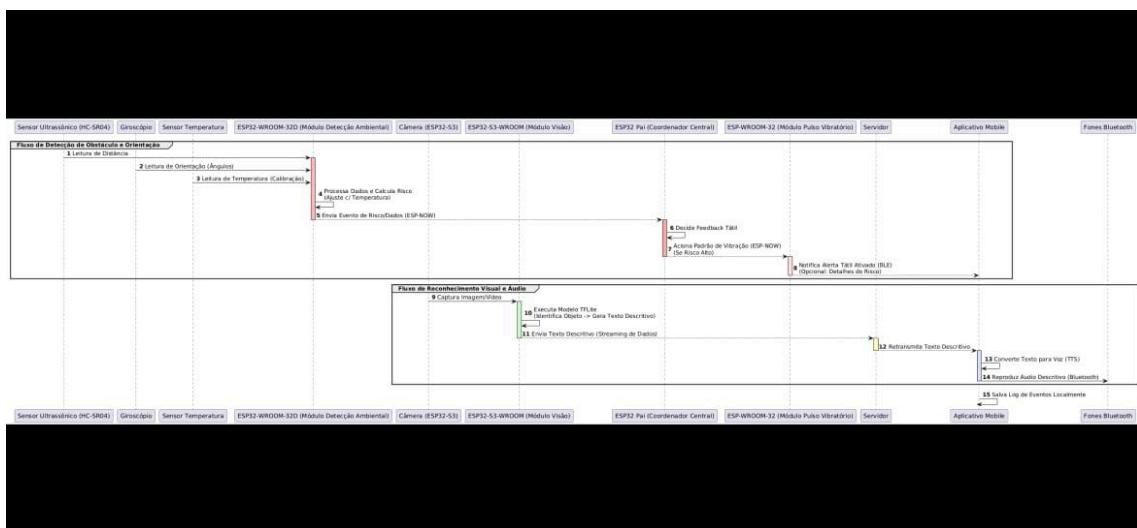
Descrição textual das principais classes:

- **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).

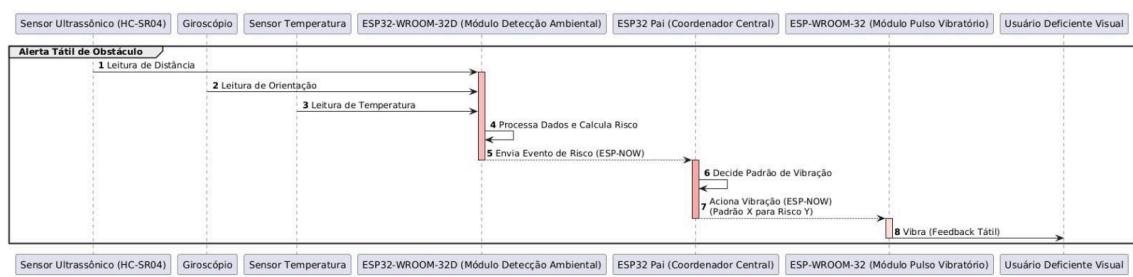
- **LogManager:** salva e exporta logs.

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:** ≥ 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.

MANUAL DA EQUIPE – PROJETO ÓCULOS INTELIGENTE

1. Objetivo do Manual

Este manual define **funções, responsabilidades, regras internas, fluxos de trabalho e procedimentos padrão** para a equipe do projeto “Óculos Inteligente de Baixo Custo”.

Serve como referência para manter **organização, alinhamento e consistência** durante todo o ciclo de desenvolvimento.

2. Estrutura da Equipe e Papéis

Product Owner – Fabio Brasileiro

- Define prioridades do backlog.
- Aprova critérios de aceitação.
- Alinha requisitos com stakeholders.

Scrum Master – Heloísa Cativo

- Garante que o time siga Scrum.
- Remove impedimentos.
- Facilita reuniões e boa comunicação.

Tech Lead / Arquiteto – José Filho

- Define arquitetura do sistema.
- Aprova decisões técnicas.
- Orienta devs sobre boas práticas.

Desenvolvedores Backend – Victor Fernandes, Victor Abreu, Guilherme Reis

- Implementam lógica embarcada e APIs.
- Integram sensores, IA e BLE.
- Criam testes de unidade.

Desenvolvedores Frontend/Android – Luis Oliveira, Eric Silva, João Pereira

- Desenvolvem app Android.
- Criam telas, acessibilidade e TTS.
- Integram BLE e calibradores do sistema.

QA / Testador – Mateus Miranda

- Cria planos e casos de testes.
- Valida módulos e sprint releases.
- Registra bugs e acompanha correções.

Config Manager – José Filho

- Cuida de versionamento (Git).
- Organiza branches e releases.
- Garante integridade do repositório.

CCB – Victor Abreu

- Aprova solicitações de mudança.
- Avalia impacto técnico e de cronograma.

3. Regras Internas da Equipe

3.1 Comunicação

- Canal principal: **WhatsApp + Discord**
- Reuniões:
 - **Daily:** 15 min
 - **Revisão de Sprint:** ao final de cada sprint
 - **Retrospectiva:** após review
- Chamadas urgentes: marcar @todos no Discord.

3.2 Regras de Commit (Git)

Sempre seguir o padrão:

`type: descrição curta`

- `o que foi feito`
- `motivo da mudança`

Types permitidos:

`feat, fix, docs, test, refactor, perf, style`

Branch pattern:

- `main` → versão estável
- `develop` → versão de integração
- `feature/nome` → novas funções
- `hotfix/nome` → correções urgentes

3.3 Boas Práticas de Código

- Nome de variáveis claro e objetivo.
- Um commit por tarefa.
- Evitar código duplicado.
- Sempre documentar funções principais.
- Criar testes unitários onde possível.

4. Procedimentos Operacionais

4.1 Desenvolvimento de Hardware

- Validar sensores antes de integrar.
- Testar HC-SR04 em bancada com régua.
- Testar vibracall com PWM em três níveis.
- Testar câmera com snapshot antes da IA.
- Registrar consumo da bateria a cada iteração.

4.2 Desenvolvimento do Firmware (ESP32)

Fluxo padrão:

1. Implementar leitura de sensores.
2. Garantir tratamento de erros.
3. Enviar eventos via BLE.
4. Testar latência e estabilidade.

5. Integrar com app Android.

Checklist interno:

- Distância
- Vibração proporcional
- Captura de imagem
- Inferência TFLite
- BLE

4.3 Desenvolvimento do App Android

- Criar telas simples e acessíveis.
- Implementar TTS responsivo.
- Garantir sincronização BLE.
- Incluir calibração (thresholds e vibração).
- Feedback por voz e botões grandes.

4.4 Testes

QA deve validar:

- Funcionalidade dos sensores
- Reconhecimento de objetos
- Latência ponta-a-ponta
- Uso com 1–5 usuários reais
- Exportação de logs
- Estabilidade da conexão BLE

5. Organização do Repositório

```
/docs
  ERS/
    diagramas/

/hardware
  esp32/
    esquemáticos/

/app
  android app/

/firmware
  modules/
    ble/
    vision/

/tests
  unit/
  integration/
```

6. Solicitações de Mudança (CCB)

Qualquer alteração no sistema deve seguir:

1. Abrir issue: “Solicitação de Mudança”.
2. CCB avalia impacto.
3. Decisão registrada no GitHub.
4. Implementação inicia somente após aprovação.

7. Fluxo de Sprint (Scrum)

Duração: 7 dias

Dia 1 – Planejamento

Definição do backlog da sprint.

Dias 2–6 – Execução

Desenvolvimento, testes e integração.

Dia 7 – Review + Retrospectiva

8. Regras de Qualidade

- Latência \leq 400 ms
- Acurácia \geq 70%
- Autonomia \geq 2 horas
- BLE estável
- Vibração perceptível
- Logs funcionando

9. Checklist Final da Sprint (interno)

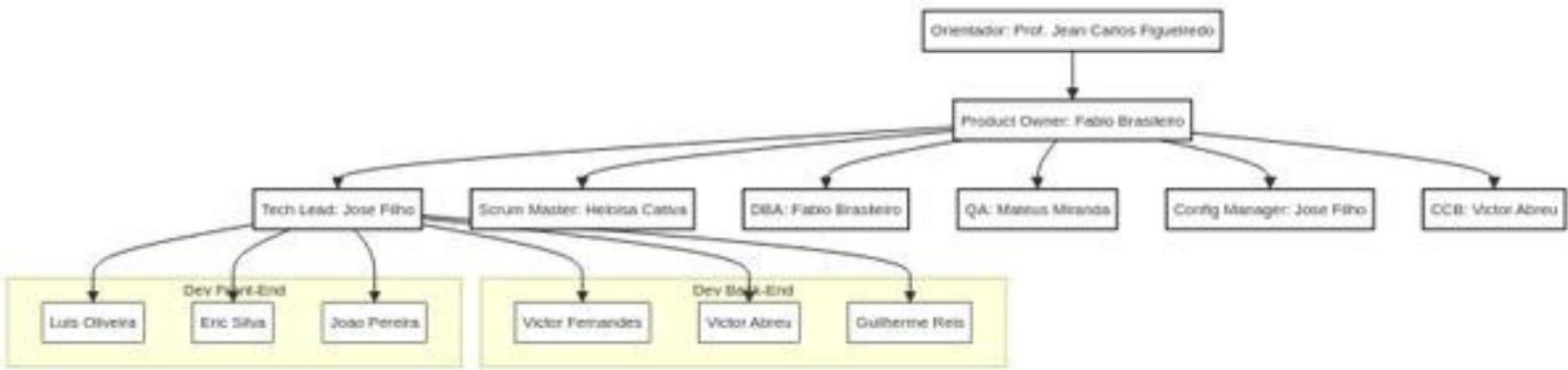
- Código testado
- Testes feitos pelo QA
- Documentação atualizada

- Bugs corrigidos
- Branch mergeada corretamente
- Versão registrada



10. Contatos e Papéis Rápidos

- **PO:** decisões de produto
- **SM:** processos ágeis
- **Tech Lead:** decisões técnicas
- **QA:** testes
- **Config Manager:** Git
- **CCB:** aprova mudanças



1. Plano de projeto

Projeto:

Óculos inteligente de baixo custo que detecta perigos ambientais e fornece alertas táticos 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 desenvolver 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, Guilherme Reis
- 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: Total de funcionalidades concluídas / 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.

Guia Passo a Passo para o Desenvolvimento do Projeto "Óculos Inteligente de Baixo Custo"

Este documento apresenta um guia passo a passo detalhado para a execução do projeto "Óculos Inteligente de Baixo Custo para Auxílio à Locomoção de Pessoas com Deficiência Visual", conforme o **Modelo de Ciclo de Vida Incremental com Sprints Semanais** definido no Documento de Análise de Requisitos. O projeto tem uma duração total de 6 semanas e visa a entrega de um Protótipo de Prova de Conceito (PoC) funcional.

1. Estrutura de Governança e Papéis

A execução do projeto é baseada em uma estrutura de equipe ágil, com papéis e responsabilidades claramente definidos, conforme o organograma e a matriz de responsabilidades do documento de origem.

Papel	Responsável	Foco Principal
Product Owner (PO)	Fabio Brasileiro	Maximizar o valor do produto, gerenciar e priorizar o <i>Backlog</i> de funcionalidades.
Scrum Master (SM)	Heloisa Cativo	Garantir a adesão ao processo <i>Scrum</i> , remover impedimentos e facilitar as reuniões.
Tech Lead / Arquiteto	Jose Filho	Definir a arquitetura do sistema (Módulos 1, 2 e 3), garantir a integridade técnica e orientar os desenvolvedores.
Desenvolvedores (Devs)	Luis Oliveira, Eric Silva, João Pereira, Victor Fernandes, Victor Abreu, Guilherme Reis	Implementação do <i>firmware</i> embarcado, desenvolvimento do aplicativo Android e integração dos módulos.
QA/Testador	Mateus Miranda	Planejar e executar testes funcionais, de usabilidade e de desempenho (latência e acurácia).

2. Plano de Execução Incremental (6 Sprints)

O desenvolvimento é dividido em seis *sprints* semanais, cada uma focada na entrega de um incremento funcional do sistema. O objetivo é validar hipóteses de negócio e incorporar *feedback* de forma contínua.

Sprint 1: Módulo Tátil e Base de Hardware

Foco: Estabelecer a base de *hardware* e a funcionalidade primária de detecção de distância e *feedback* tátil.

Atividade	Requisitos Chave	Produto de Entrega
1.1. Implementação do Módulo 1	RF01 (Medição de Distância), RF02 (Alerta Tátil Proporcional)	Protótipo funcional do Módulo 1 (ESP32/Pico + HC-SR04 + Vibracall) capaz de medir distância e acionar vibração proporcional em ≤ 100 ms.
1.2. Auto-Teste de Inicialização	RF13 (Confirmação de Inicialização)	Rotina de <i>firmware</i> para auto-teste de sensores (HC-SR04) e emissão de sinal de "Sistema Pronto" (vibração longa/curta).
1.3. Monitoramento de Energia	RF12 (Alerta de Bateria Baixa)	Implementação do circuito de monitoramento de bateria e lógica de alerta crítico.

Sprint 2: Módulo de Visão e Coleta de Dados

Foco: Integrar o subsistema de visão computacional e a capacidade de registro de eventos (*logging*).

Atividade	Requisitos Chave	Produto de Entrega
2.1. Implementação do Módulo 2	RF03 (Captura de Imagem)	Módulo 2 (ESP32-CAM/S3) configurado para captura periódica de imagens.
2.2. Estrutura de Logs	RF07 (Registro/Logs)	Implementação da estrutura de dados para registro local de eventos (<i>timestamp</i> , tipo, distância, <i>confidence</i>).
2.3. Treinamento do Modelo	RF04 (Reconhecimento de Objetos)	Início do treinamento e otimização do modelo TFLite para as 5 classes prioritárias (fio/ramo, degrau, cadeira/obstáculo baixo, pessoa, porta).

Sprint 3: Comunicação e Aplicativo Mínimo

Foco: Estabelecer a comunicação sem fio entre o *wearable* e o *smartphone* e desenvolver a interface básica do usuário.

Atividade	Requisitos Chave	Produto de Entrega
3.1. Conexão BLE	RF05 (Comunicação com App - BLE)	Conexão BLE estável para envio de eventos e recebimento de comandos de calibração.

Atividade	Requisitos Chave	Produto de Entrega
3.2. Interface de Status	RF11 (Verificar dispositivos pelo App)	Aplicativo Android mínimo exibindo status de conexão, nível de bateria e status operacional dos sensores.
3.3. Testes de Latência	RNF01 (Latência)	Testes de ponta a ponta para garantir que a latência de comunicação esteja dentro do limite de 400 ms.

Sprint 4: Inteligência e Feedback Sonoro (TTS)

Foco: Integrar o modelo de reconhecimento e o sistema de alerta de voz, completando o ciclo de *feedback* multimodal.

Atividade	Requisitos Chave	Produto de Entrega
4.1. Integração do Reconhecimento	RF04 (Reconhecimento de Objetos)	Modelo TFLite integrado ao <i>firmware</i> (ou App, via <i>offload</i>) e funcional, com acurácia alvo de 70%.
4.2. Alerta de Voz (TTS)	RF06 (TTS no App)	Implementação da funcionalidade de <i>Text-to-Speech</i> (TTS) no App para reproduzir alertas verbais curtos.
4.3. Regras de Negócio	RN01, RN02, RN03, RN04	Implementação da lógica de priorização de alertas (distância vs. perigo suspenso) e intensidade de vibração.

Sprint 5: Usabilidade e Robustez

Foco: Refinar a experiência do usuário, adicionando controles de usabilidade e mecanismos de robustez.

Atividade	Requisitos Chave	Produto de Entrega
5.1. Controles de Usuário	RF08 (Gestos/Botões de Controle), RF14 (Modo de Silêncio Temporário)	Implementação de botão físico no <i>wearable</i> e funcionalidade no App para pausar temporariamente os alertas.
5.2. Interface de Calibração	RF10 (Atualização de Parâmetros via App), RNF04 (Interface de calibração)	Interface no App para ajustes de <i>thresholds</i> de distância, sensibilidade de vibração e volume de TTS.
5.3. Fallback de Processamento	RF09 (Fallback de processamento)	Lógica de <i>offload</i> da imagem para o <i>smartphone</i> para inferência, caso a latência local seja comprometida.

Sprint 6: Validação, Implantação e Entrega

Foco: Testes finais de campo, validação dos critérios de aceitação e preparação da entrega do PoC.

Atividade	Requisitos Chave	Produto de Entrega
6.1. Testes de Campo com Usuários	RNF03 (Autonomia), RNF02 (Acurácia)	Relatório de testes de campo com 1-5 usuários, validando autonomia (≥ 2 horas) e acurácia em ambiente real.
6.2. Exportação de Logs	RF15 (Exportação de Logs de Teste)	Funcionalidade no App para exportar logs de eventos em formato CSV ou JSON para análise <i>offline</i> .
6.3. Implantação e Documentação	-	Versão Candidata (RC) finalizada e documentação técnica mínima do PoC.

3. Critérios de Aceitação do PoC

O sucesso do projeto será medido pela aderência aos seguintes critérios de aceitação, a serem validados na Sprint 6:

Critério	Meta	Requisito Relacionado
Acurácia	$\geq 70\%$ nas 5 classes prioritárias em ambiente controlado.	RF04, RNF02
Latência	Média de alerta (detecção \rightarrow feedback) ≤ 400 ms.	RNF01
Autonomia	Duração mínima de prova ≥ 2 horas com ciclo de uso típico.	RNF03
Validação	Testes preliminares realizados com 1-5 usuários.	-
Funcionalidade	Integração estável e funcional dos Módulos 1, 2 e 3.	RF05, RF11

ID	Tipo	Descrição do Requisito (Resumo)
RF01	Funcional	Medição de Distância (uso do HC-SR04 em 200 ms).
RF02	Funcional	Alerta Tátil Proporcional (vibração com intensidade proporcional à distância).
RF03	Funcional	Captura de Imagem (a cada 100 ms se a distância for crítica).
RF04	Funcional	Reconhecimento de Objetos (disponibilizar inferência local ou via App).
RF05	Funcional	Comunicação com App (transferência de dados, timestamp, confidence via BLE).
RF06	Funcional	TTS no App (mensagens de áudio correspondentes ao evento).
RF07	Funcional	Armazenamento Local (eventos/logs por 30 dias para análise offline).
RF08	Funcional	Controles por Hardware (botão liga/desliga, silêncio temporário, etc.).
RF09	Funcional	Falha de Processamento (notificar se a inferência local falhar).
RF10	Funcional	Atualização de Parâmetros via App (distância, sensibilidade, volume TTS).
RF11	Funcional	Verificar dispositivos pelo App (status de conexão/bateria).
RF12	Funcional	Alerta de Bateria (ativação de alerta sonoro/TTS quando atingir nível crítico).
RF13	Funcional	Confirmação de Inicialização (vibração + TTS "Sistema Ativo").
RF14	Funcional	Modo de Silêncio Temporário (silenciar temporariamente todos os alertas).
RF15	Funcional	Exportação de Logs de Teste (CSV/JSON para análise de desempenho offline).
RNF01	Não Funcional	Latência ponta-a-ponta (vibração < 400 ms).
RNF02	Não Funcional	Acurácia mínima de \$70\%\$ em ambiente controlado.
RNF03	Não Funcional	Autonomia mínima de \$\sim 4\$ horas de uso típico.

RNF04	Não Funcional	Interface de calibração acessível no App.
RNF Outros	Não Funcional	Disponibilidade: \$99.9\%\$ durante a sessão de uso.
RNF Outros	Não Funcional	Dados de usuários e logs armazenados localmente.

Caso de Uso Relacionado	Módulo / Componente
UC04 - Medir Distância	Módulo de Sensoriamento
UC02 - Emitir Alerta Tátil	Módulo de Feedback Tátil
UC07 - Capturar Imagem	Módulo de Captura
UC08 - Reconhecer Objeto	Módulo de Processamento
UC09 - Transmitir Dados	Módulo de Comunicação (BLE)
UC03 - Emitir Alerta Sonoro	Módulo de Áudio (App)
UC10 - Armazenar Log	Módulo de Armazenamento
UC11 - Controlar Dispositivo	Controlador Principal
UC12 - Notificar Falha	Módulo de Processamento
UC05 - Calibrar Sensores	Interface de Configuração (App)
UC13 - Monitorar Status	Módulo de Comunicação (App)
UC03 - Emitir Alerta Sonoro	Hardware / Energia
UC01 - Detectar Obstáculo	Módulo de Feedback Tátil
UC11 - Controlar Dispositivo	Controlador Principal
UC10 - Armazenar Log	Módulo de Comunicação (App)
—	Módulo de Sensoriamento
—	Módulo de Sensoriamento
—	Hardware / Energia

—	Interface de Configuração (App)
—	Confiabilidade
—	Segurança

Caso de Teste Relacionado	Prioridade	Status
CT01	Alta	Em Desenvolvimento
CT01, CT02, CT05	Alta	Em Desenvolvimento
CT03	Alta	Pendente
—	Alta	Pendente
CT03	Alta	Aprovado
—	Média	Pendente
—	Alta	Aprovado
—	Média	Em Desenvolvimento
—	Média	Pendente
—	Média	Pendente
—	Média	Aprovado
—	Alta	Em Análise
—	Alta	Em Desenvolvimento
—	Média	Aprovado
—	Média	Pendente
CT01 (Implícito)	Alta	Em Desenvolvimento
CT04 (Implícito)	Alta	Em Análise
CT07 (Implícito)	Alta	Em Análise

CT05 (Implícito)	Média	Pendente
—	Alta	Em Análise
—	Alta	Aprovado

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 tátteis 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 ≤ 400 ms, autonomia de prova ≥ 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 ≤ 400 ms, a test battery life of ≥ 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 tátteis (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 táticos 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 desenvolver 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, Guilherme Reis
- 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: Total de funcionalidades concluídas / 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	Medição de Distância: O sistema deve medir distância em frente ao usuário usando HC-SR04 e disponibilizar leitura atual a cada 200–500 ms.	Alta	-
RF02	Alerta Tátil Proporcional: Ao detectar distância abaixo de thresholds configuráveis, o dispositivo deve acionar o vibracall com intensidade proporcional (fraco/médio/alto) em ≤ 100 ms após leitura.	Alta	RF01

RF03	Captura de Imagem: O sistema deve capturar imagens a cada X segundos (configurável) e disponibilizar para inferência local ou envio ao app quando solicitado.	Alta	RF02
RF04	Reconhecimento de Objetos (5 classes prioritárias): 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	Comunicação com App (BLE): Enviar eventos (tipo, distância, timestamp, confidence) via BLE para app Android e receber comandos de calibração.	Alta	RF04
RF06	TTS no App: O app deve reproduzir mensagem curta correspondente ao evento (ex.: “fio à frente, 1,2 metros”) em ≤ 400 ms após recebimento do evento	Média	RF05
RF07	Registro/Logs: Registrar localmente eventos com campos: timestamp, tipo, distância, confidence, bateria.	Alta	RF03
RF08	Gestos/Botões de Controle: Permitir ligar/desligar alertas via botão físico no wearable.	Médio	
RF09	Fallback de processamento: Caso a inferência local não alcance metas de latência ou acurácia, permitir offload da imagem para o smartphone para inferência.	Médio	
RF10	Atualização de Parâmetros via App: Permitir ajustes em thresholds de distância, sensibilidade de vibração, taxa de captura de imagem e volume de TTS pelo app.	Médio	
RF11	Verificar dispositivos pelo App: 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	Alerta de Bateria Baixa: 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	Confirmação de Inicialização (Auto-Teste): 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	Modo de Silêncio Temporário: O usuário deve poder pausar (silenciar) temporariamente todos os alertas (tátteis e sonoros) através de um comando no wearable (ex: um clique duplo no botão RF08) ou no App (via RF10).	Média	RF08, RF10
RF15	Exportação de Logs de Teste: 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 ≤ 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 ≤ 400 ms; vibração acionada em ≤ 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 \leq 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 ± 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)		Imagen 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 ≤ 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.

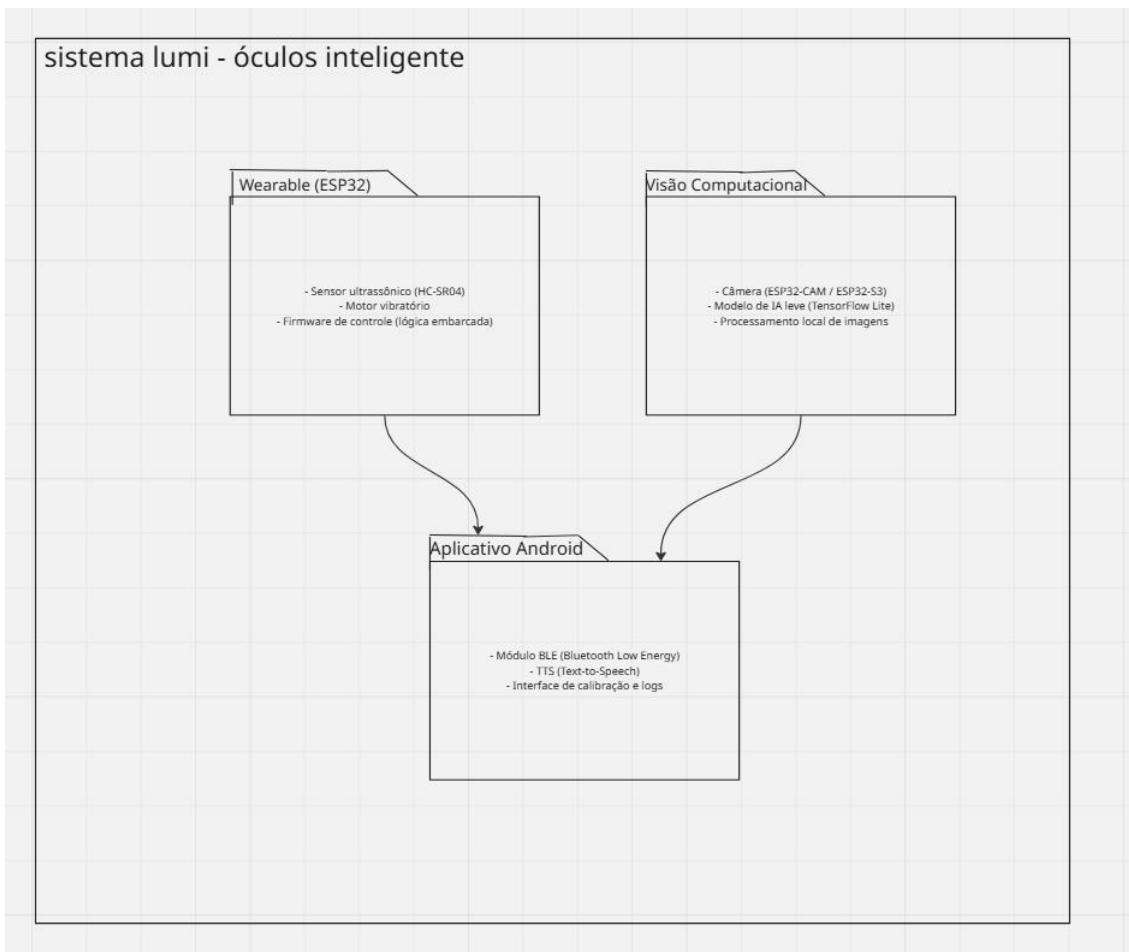
3. Modelagem de Subsistemas

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.

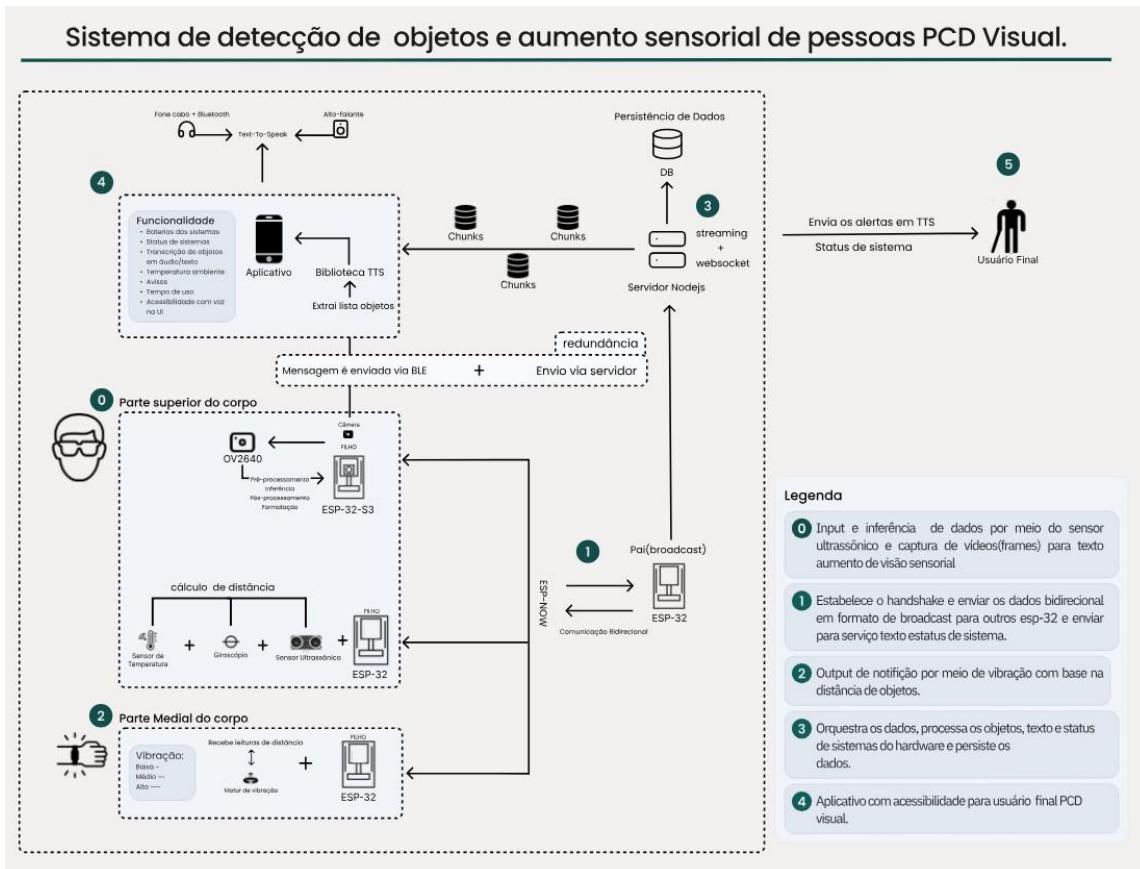
A arquitetura é organizada de forma hierárquica e redundante, permitindo que múltiplos **módulos ESP-32** atuem de forma cooperativa (modelo *pai-filho*) com comunicação bidirecional via BLE e Wi-Fi. O diagrama também evidencia o uso de streaming de vídeo, inferência local com pós-processamento, e envio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



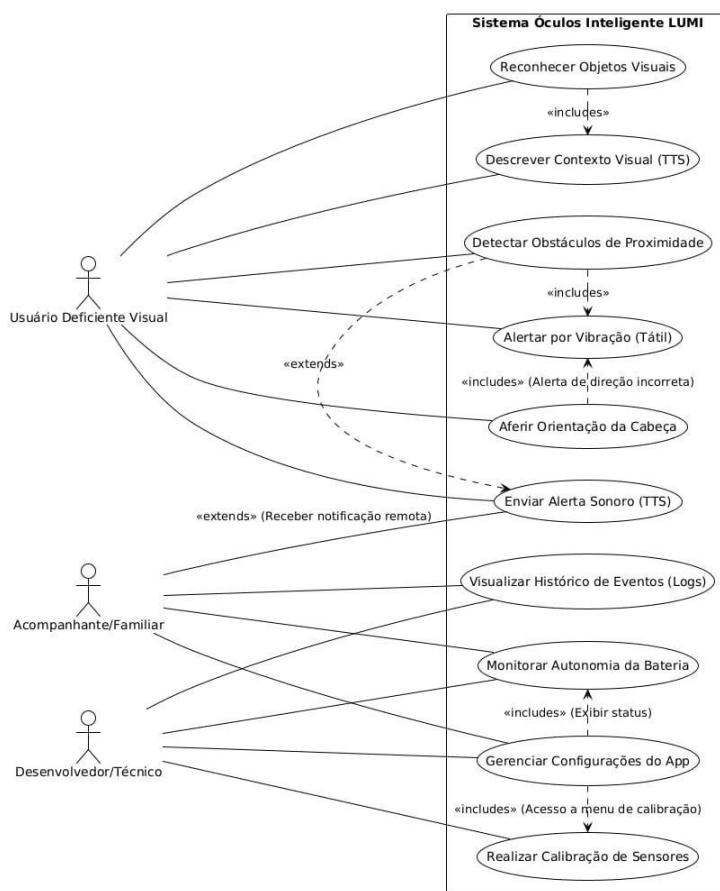
(Imagem – Diagrama de Visual)

4. Modelagem de Casos de Uso

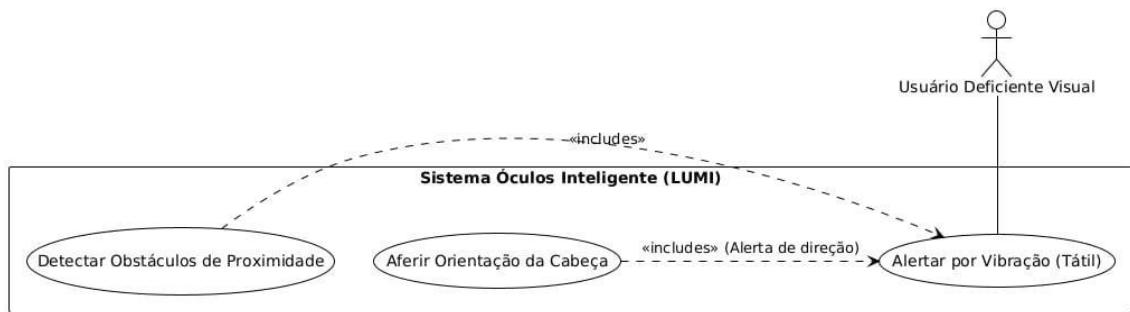
4.1 Atores

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)



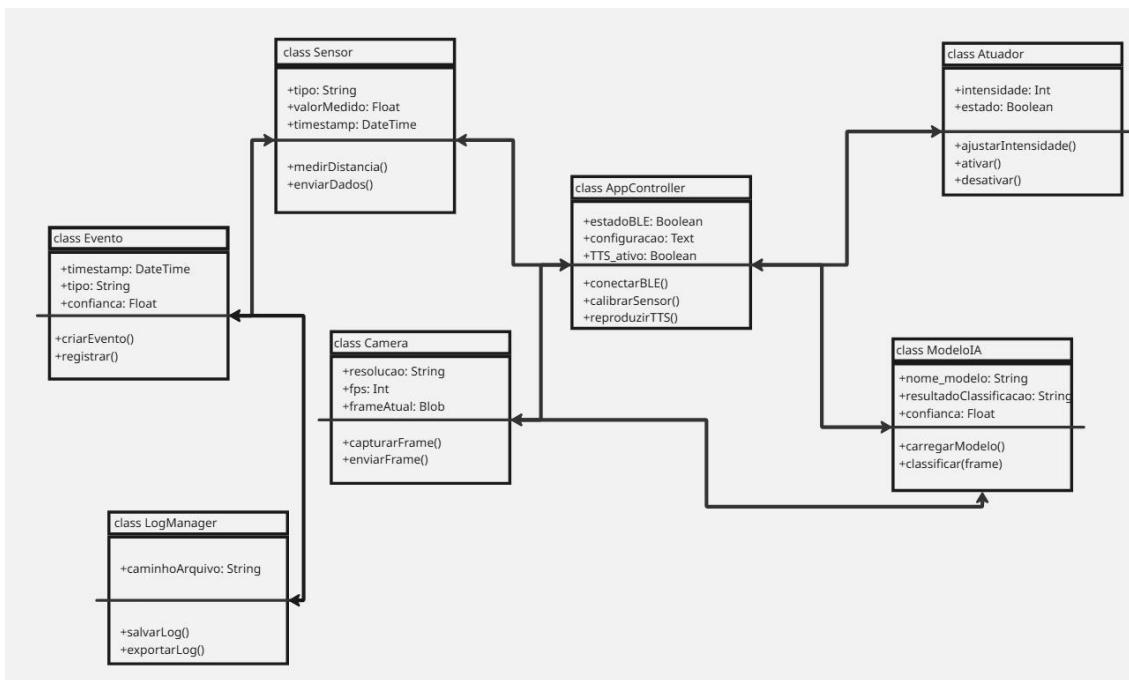
(Imagem – Diagrama de Casos de Uso do Sistema(específico))

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)

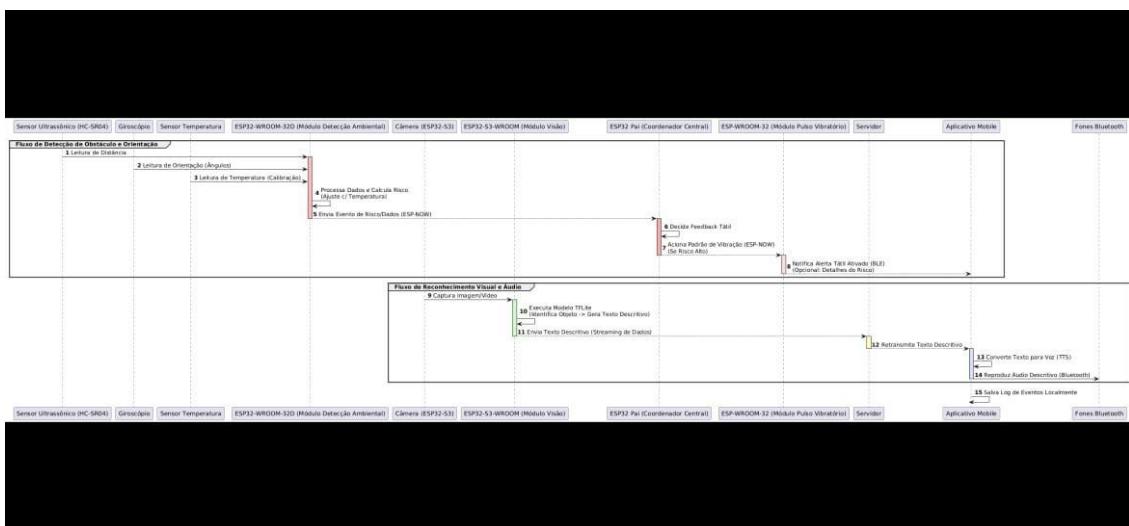
Descrição textual das principais classes:

- **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).

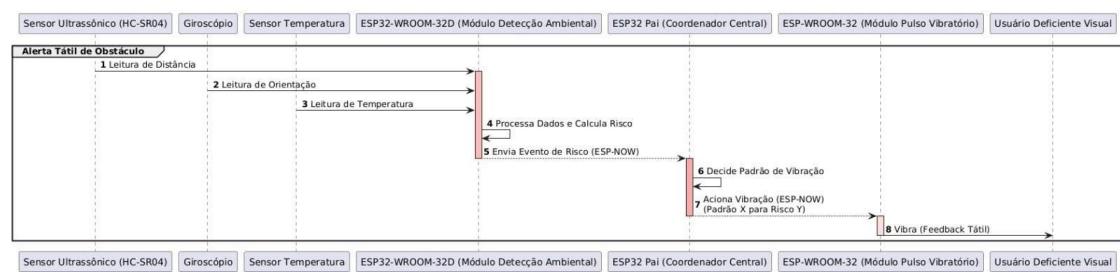
- **LogManager:** salva e exporta logs.

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:** ≥ 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.

RELATÓRIO DE CONTROLE E REGISTRO DE CONFIGURAÇÃO (STATUS ACCOUNTING REPORT)

Projeto: ÓCULOS INTELIGENTE DE BAIXO CUSTO (LUMI) **Versão do Documento:** 1.1 **Data:** 09/11/2025 **Aprovador:** Prof. Jean Carlos Figueiredo (CCB) **Revisão:** Equipe DEV – Engenharia de Software FAMETRO

1. Propósito

Este documento apresenta o Status Accounting (Controle e Registro de Configuração) do projeto LUMI, com o objetivo de garantir a rastreabilidade e visibilidade do progresso de artefatos de software e hardware durante o ciclo de vida do produto.

Inclui informações sobre:

- Itens de configuração (CIs) identificados;
- Status atual de cada item;
- Mudanças aprovadas pela Change Control Board (CCB);
- Versões liberadas e *baseline* vigente.

2. Itens de Configuração (Configuration Items – CIs)

ID	Tipo	Descrição	Versão	Repositório / Local	Status	Responsável
CI-SW-001	Documento	Documento de Requisitos (DR)	1.3	GitHub / docs/requirements	Aprovado	Mateus M. Miranda
CI-SW-002	Código-fonte	Firmware ESP32 (Sensores + Vibração)	1.1	GitHub / src/esp32_main	Em Teste (QA)	Victor Abreu
CI-SW-003	Código-fonte	Módulo IA (TFLite ESP32-CAM)	1.0	GitHub / src/vision_ai	Em Teste (QA)	Victor Fernandes
CI-SW-004	Aplicativo Android	App “LUMI Assist” (TTS + BLE + Logs)	1.0	GitHub / app/android_lumi	Em Teste (QA)	Joao Pereira
CI-HW-001	Hardware	Montagem do protótipo (sensor HC-SR04 + ESP32)	Rev. B	Laboratório FAMETRO	Integrado	Luis Oliveira
CI-TEST-001	Documento	Plano de Teste Funcional	1.1	docs/testing/PT-LUMI.pdf	Aprovado	Mateus M. Miranda
CI-MGMT-001	Documento	Plano de Gerenciamento de Configuração	1.1	docs/config/CM_PLAN_LUMI.pdf	Aprovado	José Filho
CI-DOC-001	Documento	Documento de Análise de Requisitos	1.4	docs/analysis	Aprovado	Heloisa Cativo

3. Baseline de Configuração Vigente

Tipo de Baseline	Data	Itens Incluídos	Observações
Functional Baseline (FB)	01/10/2025	CI-SW-001, CI-DOC-001	Requisitos e arquitetura aprovados pelo CCB.
Allocated Baseline (AB)	15/10/2025	CI-SW-002, CI-HW-001, CI-TEST-001	Integração parcial dos módulos sensoriais e app.
Product Baseline (PB)	09/11/2025	CI-SW-003, CI-SW-004, CI-SW-002, CI-TEST-001	Todos os artefatos de software e teste em fase de QA final para submissão à INOVATECH.

4. Controle de Mudanças (Change Requests – CRs)

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-001	PO	Commit Inicial	06/09/2025	Aprovada	Implementada	37abf1d
CR-002	PO	Adiciona a estrutura inicial (scaffold) do projeto backend e seu esquema de banco de dados.	20/09/2025	Aprovada	Implementada	ed8cab6
CR-003	PO	Adicionar código do sensor ultrassônico HCSR04 em C++.	12/10/2025	Aprovada	Implementada	1f8f21
CR-004	PO	Atualizar título do projeto no README.md.	12/10/2025	Aprovada	Implementada	fd18058
CR-005	PO	Remover código do sensor ultrassônico HCSR04 e adicionar configuração do projeto	12/10/2025	Aprovada	Implementada	2e4b8b9
CR-006	PO	Merge na branch main	12/10/2025	Aprovada	Implementada	f34276c
CR-007	PO	Adiciona script reset-project (script para reiniciar o projeto para inicializar a estrutura do projeto.	21/10/2025	Aprovada	Implementada	2885dee
CR-008	PO	Atualiza o package.json para simplificar o script ‘start’ e adicionar o comando de build do EAS.	21/10/2025	Aprovada	Implementada	ad8db63
CR-009	PO	Atualização do README.md	23/10/2025	Aprovada	Implementada	2e68b5e
CR-010	PO	Adiciona a estrutura inicial do projeto e implementa a comunicação o ESP-NOW para detecção de distância e controle de vibração.	26/10/2025	Aprovada	Implementada	cbc669
CR-011	PO	Remove arquivos obsoletos (descontinuados) e atualiza a estrutura do projeto.	26/10/2025	Aprovada	Implementada	00f5113
CR-012	PO	Implementa o servidor de desenvolvimento com WebSocket para PCD Visual	28/10/2025	Aprovada	Implementada	60e720d
CR-013	PO	Adiciona o componente Bateria (Battery) e atualiza os mapeamentos de ícones no IconSymbol	28/10/2025	Aprovada	Implementada	428e4c7
CR-014	PO	Substituir IconSymbol por Imagem (Image) dos óculos da HomeScreen	28/10/2025	Aprovada	Implementada	4aa4aff
CR-015	PO	Atualiza a configuração do dependabot para o ecossistema de pacotes	28/10/2025	Aprovada	Implementada	6015feb
CR-016	DEV	Adicionar contexto do App	28/10/2025	Aprovada	Implementada	2a17d03
CR-017	DEV	Adicionar estrutura base	28/10/2025	Aprovada	Implementada	6e6b8d2
CR-018	PO	Adicionar servidor de visão para PCD com suporte a Websocket e rotas HTTP	29/10/2025	Aprovada	Implementada	f4ff1cc
CR-019	PO	Inicia a API de Visão (Vision API) com endpoints de extratos de extremidade (edge) de verificação de saúde (health check) e detecção	29/10/2025	Aprovada	Implementada	58048df
CR-020	PO	Adiciona o cliente Websocket para ESP32-CAM com manipulação de detecção e comandos	29/10/2025	Aprovada	Implementada	936b497
CR-021	PO	Adiciona arquivos via upload (inventario de itens de configuração)	04/11/2025	Aprovada	Implementada	2bf3c75

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-022	PO	Adiciona arquivos via upload	04/11/2025	Aprovada	Implementada	4de820e
CR-023	PO	Adiciona arquivos via upload	04/11/2025	Aprovada	Implementada	4fe048b
CR-024	PO	Rename status accounting report.pdf to delet	11/11/2025	Aprovada	Implementada	b84513e
CR-025	PO	Rename doc padrao.pdf to delet_2	11/11/2025	Aprovada	Implementada	b4b37fa
CR-026	PO	Rename inventario de itens de configuraçao.pdf to delet_3	11/11/2025	Aprovada	Implementada	c7cb6b2
CR-027	PO	atualizações locais	12/11/2025	Aprovada	Implementada	b215906
CR-028	PO	Adiciona novas pastas ao projeto	12/11/2025	Aprovada	Implementada	c1875d3
CR-029	PO	Adicionando novos documentos	12/11/2025	Aprovada	Implementada	dbeed15
CR-030	PO	Add kaz-image-captioning subproject	13/11/2025	Aprovada	Implementada	4dcbb58e
CR-031	PO	Fix: Adiciona kaz-image-captioning como diretório normal (não submodule)	13/11/2025	Aprovada	Implementada	5a15e29
CR-032	PO	Refactor: Replace argparse alias with direct import and add run script for testing.	14/11/2025	Aprovada	Implementada	a2e0a3b
CR-033	PO	Add MPU6050 testing documentation and isolated test code.	15/11/2025	Aprovada	Implementada	9d7d2aa
CR-034	PO	Merge branch ‘main’ of github.com:fabiobrasileiroo/sistema_de_pagamento_escolar_por_nfc	15/11/2025	Aprovada	Implementada	a355993
CR-035	PO	Merge branch ‘main’ of github.com:fabiobrasileiroo/consciencia-espacial-PCD-visual	16/11/2025	Aprovada	Implementada	7198f12
CR-036	PO	feat: Add SSE and WebSocket test scripts for monitoring and debugging	16/11/2025	Aprovada	Implementada	22c0219
CR-037	PO	feat: Adiciona status do servidor e dispositivos conectados na tela de configurações	16/11/2025	Aprovada	Implementada	606b208
CR-038	PO	feat: Adiciona exibição da distância do objeto detectado e configuração da URL da API	16/11/2025	Aprovada	Implementada	2e801b8
CR-039	PO	style: Ajusta formatação e espaçamento no componente SettingsScreen	16/11/2025	Aprovada	Implementada	16f4088
CR-040	PO	Refactor Settings Screen: Replace emoji icons with Lucide icons, enhance loading state with SkeletonLoader, and update old styles file and consolidate styles into a new shared styles...	16/11/2025	Aprovada	Implementada	a4f1ce2
CR-041	PO	style: Ajusta a formatação do código nos componentes SettingsScreen e SkeletonLoader	16/11/2025	Aprovada	Implementada	2fdafaa
CR-042	PO	Create README.md for object detection project	16/11/2025	Aprovada	Implementada	dd5ffc4

ID	Origem	Descrição	Data da Solicitação	Decisão do CCB	Status	Hash
CR-043	PO	feat: Adiciona instalação do googletrans e atualiza script de execução com comandos de instalação de pacotes para...	16/11/2025	Aprovada	Implementada	d6c30cd
CR-044	PO	feat: Adiciona novos scripts para captura e processamento de imagens com ESP32-CAM, incluindo suporte a streaming e...	16/11/2025	Aprovada	Implementada	3ef875b
CR-045	PO	Merge remote-tracking branch ‘refs/remotes/origin/main’	17/11/2025	Aprovada	Implementada	6324c83
CR-046	PO	Add ESP32-CAM firmware with captive portal and camera configuration,	17/11/2025	Aprovada	Implementada	d057722
CR-047	PO	feat: Atualiza envio de detecções para servidor via HTTP POST e ajusta script de execução.	17/11/2025	Aprovada	Implementada	071250d
CR-048	PO	Merge remote-tracking branch ‘refs/remotes/origin/main’	17/11/2025	Aprovada	Implementada	3ab29c0
CR-049	PO	feat: Atualiza o módulo de motor de vibração com melhorias na configuração WiFi e mensagens de status, além de...	17/11/2025	Aprovada	Implementada	d029742
CR-050	PO	Merge branch ‘main’ of github.com:fabriobrasileiroo/sistema_de_pagamento_escolar_por_nfc	17/11/2025	Aprovada	Implementada	f0637dd
CR-051	PO	feat: Implement webcam capture functionality with realtime/manual movement and...	17/11/2025	Aprovada	Implementada	1d182f9
CR-052	PO	Add final model image to README	18/11/2025	Aprovada	Implementada	

5. Resumo do Status Atual

Área	Percentual Concluído	Observações
Firmware ESP32	95%	Comunicação BLE estável, calibração finalizada. Aguardando testes de QA.
Módulo IA (TFLite)	90%	Modelo leve implementado e integrado. Aguardando testes de QA.
Aplicativo Android	90%	Tela de calibração e logs concluídas. Exportação CSV implementada. Aguardando testes de QA.
Hardware Integrado	90%	Protótipo funcional testado em bancada.
Testes e QA	75%	Casos CT01–CT03 executados e aprovados. Início dos testes de integração (CT04–CT06).

6. Histórico de Versões e Auditorias

Data	Auditor	Escopo	Resultado	Ações Corretivas
05/10/2025	José Filho	Repositório Git (controle de branches)	Sem não-conformidades	—
18/10/2025	CCB	Requisitos vs Casos de Uso	100% rastreável	—
25/10/2025	QA	Plano de Testes	Corrigidos 2 desvios menores em CT03	—
02/11/2025	CCB	Status Accounting Review	Status OK	Nenhuma ação requerida
09/11/2025	QA	Product Baseline (PB) 1.0	Artefatos prontos para testes de aceitação.	Nenhuma ação requerida

7. Conclusão e Próximos Passos

- O projeto LUMI encontra-se na fase de Qualidade e Testes (QA), com o **Product Baseline (PB) 1.0** estabelecido em 09/11/2025.
- Todos os Itens de Configuração de software (CI-SW-002, CI-SW-003, CI-SW-004) atingiram a Versão 1.0 e estão sob controle de configuração.
- A próxima atualização do Status Accounting (v1.2) incluirá:
 - Resultados de teste de campo e testes de aceitação;
 - Atualização final do Product Baseline (PB) após aprovação do CCB;
 - CRs implementados e revalidados.

Previsão de auditoria final interna: 18/11/2025. **Entrega PoC:** 22/11/2025.

8. Assinaturas

Nome	Cargo	Assinatura	Data
José Filho	Configuration Manager	_____	09/11/2025
Heloisa Cativo	Scrum Master	_____	09/11/2025
Jean Carlos Figueiredo	Orientador / CCB Chair	_____	09/11/2025

ANÁLISE IMPACTO.xlsx - ANÁLISE IMPACTO

	File Size (bytes)		Path
VALIDA\303\207\303\203O.xlsx"	NaN		"docs/VALIDA\303\207\303\203O.xlsx"
VALIDA\303\207\303\203O.xlsx"	NaN		"docs/change-management/VALIDA\303\207\303\203O.xlsx"
settings.json	3.0		.vscode/settings.json
DR-conciencia-espacial-pdc-visual.pdf	602602.0		docs/DR-conciencia-espacial-pdc-visual.pdf
DR.pdf	Nan		docs/DR.pdf
ERS-conciencia-espacial-pdc-visual.pdf	1473624.0		docs/ERS-conciencia-espacial-pdc-visual.pdf
ERS.OculosEspecial.pdf	NaN		docs/ERS.OculosEspecial.pdf
ERS.pdf	NaN		docs/ERS.pdf
Manual_Equipe_OculosEspecial.pdf	133229.0		docs/Manual_Equipe_OculosEspecial.pdf
OculosEspecial_Relatorio-Consolidado_Final.pdf	NaN		docs/OculosEspecial_Relatorio-Consolidado_Final.pdf
Organograma_OculosEspecial.PNG	23210.0		docs/Organograma_OculosEspecial.PNG
Organograma_OculosEspecial.pdf	NaN		docs/Organograma_OculosEspecial.pdf
PLANEJAMENTO.xlsx	NaN		docs/PLANEJAMENTO.xlsx
PP-conciencia-espacial-pdc-visual.pdf	317487.0		docs/PP-conciencia-espacial-pdc-visual.pdf
PP.pdf	NaN		docs/PP.pdf
Passo_a_Passo_OculosEspecial.pdf	112180.0		docs/Passo_a_Passo_OculosEspecial.pdf
Relatorio-Consolidado_Final.pdf	706432.0		docs/Relatorio-Consolidado_Final.pdf
Relatorio_Consolidado_Final.docx	NaN		docs/Relatorio_Consolidado_Final.docx
TRIAGEM.xlsx	NaN		docs/TRIAGEM.xlsx
REQUISITOS.xlsx	10320.0		docs/change-management/REQUISITOS.xlsx
TRIAGEM.xlsx	12644.0		docs/change-management/TRIAGEM.xlsx
requisitos.csv	NaN		docs/requisitos.csv
requisitos.xlsx	NaN		docs/requisitos.xlsx
INTEGRACAO_SERVIDOR_SSE.cpp	NaN		firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp
INTEGRACAO_WEBSOCKET.cpp	NaN		firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp
WEBSOCKET_VS_HTTP.md	NaN		firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md
inovatech-2025.code-workspace	198.0	firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace	
main.cpp	28670.0	firmware/esp32-pai-mestre/src/main.cpp	
main.cpp	12809.0	firmware/modulo1-sensor/src/main.cpp	
EXECUTAR.md	3335.0	kaz-image-captioning/EXECUTAR.md	
INSTALL.md	7350.0	kaz-image-captioning/INSTALL.md	
README.md	14430.0	kaz-image-captioning/README.md	
dep.sh	18.0	kaz-image-captioning/dep.sh	
install.ps1	1618.0	kaz-image-captioning/install.ps1	
install.sh	2592.0	kaz-image-captioning/install.sh	
launcher.py	2089.0	kaz-image-captioning/launcher.py	
requirements-unified.txt	354.0	kaz-image-captioning/requirements-unified.txt	
requirements.txt	2031.0	kaz-image-captioning/requirements.txt	
run.sh	757.0	kaz-image-captioning/run.sh	
run_esp.sh	172.0	kaz-image-captioning/run_esp.sh	
run_webcam.sh	67.0	kaz-image-captioning/run_webcam.sh	
phone_camera_to_server.py	11923.0	kaz-image-captioning/src/phone_camera_to_server.py	
unified_camera_detection.py	24057.0	kaz-image-captioning/src/unified_camera_detection.py	
package.json	NaN	package.json	
README.md	6170.0	pcd-visual-app/README.md	
index.tsx	29657.0	pcd-visual-app/app/(tabs)/index.tsx	
settings.tsx	36280.0	pcd-visual-app/app/(tabs)/settings.tsx	
history-item-card.tsx	5747.0	pcd-visual-app/components/history-item-card.tsx	
icon-symbol.tsx	1458.0	pcd-visual-app/components/ui/icon-symbol.tsx	
AppContext.tsx	18859.0	pcd-visual-app/contexts/AppContext.tsx	
package.json	2522.0	pcd-visual-app/package.json	
tts-service.ts	1782.0	pcd-visual-app/services/tts-service.ts	
styles.ts	5402.0	pcd-visual-app/styles/styles.ts	
script_inicial.sh	1367.0	script_inicial.sh	

APROVAÇÃO CCB.xlsx - Aprovação CCB

APROVAÇÃO DO CHANGE CONTROL BOARD (CCB)			Unnamed: 1	Unnamed: 2	Unnamed: 3
Projeto:	NaN	NaN	NaN	NaN	NaN
Data da Aprovação:	2025-11-18 18:14:08		NaN	NaN	NaN
MEMBROS DO CCB (Contribuidores)	NaN	NaN	NaN	NaN	NaN
	NaN	NaN	NaN	NaN	NaN
Nome	# Commits	Papel	Status	Aprovação	
Fábio Henrique	23	Líder Técnico	Aprovado		
Fábio Henrique Mendes Brasileiro	22	Desenvolvedor	Aprovado		
fabiobrasileiroo	12	Desenvolvedor	Aprovado		
Mateus Miranda	9	Contribuidor	Aprovado		
heloisacativo	9	Contribuidor	Aprovado		
Mateus Mike	3	Contribuidor	Aprovado		
LevyAbreu	2	Contribuidor	Aprovado		
Gabriel	1	Contribuidor	Aprovado		

APROVAÇÃO CCB.xlsx - APROVAÇÃO CCB1

Empty DataFrame
Columns: []
Index: []

APROVAÇÃO CCB.xlsx - APROVAÇÃO CCB2

Empty DataFrame
Columns: []
Index: []

ENCERRAMENTO.xlsx - Encerramento

Unnamed: 0	ENCERRAMENTO DE MUDANÇAS	Unnamed: 2
NaN	NaN	NaN
NaN	Projeto: consciencia-espacial-PCD-visual	
NaN	Data de Encerramento:	2025-11-18 18:14:08
NaN	Versão Final App:	NaN
NaN	Versão Final Backend:	1.0.0
NaN	NaN	NaN
NaN	RESUMO EXECUTIVO	NaN
NaN	Total de Commits:	30
NaN	Total de Contribuidores:	8
NaN	Último Commit:	c2955ea em 2025-11-18
NaN	Branch:	main
NaN	NaN	NaN
PRINCIPAIS ENTREGAS	DESCRÍÇÃO	NaN
IC1	Sistema de detecção de objetos com ESP32-CAM	NaN
IC2	Backend WebSocket/SSE para comunicação em tempo real	NaN
IC3	App mobile React Native com acessibilidade	NaN
IC4	Firmware ESP32 com ESP-NOW para comunicação entre módulos	NaN
IC5	Sistema de captioning de imagens com IA (kaz-image-captioning)	NaN
IC6	Documentação técnica completa	NaN
IC7	Total de 30 commits implementados	NaN

ENCERRAMENTO.xlsx - ENCERRAMENTO1

Key
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\n

ENCERRAMENTO.xlsx - ENCERRAMENTO2

Key
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\n

PLANEJAMENTO.xlsx - Planejamento

PLANEJAMENTO DE

Último

A

Merge remote-tracking branch 'refs/remotes/orig
feat: Add status accounting report PDF
Add final model image t
feat: Implement webcam capture functionality with realtime/manu
Merge branch 'main' of github.com:fabiobrasileiroo/sistema_de_pagamento_escolar
feat: Atualiza o módulo de motor de vibração com melhorias na configuração WiFi e mensagens de status, além de adicionar arquivos .gitignore
Merge remote-tracking branch 'refs/remotes/orig
feat: Atualiza envio de detecções para servidor via HTTP POST e ajusta script de
Add ESP32-CAM firmware with captive portal and camera confi
Merge remote-tracking branch 'refs/remotes/orig

PLANEJAMENTO.xlsx - PLANEJAMENTO1

Key
Repository
Branch
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\nTags
Generated On

PLANEJAMENTO.xlsx - PLANEJAMENTO2

Key
Repository
Branch
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\nTags
Generated On

REQUISITOS.xlsx - requisitos

ID	Tipo	Descrição do Requisito (Resumo)	Caso de Uso Relacionado	Módulo
RF01	Funcional	Medição de Distância (uso do HC-SR04 em 200 ms).	UC04 - Medir Distância	Módulo
RF02	Funcional	Alerta Tátil Proporcional (vibração com intensidade proporcional à distância).	UC02 - Emitir Alerta Tátil	Módulo
RF03	Funcional	Captura de Imagem (a cada 100 ms se a distância for crítica).	UC07 - Capturar Imagem	Módulo
RF04	Funcional	Reconhecimento de Objetos (disponibilizar inferência local ou via App).	UC08 - Reconhecer Objeto	Módulo
RF05	Funcional	Comunicação com App (transferência de dados, timestamp, confidence via BLE).	UC09 - Transmitir Dados	Módulo
RF06	Funcional	TTS no App (mensagens de áudio correspondentes ao evento).	UC03 - Emitir Alerta Sonoro	Módulo
RF07	Funcional	Armazenamento Local (eventos/logs por 30 dias para análise offline).	UC10 - Armazenar Log	Módulo
RF08	Funcional	Controles por Hardware (botão liga/desliga, silêncio temporário, etc.).	UC11 - Controlar Dispositivo	Controle
RF09	Funcional	Falha de Processamento (notificar se a inferência local falhar).	UC12 - Notificar Falha	Módulo
RF10	Funcional	Atualização de Parâmetros via App (distância, sensibilidade, volume TTS).	UC05 - Calibrar Sensores Interface	Módulo
RF11	Funcional	Verificar dispositivos pelo App (status de conexão/bateria).	UC13 - Monitorar Status	Módulo de Controle
RF12	Funcional	Alerta de Bateria (ativação de alerta sonoro/TTS quando atingir nível crítico).	UC03 - Emitir Alerta Sonoro	Interface
RF13	Funcional	Confirmação de Inicialização (vibração + TTS "Sistema Ativo").	UC01 - Detectar Obstáculo	Módulo
RF14	Funcional	Modo de Silêncio Temporário (silenciar temporariamente todos os alertas).	UC11 - Controlar Dispositivo	Controlador
RF15	Funcional	Exportação de Logs de Teste (CSV/JSON para análise de desempenho offline).	UC10 - Armazenar Log	Módulo
RNF01	Não Funcional	Latência ponta-a-ponta (vibração < 400 ms).	—	Módulo
RNF02	Não Funcional	Acurácia mínima de 70% em ambiente controlado.	—	Módulo
RNF03	Não Funcional	Autonomia mínima de 4 horas de uso típico.	—	Habilidades
RNF04	Não Funcional	Interface de calibração acessível no App.	—	Interface de Controle
RNF Outros	Não Funcional	Disponibilidade: 99.9% durante a sessão de uso.	—	—
RNF Outros	Não Funcional	Dados de usuários e logs armazenados localmente.	—	—
RF01	Funcional	Medição de Distância (HC-SR04 em 200 ms)	UC04 - Medir Distância	Script
RF02	Funcional	Alerta Tátil Proporcional	UC02 - Emitir Alerta Tátil	Firmware
RF03	Funcional	Captura de Imagem (a cada 100 ms)	UC07 - Capturar Imagem	App
RF04	Funcional	Reconhecimento de Objetos	UC08 - Reconhecer Objeto	App
RF05	Funcional	Comunicação com App (BLE)	UC09 - Transmitir Dados	App Móvel
RF06	Funcional	TTS no App	UC03 - Emitir Alerta Sonoro	App
RF09	Funcional	Falha de Processamento	UC12 - Notificar Falha	App
RF10	Funcional	Atualização de Parâmetros via App	UC05 - Calibrar Sensores	App
RF11	Funcional	Verificar dispositivos pelo App	UC13 - Monitorar Status	App
RF12	Funcional	Alerta de Bateria	UC03 - Emitir Alerta Sonoro	Firmware
RF13	Funcional	Confirmação de Inicialização	UC01 - Detectar Obstáculo	App
RF14	Funcional	Modo de Silêncio Temporário	UC11 - Controlar Dispositivo	App
RF15	Funcional	Exportação de Logs de Teste	UC10 - Armazenar Log	App
RNF01	Não Funcional	Latência ponta-a-ponta (< 400 ms)	—	App
RNF02	Não Funcional	Acurácia mínima (70%)	—	App Móvel
RNF03	Funcional	[NOVO] Controle de Modo: Conexão WS	UC-09 - Transmitir Dados	App
RNF04	Funcional	[NOVO] Controle de Modo: Alternar para MANUAL	UC-11 - Controlar Dispositivo	App Móvel
RNF Outros	Funcional	[NOVO] Controle de Modo: Alternar para REALTIME	UC-11 - Controlar Dispositivo	App Móvel
RNF Outros	Funcional	[NOVO] Controle de Modo: Captura Manual	UC-07 - Capturar Imagem	App

REQUISITOS.xlsx - REQUISITOS1

Package Name	Version	Path
pcd-vision-backend	1.0.0	back-end/package.json
change-management-automation	0.1.0	docs/change-management copy/automation/package.json
nest-vision-api	1.0.0	nest-vision-api/package.json
pdc-visual-app	1.0.0	pcd-visual-app/package.json

REQUISITOS.xlsx - REQUISITOS2

Package Name	Version	Path
pcd-vision-backend	1.0.0	back-end/package.json
change-management-automation	0.1.0	docs/change-management copy/automation/package.json
nest-vision-api	1.0.0	nest-vision-api/package.json
pdc-visual-app	1.0.0	pcd-visual-app/package.json

RFC REQUESTS.xlsx - RFC REQUESTS

Hash	Date	Author
57acdb99b59de48bd6f4448d2b97333641f41eb6	2025-11-30T18:49:51	fabiobrasileiroo
08a17315f2b4fe4c7908dada0b069fe732ea0be4	2025-11-30T14:41:56	Mateus Mike
d1ab91f3956c3d143ac5c4326ebfe88c9967d659	2025-11-28T17:20:25	fabiobrasileiroo
8c223fef77a4f346b6b7179207a55b7ac5f06951	2025-11-27T16:23:20	fabiobrasileiroo
25608798d67417a86c30d07c4e92b93728a5ad4d	2025-11-27T16:17:59	fabiobrasileiroo
ef5b32a276575f9ef4f7895f3dde1a93eb27aae9	2025-11-27T16:17:40	fabiobrasileiroo
b319f21f68fcf11cc45f1e8d4b2d21c49ccb173d	2025-11-27T13:02:06	fabiobrasileiroo
67e28dd49ea4937751bf8be268abee3aeeec33f95	2025-11-27T02:55:58	fabiobrasileiroo
fe70cbf83fee2f643d89b26c1ca0e937c29ec191	2025-11-26T16:38:37	fabiobrasileiroo
1544531a68a4bb191c7c742b34022b5cf816c3f2	2025-11-26T11:42:57	fabiobrasileiroo
4bce5ad27dafe76b26642dcc08ac6e993598f03f	2025-11-24T18:26:48	fabiobrasileiroo
755ec7e061b01cd30ee0864876932ad9064e7b86	2025-11-23T18:51:16	Fábio Henrique Mendes Brasileiro
bb5d1947a9c18ab2b348a0928c346233eфе2ад4а	2025-11-23T18:30:13	Fábio Henrique Mendes Brasileiro
6a991bd14a74f4b24656425f7f68a679a1302b3c	2025-11-23T05:37:51	Fábio Henrique Mendes Brasileiro
1441c5acf11600ba936d2c17ff7fa6e760d667e	2025-11-23T05:07:31	Fábio Henrique Mendes Brasileiro
60cc9cd9af88a2dd07d38d17ef026627aba9026c	2025-11-23T05:07:18	Fábio Henrique Mendes Brasileiro
7e44687ad490b8d484fdb94cb260b3d90babad6	2025-11-23T03:49:58	Fábio Henrique Mendes Brasileiro
55e3e61c8f1487605792271c4b2633e38a1dc4696	2025-11-21T17:49:59	LevyAbreu
412cc136937209228e02e2c8527d55cc13b5151b	2025-11-19T15:34:17	Mateus Mike
a1dd6d6989505ad8439ae2c729b7a8f8e25c04f3	2025-11-19T14:54:04	Mateus Mike
02b567969f0cf8b975f727ad21a34a7a073ebbbf	2025-11-19T14:29:38	Mateus Mike
3f11ff0c23caf4d46c78f605f8c653e0fdb949de	2025-11-19T14:16:09	Mateus Mike
039ec2e6c21e90eba24ae09de27e44535674f416	2025-11-19T14:11:41	Mateus Mike
1735661b112f57d8140843204c9197e9d097b7c5	2025-11-19T13:57:33	Fábio Henrique Mendes Brasileiro
212f16e70910661aaф1cb5e2acec45c773d57928	2025-11-19T13:55:48	fabiobrasileiroo
161407acb267754f144c75a957a599b722e53e08	2025-11-19T13:55:16	fabiobrasileiroo
50c5c0aa969c31632ad69f199b6c1d8f5a920930	2025-11-18T23:47:27	LevyAbreu
1c59bcea0d0bdbbf5d27db79a1a11e9b9bc5034	2025-11-18T23:40:45	LevyAbreu
825fe0efcc6e6af6cbf570c2f3c574172c64692e	2025-11-18T22:22:42	fabiobrasileiroo
c2955ead6c50dd7cfeec584947bd07b625c4970e	2025-11-18T22:12:25	fabiobrasileiroo
0ad4c6b32c6f655b833bbca428bb3245cf346a6a	2025-11-18T22:12:08	fabiobrasileiroo
1d182f9b2b4e5eb6d7afac455f1e4040f60dce51	2025-11-18T20:12:26	Fábio Henrique Mendes Brasileiro
f0637ddeb6c466e03f06ba0823e75f9c1d29a11e	2025-11-17T17:53:17	fabiobrasileiroo
d029742ccb431e0b9f947bab82de0023cefb03a	2025-11-17T01:30:10	Fábio Henrique
3ab29c0f1513ce250ae702ddf3160b5fc72beb7b	2025-11-17T01:30:04	Fábio Henrique
071250df63a2c15c2582db34e92a79e203586881	2025-11-17T01:13:47	Fábio Henrique Mendes Brasileiro
d0577221d1cd5d0085b9683fd69af224c607d37a	2025-11-17T01:13:16	Fábio Henrique Mendes Brasileiro
6324c83715342e75c579afa38edfc5cc11d746c0	2025-11-17T00:01:09	Fábio Henrique
3ef875b243d257fc5555da27503992d6294e7a49	2025-11-16T23:53:17	Fábio Henrique Mendes Brasileiro
d6c30cddd4dc24c62e76e15dfad3d3bb0c06d9bd	2025-11-16T23:52:58	Fábio Henrique Mendes Brasileiro
dd5ffc45e0437ffe7d7e19aec58f6cd7cf8fb5c	2025-11-16T23:52:27	Fábio Henrique Mendes Brasileiro
2fdafaafc24540d2c5ff2cd6f2decdd60ba9b0cf	2025-11-16T22:26:03	Fábio Henrique Mendes Brasileiro
a4f1ce26a065473fee25334392ca75cf9a8f8f50	2025-11-16T17:52:54	Fábio Henrique Mendes Brasileiro
16f408809564082375dc840ce41e0699e7e2f56b	2025-11-16T17:52:40	Fábio Henrique Mendes Brasileiro
2e801b89b368cc66e123a35534bb5d0639c34eba	2025-11-16T16:12:11	Fábio Henrique Mendes Brasileiro
606b2089d3f683fa57b04c8d64958639df6ad838	2025-11-16T16:11:23	Fábio Henrique Mendes Brasileiro
22c0219feb4d6d85d348b264ea40a3d52abc0958	2025-11-16T15:56:06	Fábio Henrique Mendes Brasileiro
7198f126a0f633b61db0071dde3ceee53b12e97e	2025-11-16T15:20:35	Fábio Henrique feat: Add SSE and WebSocket test script
646844be2d4ae02f9d6ce505166e87787cd8341b	2025-11-16T01:10:29	heilosacativo
a3559934a04d9b99862d7377a509ef2d75a24cc8	2025-11-15T18:40:59	Fábio Henrique
9d7d2aa20573ca9f48492fb5c106dbfb1c1dbdb3c	2025-11-15T18:39:05	Fábio Henrique
a2e0a3bac851cb5ad11e84c6561c5a959c219814	2025-11-14T21:28:12	fabiobrasileiroo
5a15e29e2f5e3de64b11dc50cac7862e5797ccb0	2025-11-13T15:15:34	fabiobrasileiroo
4dc58e828c2b81e6e3823a9072174d7ac5b847e	2025-11-13T15:07:39	fabiobrasileiroo
dbeed151791bc94cd9cac512c8bd4cd24b9d67ea	2025-11-12T17:46:36	Mateus Mike
c1875d3879a4a547050a1caa09856214815a5fa0	2025-11-12T17:43:47	Mateus Mike
b2159067b8a7b8282ff0880ae47d73d0a0fa520d	2025-11-12T17:14:07	Mateus Mike
c7cb6b24fe99493c42905a30fc33e454f2bc3e7	2025-11-11T15:50:24	Mateus Miranda
b4b37facdd2e5ba6ab136657a195346ebaebb674	2025-11-11T15:50:11	Mateus Miranda
b84513e2276ac247fb5785747941c94ce8ff95c9	2025-11-11T15:49:25	Mateus Miranda
4fe048b5a287b7d68ecc42e0a402f336fad0fa61	2025-11-11T15:47:57	Mateus Miranda
4de820ecb2ea664389cf78b20b0d96cb65270a0	2025-11-11T15:32:23	Mateus Miranda
2bf3c754c376da0cd4f895983bc11522aed3656f	2025-11-11T15:22:46	Mateus Miranda
fe86091d910afac61f81b4400a221c982d549f6c	2025-11-04T20:58:11	Mateus Miranda
fc9b57b09765ce0b472e0c5a51fcc3e8ce957f1e	2025-11-04T20:57:42	Mateus Miranda
b7ceaf3e3bcab4a1a2b9f846695ee73514c9b4a8	2025-11-04T20:56:58	Mateus Miranda
eedeb68dde8f915c300184830bb3d152552fb1ce0	2025-11-03T17:30:31	Fábio Henrique Mendes Brasileiro

9ceef2c0de571a7d509394129d25ae759003b671 2025-11-03T17:30:00 Fábio Henrique Mendes Brasileiro
d915adcdfad71e39d07b33670b12c92970731f16 2025-11-03T17:29:20 Fábio Henrique Mendes Brasileiro
7209e8acaeea20c42614e5dc0929dcad94d1c56b 2025-11-03T11:48:06 Fábio Henrique Mendes Brasileiro
452ec6023e750968cad4423f77e88cb48e772b9c 2025-11-01T23:23:46 Fábio Henrique
1e44bb9cdac5df45bb1dc1af3e98c17e7eb6e34e 2025-11-01T23:20:18 Fábio Henrique
e206c7fe7a719dd0102f64193696586759fe35a3 2025-11-01T23:00:34 Fábio Henrique
7ef27155e9f023be7f763fda88379043b0c26408 2025-11-01T22:49:17 Fábio Henrique
605027acba54521b7b45b3fa7ee1495aa96106c6 2025-11-01T17:57:41 Fábio Henrique
7e5bb7f2d2e2e2a3aa44712bd2e772c1777889d1 2025-11-01T16:57:07 Fábio Henrique
ade8c69432a9538a96b7f01221e8bd86f21f75a0 2025-11-01T16:26:23 Fábio Henrique Mendes Brasileiro
4eda64d20b6a8f0df9f41e3037cc55c12f3d1e4e 2025-11-01T16:24:34 Fábio Henrique
5161a725d15eaff46346fdcff8bba89617be2238 2025-11-01T16:15:46 Fábio Henrique
f6b60ef9088d7b86aa4a5e7dba2e917b4ef38940 2025-10-31T20:03:34 fabiobrasileiroo
936b497523036181ca3882a087f5f50d5d3f65f0 2025-10-30T01:20:43 Fábio Henrique
58048dfdd47bfa7916b1a51e0943f7b35d7dab3f 2025-10-30T00:42:24 Fábio Henrique
f4ff1ccb4e27ebd382d17cf237fb80b3aba71f4 2025-10-29T23:40:54 Fábio Henrique
6e6b8d2845b41f4d47cb59a3558da36858ce1efb 2025-10-28T23:32:38 LevyAbreu
2a17d03107ac47c7ad0ae2e480d43c9b6c0eec5 2025-10-28T23:19:00 LevyAbreu
6015feb682728616b8dd99c067f645b4ce391304 2025-10-28T17:44:09 Fábio Henrique Mendes Brasileiro
4aa4aff2c938d99ffdad58774bf8fd691cddc10d 2025-10-28T17:06:51 fabiobrasileiroo
428e4c7a35e76d92afd28bac09b9177b89de9741 2025-10-28T16:57:39 fabiobrasileiroo
60e720dfc3294548f960f3c4e0c4dbc2037733b1 2025-10-28T16:26:50 fabiobrasileiroo
00f5113208f6da039d574682c783c214aefd73cf 2025-10-26T21:38:41 Fábio Henrique
ccbf669c51e175550149359797c5563afcc3007c 2025-10-26T21:36:12 Fábio Henrique
2e68b5ee7cad2281e06f9231d963e8cd29eba39 2025-10-23T17:35:46 Fábio Henrique Mendes Brasileiro
ad8db634b4fa033dcad569203a31f3c09fee4c41 2025-10-21T21:25:32 fabiobrasileiroo
2885dee100a8bacf2532bc609939a2f79d09d3a4 2025-10-21T15:32:20 fabiobrasileiroo
f34276c0183e55955b4cb7385eb44577ac5cc14c 2025-10-13T00:26:45 Fábio Henrique
2e4b8b9fabf20553abff752fc247f11be913fb43 2025-10-13T00:26:34 Fábio Henrique
fd18058311ed027af40d29c2793199090bb42888 2025-10-13T00:05:00 Fábio Henrique Mendes Brasileiro
1fb8f2198f212e132aa3c1d2dabafcfa9fcc4829 2025-10-12T23:58:55 Fábio Henrique
ed8cab6f8e8c9e86aef3169875d8acc5ed077522 2025-09-21T00:08:46 Fábio Henrique
37abf1dfabb48d902e001a2af7d67c60637e9ed5 2025-09-06T18:34:31 Fábio Henrique Mendes Brasileiro

TRIAGEM.xlsx - Triagem

TRIAGEM DE MUDANÇAS

NaN

Projeto:

NaN

COMMITS RECENTES PARA TRIAGEM

NaN

SHA

c2955ea

0ad4c6b

1d182f9

f0637dd

d029742

3ab29c0

071250d

d057722

6324c83

3ef875b

d6c30cd

dd5ffc4

2fdafaa

a4f1ce2

16f4088 Refactor Settings Screen: Replace emoji icons with Lucide icons, enhance loading state with SkeletonLoader, and up

TRIAGEM.xlsx - TRIAGEM1

Changed Files (30 days)

"docs/VALIDA\303\207\303\203O.xlsx"

"docs/change-management/VALIDA\303\207\303\203O.xlsx"

.vscode/settings.json

docs/DR-conciencia-espacial-pdc-visual.pdf

docs/DR.pdf

docs/ERS-conciencia-espacial-pdc-visual.pdf

docs/ERS.OculosEspecial.pdf

docs/ERS.pdf

docs/Manual_Equipe_OculosEspecial.pdf

docs/OculosEspecial_Relatorio-Consolidado_Final.pdf

docs/Organograma_OculosEspecial.PNG

docs/Organograma_OculosEspecial.pdf

docs/PLANEJAMENTO.xlsx

docs/PP-conciencia-espacial-pdc-visual.pdf

docs/PP.pdf

docs/Passo_a_Passo_OculosEspecial.pdf

docs/Relatorio-Consolidado_Final.pdf

docs/Relatorio_Consolidado_Final.docx

docs/TRIAGEM.xlsx

docs/change-management/REQUISITOS.xlsx

docs/change-management/TRIAGEM.xlsx

docs/requisitos.csv

docs/requisitos.xlsx

firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp

firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp

firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md

firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace

firmware/esp32-pai-mestre/src/main.cpp

firmware/modulo1-sensor/src/main.cpp

kaz-image-captioning/EXECUTAR.md

kaz-image-captioning/INSTALL.md

kaz-image-captioning/README.md

kaz-image-captioning/dep.sh

kaz-image-captioning/install.ps1

kaz-image-captioning/install.sh

kaz-image-captioning/launcher.py

kaz-image-captioning/requirements-unified.txt

kaz-image-captioning/requirements.txt

kaz-image-captioning/run.sh

kaz-image-captioning/run_esp.sh

kaz-image-captioning/run_webcam.sh

kaz-image-captioning/src/phone_camera_to_server.py

```
kaz-image-captioning/src/unified_camera_detection.py  
    package.json  
        pcd-visual-app/README.md  
        pcd-visual-app/app/(tabs)/index.tsx  
        pcd-visual-app/app/(tabs)/settings.tsx  
pcd-visual-app/components/history-item-card.tsx  
    pcd-visual-app/components/ui/icon-symbol.tsx  
    pcd-visual-app/contexts/AppContext.tsx  
        pcd-visual-app/package.json  
    pcd-visual-app/services/tts-service.ts  
    pcd-visual-app/styles/styles.ts  
    script_inicial.sh
```

TRIAGEM.xlsx - TRIAGEM2

```
    Changed Files (30 days)  
    "docs/VALIDA\303\207\303\203O.xlsx"  
"docs/change-management/VALIDA\303\207\303\203O.xlsx"  
    .vscode/settings.json  
docs/DR-conciencia-espacial-pdc-visual.pdf  
    docs/DR.pdf  
docs/ERS-conciencia-espacial-pdc-visual.pdf  
    docs/ERS.OculosEspecial.pdf  
    docs/ERS.pdf  
docs/Manual_Equipe_OculosEspecial.pdf  
docs/OculosEspecial_Relatorio-Consolidado_Final.pdf  
    docs/Organograma_OculosEspecial.PNG  
    docs/Organograma_OculosEspecial.pdf  
    docs/PLANEJAMENTO.xlsx  
docs/PP-conciencia-espacial-pdc-visual.pdf  
    docs/PP.pdf  
docs/Passo_a_Passo_OculosEspecial.pdf  
docs/Relatorio-Consolidado_Final.pdf  
docs/Relatorio_Consolidado_Final.docx  
    docs/TRIAGEM.xlsx  
docs/change-management/REQUISITOS.xlsx  
    docs/change-management/TRIAGEM.xlsx  
        docs/requisitos.csv  
        docs/requisitos.xlsx  
firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp  
firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp  
firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md  
firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace  
    firmware/esp32-pai-mestre/src/main.cpp  
    firmware/modulo1-sensor/src/main.cpp  
        kaz-image-captioning/EXECUTAR.md  
        kaz-image-captioning/INSTALL.md  
        kaz-image-captioning/README.md  
        kaz-image-captioning/dep.sh  
    kaz-image-captioning/install.ps1  
    kaz-image-captioning/install.sh  
    kaz-image-captioning/launcher.py  
kaz-image-captioning/requirements-unified.txt  
    kaz-image-captioning/requirements.txt  
    kaz-image-captioning/run.sh  
    kaz-image-captioning/run_esp.sh  
    kaz-image-captioning/run_webcam.sh  
kaz-image-captioning/src/phone_camera_to_server.py  
kaz-image-captioning/src/unified_camera_detection.py  
    package.json  
        pcd-visual-app/README.md  
        pcd-visual-app/app/(tabs)/index.tsx  
        pcd-visual-app/app/(tabs)/settings.tsx  
pcd-visual-app/components/history-item-card.tsx  
    pcd-visual-app/components/ui/icon-symbol.tsx  
    pcd-visual-app/contexts/AppContext.tsx  
        pcd-visual-app/package.json  
    pcd-visual-app/services/tts-service.ts  
    pcd-visual-app/styles/styles.ts  
    script_inicial.sh
```

VALIDAÇÃO.xlsx - Validação

VALIDAÇÃO DE MUDANÇAS		Unnamed: 1		Unnamed: 2			Unnamed: 3	Unnamed: 4
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Projeto: consciencia-espacial-PCD-visual								
Ambiente:	Produção		NaN	NaN	NaN		NaN	
NaN	NaN	NaN	NaN	NaN	NaN		NaN	
TESTES DE VALIDAÇÃO		NaN		NaN	NaN	NaN		NaN
NaN	NaN		NaN	NaN	NaN		NaN	
Módulo	Teste	Critério	Resultado				Observações	
back-end	Testes de integração WebSocket	Conexões	ESP32 e App funcionando	Passou	API funcionando corretamente			
back-end	Teste de SSE	Eventos enviados corretamente	Passou	Streaming	de eventos OK			
pdc-visual-app	Testes E2E mobile	Todas as features	funcionando	Passou	App responsivo e funcional			
firmware	Teste de comunicação ESP-NOW	Módulos se comunicando	Passou		Latência < 100ms			
firmware	Teste sensor HC-SR04	Leitura de distância precisa	Passou	Precisão de ±2cm				
kaz-image-captioning	Teste de inferência	Captions geradas corretamente	Passou	Modelo v2	funcionando			

VALIDAÇÃO.xlsx - VALIDAÇÃO1

Package HasTestScript	
pcd-vision-backend	yes
change-management-automation	no
nest-vision-api	yes
pdc-visual-app	yes

VALIDAÇÃO.xlsx - VALIDAÇÃO2

Package HasTestScript	
pcd-vision-backend	yes
change-management-automation	no
nest-vision-api	yes
pdc-visual-app	yes

README.md

Change Management Automation

Este script gera um arquivo Excel (CHANGE_MANAGEMENT_AUTOGENERATED.xlsx) na pasta docs/change-management com informações extraídas do repositório e do Git.

Planilhas criadas:

- PLANEJAMENTO: Visão geral do repositório e branch
- REQUISITOS: Lista de pacotes (package.json encontrados)
- RFC REQUESTS: Lista dos últimos commits
- TRIAGEM: Arquivos modificados nos últimos 30 dias
- APROVAÇÃO CCB: Lista de contribuidores e e-mails
- ANÁLISE IMPACTO: Arquivos alterados e seus tamanhos
- VALIDAÇÃO: Indicador de existência do script test nos pacotes
- ENCERRAMENTO: Informações de fechamento (último commit)

Como usar:

- 1) Instalar dependências:

```
bash  
cd docs/change-management/automation  
npm install
```

- 2) Executar:

```
```bash  
npm run generate
opcional: sobrescrever os templates existentes com --overwrite
node generate_change_management.js --overwrite
```
```

Notas:

- O script deve ser executado a partir do repositório com git disponível e histórico de commits.
- A flag --overwrite irá sobrescrever os arquivos de planilhas existentes em docs/change-management com os dados gerados.

generate_change_management.py

```
#!/usr/bin/env python3
"""
generate_change_management.py
Automação em Python para gerar o Excel de Change Management com dados do repositório
"""

import argparse
import json
import os
import sys
from datetime import datetime, timedelta
from pathlib import Path
import subprocess

import pandas as pd
from git import Repo
from openpyxl import load_workbook

def find_package_json(root: Path, ignore_dirs=None):
    if ignore_dirs is None:
        ignore_dirs = {"node_modules", "venv", ".git", ".venv", ".cache", "dist", "build"}
    results = []
    for dirpath, dirnames, filenames in os.walk(root):
        # skip ignored dirs
        parts = Path(dirpath).parts
        if any(p in ignore_dirs for p in parts):
            continue
        if 'package.json' in filenames:
            full = Path(dirpath) / 'package.json'
            # skip the generator package.json itself
            if 'docs/change-management/automation' not in str(full):
                results.append(full)
    return results

def read_package_json(path: Path):
    try:
        return json.loads(path.read_text(encoding='utf8'))
    except Exception:
        return None

def gather_data(repo_root: Path, since_days: int = 30, max_commits: int = 200):
    repo = Repo(str(repo_root))
    if repo.bare:
        raise RuntimeError('Repository is bare or not a Git repo')

    branch = repo.active_branch.name if repo.head.is_valid() else 'HEAD'
    latest_commit = repo.head.commit

    # Contributors via shortlog
    try:
        shortlog = repo.git.shortlog('-sne')
        contributors = []
        for line in shortlog.splitlines():
            line = line.strip()
            if not line:
                continue
            count, rest = line.split('\t', 1) if '\t' in line else (None, line)
            if count is None:
                parts = line.split(' ', 1)
                count = parts[0]
                rest = parts[1] if len(parts) > 1 else ''
                import re
                m = re.match(r"(?P<name>.+) <(?P<email>.+)>", rest)
                if m:
                    contributors.append({'commits': count.strip(), 'name': m.group('name').strip(), 'email': m.group('email').strip()})
            else:
                contributors.append({'commits': count.strip(), 'name': rest.strip(), 'email': ''})
    except Exception:
        contributors = []
    commits = []

    return {
        'contributors': contributors,
        'commits': commits
    }
```

```

for i, c in enumerate(repo.iter_commits(max_count=max_commits)):
    commits.append({'hash': c.hexsha, 'date': datetime.utcnow().fromtimestamp(c.committed_date).isoformat(), 'message': c.message.strip(), 'author': c.author.name, 'email': c.author.email})

tags = [t.name for t in repo.tags]

package_files = find_package_json(repo_root)
packages = []
for p in package_files:
    pj = read_package_json(p)
    if pj is None:
        continue
    packages.append({'name': pj.get('name', Path(p).parent.name), 'version': pj.get('version', ''), 'path': str(p.relative_to(repo_root)), 'dependencies': pj.get('dependencies', {}), 'devDependencies': pj.get('devDependencies', {})})

since_date = datetime.utcnow() - timedelta(days=since_days)
changed_files_set = set()
for c in repo.iter_commits(since=since_date.isoformat()):
    for f in c.stats.files.keys():
        changed_files_set.add(f)
changed_files = sorted(changed_files_set)

return {
    'repoName': repo_root.name,
    'branch': branch,
    'latest': {'hash': latest_commit.hexsha, 'date': datetime.utcnow().fromtimestamp(latest_commit.committed_date).isoformat(), 'author': latest_commit.author.name, 'message': latest_commit.message.strip()},
    'contributors': contributors,
    'commits': commits,
    'tags': tags,
    'packages': packages,
    'changedFiles': changed_files,
}
}

def write_xlsx(data, out_file: Path, repo_root: Path, overwrite_templates: bool = False):
    # Create sheets using pandas
    sheets = {}
    # PLANEJAMENTO
    planning = [
        ['Repository', data['repoName']],
        ['Branch', data['branch']],
        ['Latest Commit', data['latest'].get('hash', "")],
        ['Latest Commit Date', data['latest'].get('date', "")],
        ['Latest Commit Author', data['latest'].get('author', "")],
        ['Latest Commit Message', data['latest'].get('message', "")],
        ['Tags', ', '.join(data['tags'])],
        ['Generated On', datetime.utcnow().isoformat()],
    ]
    sheets['PLANEJAMENTO'] = pd.DataFrame(planning, columns=['Key', 'Value'])

    # REQUISITOS
    req_df = pd.DataFrame([{'Package Name': p['name'], 'Version': p['version'], 'Path': p['path']} for p in data['packages']])
    sheets['REQUISITOS'] = req_df

    # RFC REQUESTS
    rfc_df = pd.DataFrame(data['commits'])
    if rfc_df.empty:
        rfc_df = pd.DataFrame(columns=['Hash', 'Date', 'Author', 'Message'])
    else:
        rfc_df = rfc_df[['hash', 'date', 'author', 'message']]
        rfc_df.columns = ['Hash', 'Date', 'Author', 'Message']
    sheets['RFC REQUESTS'] = rfc_df

    # TRIAGEM
    triage_df = pd.DataFrame({'Changed Files (30 days)': data['changedFiles']})
    sheets['TRIAGEM'] = triage_df

    # APROVAÇÃO CCB
    ccb_df = pd.DataFrame(data['contributors'])
    if ccb_df.empty:
        ccb_df = pd.DataFrame(columns=['Name', 'Email', 'Commits'])
    else:
        ccb_df = ccb_df[['Name', 'Email', 'Commits']]
    sheets['APROVAÇÃO CCB'] = ccb_df

```

```

ccb_df = ccb_df[['name', 'email', 'commits']]
ccb_df.columns = ['Name', 'Email', 'Commits']
sheets['APROVAÇÃO CCB'] = ccb_df

# ANÁLISE IMPACTO
impact_rows = []
for f in data['changedFiles']:
    full_path = repo_root / f
    try:
        # compute size if exists
        size = full_path.stat().st_size if full_path.exists() else None
    except Exception:
        size = None
    impact_rows.append({'File': Path(f).name, 'Size (bytes)': size, 'Path': f})
impact_df = pd.DataFrame(impact_rows)
sheets['ANÁLISE IMPACTO'] = impact_df

# VALIDAÇÃO
validation_rows = []
for p in data['packages']:
    pj_full = repo_root / p['path']
    try:
        pj = read_package_json(pj_full) if pj_full.exists() else None
        has_test = 'yes' if pj and pj.get('scripts', {}).get('test') else 'no'
    except Exception:
        has_test = 'unknown'
    validation_rows.append({'Package': p['name'], 'HasTestScript': has_test})
sheets['VALIDAÇÃO'] = pd.DataFrame(validation_rows)

# ENCERRAMENTO
closure = [
    ['Latest Commit', data['latest'].get('hash', "")],
    ['Latest Commit Date', data['latest'].get('date', "")],
    ['Latest Commit Author', data['latest'].get('author', "")],
    ['Latest Commit Message', data['latest'].get('message', "")],
]
sheets['ENCERRAMENTO'] = pd.DataFrame(closure, columns=['Key', 'Value'])

# Write to Excel with pandas ExcelWriter
with pd.ExcelWriter(out_file, engine='openpyxl') as writer:
    for name, df in sheets.items():
        df.to_excel(writer, sheet_name=name[:31], index=False)

print('Wrote', out_file)

# Optionally overwrite templates (create separate XLSX per sheet)
if overwrite_templates:
    folder = out_file.parent
    for name, df in sheets.items():
        sub_file = folder / f'{name}.xlsx'
        with pd.ExcelWriter(sub_file, engine='openpyxl') as writer:
            df.to_excel(writer, sheet_name=name[:31], index=False)
        print('Overwrote', sub_file)
return sheets

def update_existing_templates(sheets: dict, change_mgmt_dir: Path, out_file: Path, overwrite_all: bool = False, create_backups: bool = True):
    """Update existing .xlsx files in change_mgmt_dir.

    - If file name is `CHANGE_MANAGEMENT_AUTOGENERATED.xlsx`, replace it with the full workbook
    - If file stem matches a sheet name, overwrite that workbook with the corresponding single sheet
    - If overwrite_all is True, replace all .xlsx with full workbook
    """
    # Search recursively in the folder but filter out node_modules, .venv, etc
    ignore_dirs = {'node_modules', '.venv', 'venv', '.cache', '.git'}
    all_xlsx = []
    for p in change_mgmt_dir.rglob('*xlsx'):
        if any(x in p.parts for x in ignore_dirs):
            continue
        all_xlsx.append(p)
    if not all_xlsx:
        print('No xlsx templates found in', change_mgmt_dir)
        return

    import unicodedata

```

```

def normalize(s: str):
    s = s.lower().strip()
    s = ''.join(ch for ch in unicodedata.normalize('NFD', s) if unicodedata.category(ch) != 'Mn')
)
    s = s.replace(' ', '_').replace('-', '_').replace('_', '')
    return s

# Create normalized map for sheet names
norm_sheets = {normalize(name): name for name in sheets.keys()}

for f in all_xlsx:
    try:
        stem = f.stem
        nstem = normalize(stem)
        # if file equals full generated workbook, re-write it
        if f == out_file or stem == out_file.stem:
            print('Overwriting full generated file', f)
        if create_backups and f.exists():
            bak_name = f.with_suffix(f.suffix + f'.bak.{int(datetime.now().timestamp())}')
            try:
                f.replace(bak_name)
            except Exception:
                try:
                    import shutil
                    shutil.copy2(f, bak_name)
                except Exception:
                    pass
        with pd.ExcelWriter(f, engine='openpyxl') as writer:
            for name, df in sheets.items():
                df.to_excel(writer, sheet_name=name[:31], index=False)
            continue

        # otherwise, open workbook and update matching sheets
        try:
            wb = load_workbook(f)
        except Exception:
            print('Skipping unreadable workbook', f)
            continue

        modified = False
        for name, df in sheets.items():
            if name in wb.sheetnames or normalize(name) in (normalize(s) for s in wb.sheetnames):
:
                print('Replacing sheet', name, 'in', f)
                # remove existing sheet
                try:
                    std = wb[name]
                    wb.remove(std)
                except Exception:
                    pass
                # create a new sheet and append data
                ws = wb.create_sheet(title=name[:31])
                # header
                if not df.empty:
                    ws.append(list(df.columns))
                    for row in df.itertuples(index=False, name=None):
                        ws.append(list(row))
                modified = True
            # if file didn't contain matching sheets but overwrite_all asked, write full workbook
        if modified:
            wb.save(f)
            print('Updated existing workbook', f)
        else:
            # if the file name matches one of our templates, write single sheet to file
            if nstem in norm_sheets:
                sname = norm_sheets[nstem]
                df = sheets[sname]
                print('Updating file by single sheet', f, '=>', sname)
                if create_backups and f.exists():
                    bak_name = f.with_suffix(f.suffix + f'.bak.{int(datetime.now().timestamp())}')
)
                try:
                    f.replace(bak_name)
                except Exception:
                    import shutil

```

```

        shutil.copy2(f, bak_name)
    with pd.ExcelWriter(f, engine='openpyxl') as writer:
        df.to_excel(writer, sheet_name=sname[:31], index=False)
    continue
if overwrite_all:
    print('Overwrite-all: writing full workbook to', f)
    if create_backups and f.exists():
        bak_name = f.with_suffix(f.suffix + f'.bak.{int(datetime.now().timestamp())}')
')
    try:
        f.replace(bak_name)
    except Exception:
        import shutil
        shutil.copy2(f, bak_name)
    with pd.ExcelWriter(f, engine='openpyxl') as writer:
        for name, df in sheets.items():
            df.to_excel(writer, sheet_name=name[:31], index=False)
    else:
        print('Skipping', f, '- no matching sheet and overwrite_all not set')
except Exception as e:
    print('Failed updating', f, e)

def main():
    parser = argparse.ArgumentParser(description='Generate change management xlsx from repo data')
    parser.add_argument('--out', '-o', default='CHANGE_MANAGEMENT_AUTOGENERATED.xlsx', help='XLSX filename')
    parser.add_argument('--since-days', '-s', type=int, default=30, help='Number of days to look for changed files')
    parser.add_argument('--max-commits', '-m', type=int, default=200, help='Max commits to include')
    parser.add_argument('--overwrite', action='store_true', help='Overwrite template files per sheet')
    parser.add_argument('--update-templates', action='store_true', help='Update all .xlsx templates present in docs/change-management folder')
    parser.add_argument('--update-all', action='store_true', help='Recursively update all .xlsx files found in the repository (excluding node_modules/.venv)')
    parser.add_argument('--no-backups', action='store_true', help="Don't create backups of overwritten files (default: create backups)")
    args = parser.parse_args()

    repo_root = Path(__file__).resolve().parent.parent.parent.parent.parent
    out_file = Path(__file__).resolve().parent / args.out
    print('Repo root:', repo_root)
    data = gather_data(repo_root, since_days=args.since_days, max_commits=args.max_commits)
    sheets = write_xlsx(data, out_file, repo_root, overwrite_templates=args.overwrite)
    # Update existing templates in docs/change-management
    change_mgmt_dir = Path(__file__).resolve().parent.parent
    if args.update_templates:
        update_existing_templates(sheets, change_mgmt_dir, out_file, overwrite_all=args.overwrite, create_backups=not args.no_backups)
    if args.update_all:
        print('Updating all xlsx files in repo root (this may take time)...')
        update_existing_templates(sheets, repo_root, out_file, overwrite_all=args.overwrite, create_backups=not args.no_backups)

if __name__ == '__main__':
    main()

```

requirements.txt

GitPython>=3.1.30
pandas>=2.2.2
openpyxl>=3.1.2
reportlab>=4.0
pypdf>=3.9
Pillow>=9.5
markdown>=3.4

ANÁLISE IMPACTO.xlsx - ANÁLISE IMPACTO

| | File Size (bytes) | | Path |
|--|-------------------|---|---|
| VALIDA\303\207\303\203O.xlsx" | NaN | | "docs/VALIDA\303\207\303\203O.xlsx" |
| VALIDA\303\207\303\203O.xlsx" | NaN | | "docs/change-management/VALIDA\303\207\303\203O.xlsx" |
| settings.json | 3.0 | | .vscode/settings.json |
| DR-conciencia-espacial-pdc-visual.pdf | 602602.0 | | docs/DR-conciencia-espacial-pdc-visual.pdf |
| DR.pdf | Nan | | docs/DR.pdf |
| ERS-conciencia-espacial-pdc-visual.pdf | 1473624.0 | | docs/ERS-conciencia-espacial-pdc-visual.pdf |
| ERS.OculosEspecial.pdf | NaN | | docs/ERS.OculosEspecial.pdf |
| ERS.pdf | NaN | | docs/ERS.pdf |
| Manual_Equipe_OculosEspecial.pdf | 133229.0 | | docs/Manual_Equipe_OculosEspecial.pdf |
| OculosEspecial_Relatorio-Consolidado_Final.pdf | NaN | | docs/OculosEspecial_Relatorio-Consolidado_Final.pdf |
| Organograma_OculosEspecial.PNG | 23210.0 | | docs/Organograma_OculosEspecial.PNG |
| Organograma_OculosEspecial.pdf | NaN | | docs/Organograma_OculosEspecial.pdf |
| PLANEJAMENTO.xlsx | NaN | | docs/PLANEJAMENTO.xlsx |
| PP-conciencia-espacial-pdc-visual.pdf | 317487.0 | | docs/PP-conciencia-espacial-pdc-visual.pdf |
| PP.pdf | NaN | | docs/PP.pdf |
| Passo_a_Passo_OculosEspecial.pdf | 112180.0 | | docs/Passo_a_Passo_OculosEspecial.pdf |
| Relatorio-Consolidado_Final.pdf | 706432.0 | | docs/Relatorio-Consolidado_Final.pdf |
| Relatorio_Consolidado_Final.docx | NaN | | docs/Relatorio_Consolidado_Final.docx |
| TRIAGEM.xlsx | NaN | | docs/TRIAGEM.xlsx |
| REQUISITOS.xlsx | 10320.0 | | docs/change-management/REQUISITOS.xlsx |
| TRIAGEM.xlsx | 12644.0 | | docs/change-management/TRIAGEM.xlsx |
| requisitos.csv | NaN | | docs/requisitos.csv |
| requisitos.xlsx | NaN | | docs/requisitos.xlsx |
| INTEGRACAO_SERVIDOR_SSE.cpp | NaN | | firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp |
| INTEGRACAO_WEBSOCKET.cpp | NaN | | firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp |
| WEBSOCKET_VS_HTTP.md | NaN | | firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md |
| inovatech-2025.code-workspace | 198.0 | firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace | |
| main.cpp | 28670.0 | firmware/esp32-pai-mestre/src/main.cpp | |
| main.cpp | 12809.0 | firmware/modulo1-sensor/src/main.cpp | |
| EXECUTAR.md | 3335.0 | kaz-image-captioning/EXECUTAR.md | |
| INSTALL.md | 7350.0 | kaz-image-captioning/INSTALL.md | |
| README.md | 14430.0 | kaz-image-captioning/README.md | |
| dep.sh | 18.0 | kaz-image-captioning/dep.sh | |
| install.ps1 | 1618.0 | kaz-image-captioning/install.ps1 | |
| install.sh | 2592.0 | kaz-image-captioning/install.sh | |
| launcher.py | 2089.0 | kaz-image-captioning/launcher.py | |
| requirements-unified.txt | 354.0 | kaz-image-captioning/requirements-unified.txt | |
| requirements.txt | 2031.0 | kaz-image-captioning/requirements.txt | |
| run.sh | 757.0 | kaz-image-captioning/run.sh | |
| run_esp.sh | 172.0 | kaz-image-captioning/run_esp.sh | |
| run_webcam.sh | 67.0 | kaz-image-captioning/run_webcam.sh | |
| phone_camera_to_server.py | 11923.0 | kaz-image-captioning/src/phone_camera_to_server.py | |
| unified_camera_detection.py | 24057.0 | kaz-image-captioning/src/unified_camera_detection.py | |
| package.json | NaN | package.json | |
| README.md | 6170.0 | pcd-visual-app/README.md | |
| index.tsx | 29657.0 | pcd-visual-app/app/(tabs)/index.tsx | |
| settings.tsx | 36280.0 | pcd-visual-app/app/(tabs)/settings.tsx | |
| history-item-card.tsx | 5747.0 | pcd-visual-app/components/history-item-card.tsx | |
| icon-symbol.tsx | 1458.0 | pcd-visual-app/components/ui/icon-symbol.tsx | |
| AppContext.tsx | 18859.0 | pcd-visual-app/contexts/AppContext.tsx | |
| package.json | 2522.0 | pcd-visual-app/package.json | |
| tts-service.ts | 1782.0 | pcd-visual-app/services/tts-service.ts | |
| styles.ts | 5402.0 | pcd-visual-app/styles/styles.ts | |
| script_inicial.sh | 1367.0 | script_inicial.sh | |

APROVAÇÃO CCB.xlsx - Aprovação CCB

| APROVAÇÃO DO CHANGE CONTROL BOARD (CCB) | | | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 |
|---|---------------------|---------------|------------|------------|------------|
| Projeto: | NaN | NaN | NaN | NaN | NaN |
| Data da Aprovação: | 2025-11-18 18:14:08 | | NaN | NaN | NaN |
| MEMBROS DO CCB (Contribuidores) | NaN | NaN | NaN | NaN | NaN |
| | NaN | NaN | NaN | NaN | NaN |
| Nome | # Commits | Papel | Status | Aprovação | |
| Fábio Henrique | 23 | Líder Técnico | Aprovado | | |
| Fábio Henrique Mendes Brasileiro | 22 | Desenvolvedor | Aprovado | | |
| fabiobrasileiroo | 12 | Desenvolvedor | Aprovado | | |
| Mateus Miranda | 9 | Contribuidor | Aprovado | | |
| heloisacativo | 9 | Contribuidor | Aprovado | | |
| Mateus Mike | 3 | Contribuidor | Aprovado | | |
| LevyAbreu | 2 | Contribuidor | Aprovado | | |
| Gabriel | 1 | Contribuidor | Aprovado | | |

APROVAÇÃO CCB.xlsx - APROVAÇÃO CCB1

Empty DataFrame
Columns: []
Index: []

APROVAÇÃO CCB.xlsx - APROVAÇÃO CCB2

Empty DataFrame
Columns: []
Index: []

ENCERRAMENTO.xlsx - Encerramento

| | | |
|---------------------|--|-----------------------|
| Unnamed: 0 | ENCERRAMENTO DE MUDANÇAS | Unnamed: 2 |
| NaN | NaN | NaN |
| NaN | Projeto: consciencia-espacial-PCD-visual | |
| NaN | Data de Encerramento: | 2025-11-18 18:14:08 |
| NaN | Versão Final App: | NaN |
| NaN | Versão Final Backend: | 1.0.0 |
| NaN | NaN | NaN |
| NaN | RESUMO EXECUTIVO | NaN |
| NaN | Total de Commits: | 30 |
| NaN | Total de Contribuidores: | 8 |
| NaN | Último Commit: | c2955ea em 2025-11-18 |
| NaN | Branch: | main |
| NaN | NaN | NaN |
| PRINCIPAIS ENTREGAS | DESCRÍÇÃO | NaN |
| IC1 | Sistema de detecção de objetos com ESP32-CAM | NaN |
| IC2 | Backend WebSocket/SSE para comunicação em tempo real | NaN |
| IC3 | App mobile React Native com acessibilidade | NaN |
| IC4 | Firmware ESP32 com ESP-NOW para comunicação entre módulos | NaN |
| IC5 | Sistema de captioning de imagens com IA (kaz-image-captioning) | NaN |
| IC6 | Documentação técnica completa | NaN |
| IC7 | Total de 30 commits implementados | NaN |

ENCERRAMENTO.xlsx - ENCERRAMENTO1

Key

Latest Commit

Latest Commit Date

Latest Commit Author

Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\n

ENCERRAMENTO.xlsx - ENCERRAMENTO2

Key

Latest Commit

Latest Commit Date

Latest Commit Author

Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\n

PLANEJAMENTO.xlsx - Planejamento

PLANEJAMENTO DE

Último

A

Merge remote-tracking branch 'refs/remotes/orig
feat: Add status accounting report PDF
Add final model image t
feat: Implement webcam capture functionality with realtime/manu
Merge branch 'main' of github.com:fabiobrasileiroo/sistema_de_pagamento_escolar
feat: Atualiza o módulo de motor de vibração com melhorias na configuração WiFi e mensagens de status, além de adicionar arquivos .gitignore
Merge remote-tracking branch 'refs/remotes/orig
feat: Atualiza envio de detecções para servidor via HTTP POST e ajusta script de
Add ESP32-CAM firmware with captive portal and camera confi
Merge remote-tracking branch 'refs/remotes/orig

PLANEJAMENTO.xlsx - PLANEJAMENTO1

Key
Repository
Branch
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\nTags
Generated On

PLANEJAMENTO.xlsx - PLANEJAMENTO2

Key
Repository
Branch
Latest Commit
Latest Commit Date
Latest Commit Author
Latest Commit Message feat: Adicionar planilha Excel de validação e criada configuração de espaço de trabalho para o projeto ESP32\n\nTags
Generated On

REQUISITOS.xlsx - requisitos

| ID | Tipo | Descrição do Requisito (Resumo) | Caso de Uso Relacionado | Módulo |
|------------|---------------|---|------------------------------------|-----------------------|
| RF01 | Funcional | Medição de Distância (uso do HC-SR04 em 200 ms). | UC04 - Medir Distância | Módulo |
| RF02 | Funcional | Alerta Tátil Proporcional (vibração com intensidade proporcional à distância). | UC02 - Emitir Alerta Tátil | Módulo |
| RF03 | Funcional | Captura de Imagem (a cada 100 ms se a distância for crítica). | UC07 - Capturar Imagem | Módulo |
| RF04 | Funcional | Reconhecimento de Objetos (disponibilizar inferência local ou via App). | UC08 - Reconhecer Objeto | Módulo |
| RF05 | Funcional | Comunicação com App (transferência de dados, timestamp, confidence via BLE). | UC09 - Transmitir Dados | Módulo |
| RF06 | Funcional | TTS no App (mensagens de áudio correspondentes ao evento). | UC03 - Emitir Alerta Sonoro | Módulo |
| RF07 | Funcional | Armazenamento Local (eventos/logs por 30 dias para análise offline). | UC10 - Armazenar Log | Módulo |
| RF08 | Funcional | Controles por Hardware (botão liga/desliga, silêncio temporário, etc.). | UC11 - Controlar Dispositivo | Controle |
| RF09 | Funcional | Falha de Processamento (notificar se a inferência local falhar). | UC12 - Notificar Falha | Módulo |
| RF10 | Funcional | Atualização de Parâmetros via App (distância, sensibilidade, volume TTS). | UC05 - Calibrar Sensores Interface | Módulo |
| RF11 | Funcional | Verificar dispositivos pelo App (status de conexão/bateria). | UC13 - Monitorar Status | Módulo de Controle |
| RF12 | Funcional | Alerta de Bateria (ativação de alerta sonoro/TTS quando atingir nível crítico). | UC03 - Emitir Alerta Sonoro | Interface |
| RF13 | Funcional | Confirmação de Inicialização (vibração + TTS "Sistema Ativo"). | UC01 - Detectar Obstáculo | Módulo |
| RF14 | Funcional | Modo de Silêncio Temporário (silenciar temporariamente todos os alertas). | UC11 - Controlar Dispositivo | Controlador |
| RF15 | Funcional | Exportação de Logs de Teste (CSV/JSON para análise de desempenho offline). | UC10 - Armazenar Log | Módulo |
| RNF01 | Não Funcional | Latência ponta-a-ponta (vibração < 400 ms). | — | Módulo |
| RNF02 | Não Funcional | Acurácia mínima de 70% em ambiente controlado. | — | Módulo |
| RNF03 | Não Funcional | Autonomia mínima de 4 horas de uso típico. | — | Habilidades |
| RNF04 | Não Funcional | Interface de calibração acessível no App. | — | Interface de Controle |
| RNF Outros | Não Funcional | Disponibilidade: 99.9% durante a sessão de uso. | — | — |
| RNF Outros | Não Funcional | Dados de usuários e logs armazenados localmente. | — | — |
| RF01 | Funcional | Medição de Distância (HC-SR04 em 200 ms) | UC04 - Medir Distância | Script |
| RF02 | Funcional | Alerta Tátil Proporcional | UC02 - Emitir Alerta Tátil | Firmware |
| RF03 | Funcional | Captura de Imagem (a cada 100 ms) | UC07 - Capturar Imagem | App |
| RF04 | Funcional | Reconhecimento de Objetos | UC08 - Reconhecer Objeto | App |
| RF05 | Funcional | Comunicação com App (BLE) | UC09 - Transmitir Dados | App |
| RF06 | Funcional | TTS no App | UC03 - Emitir Alerta Sonoro | App |
| RF09 | Funcional | Falha de Processamento | UC12 - Notificar Falha | App |
| RF10 | Funcional | Atualização de Parâmetros via App | UC05 - Calibrar Sensores | App |
| RF11 | Funcional | Verificar dispositivos pelo App | UC13 - Monitorar Status | App |
| RF12 | Funcional | Alerta de Bateria | UC03 - Emitir Alerta Sonoro | Firmware |
| RF13 | Funcional | Confirmação de Inicialização | UC01 - Detectar Obstáculo | App |
| RF14 | Funcional | Modo de Silêncio Temporário | UC11 - Controlar Dispositivo | App |
| RF15 | Funcional | Exportação de Logs de Teste | UC10 - Armazenar Log | App |
| RNF01 | Não Funcional | Latência ponta-a-ponta (< 400 ms) | — | App |
| RNF02 | Não Funcional | Acurácia mínima (70%) | — | App |
| RNF03 | Funcional | [NOVO] Controle de Modo: Conexão WS | UC-09 - Transmitir Dados | App |
| RNF04 | Funcional | [NOVO] Controle de Modo: Alternar para MANUAL | UC-11 - Controlar Dispositivo | App Móvel |
| RNF Outros | Funcional | [NOVO] Controle de Modo: Alternar para REALTIME | UC-11 - Controlar Dispositivo | App Móvel |
| RNF Outros | Funcional | [NOVO] Controle de Modo: Captura Manual | UC-07 - Capturar Imagem | App |

REQUISITOS.xlsx - REQUISITOS1

| Package Name | Version | Path |
|------------------------------|---------|---|
| pcd-vision-backend | 1.0.0 | back-end/package.json |
| change-management-automation | 0.1.0 | docs/change-management copy/automation/package.json |
| nest-vision-api | 1.0.0 | nest-vision-api/package.json |
| pdc-visual-app | 1.0.0 | pcd-visual-app/package.json |

REQUISITOS.xlsx - REQUISITOS2

| Package Name | Version | Path |
|------------------------------|---------|---|
| pcd-vision-backend | 1.0.0 | back-end/package.json |
| change-management-automation | 0.1.0 | docs/change-management copy/automation/package.json |
| nest-vision-api | 1.0.0 | nest-vision-api/package.json |
| pdc-visual-app | 1.0.0 | pcd-visual-app/package.json |

RFC REQUESTS.xlsx - RFC REQUESTS

| Hash | Date | Author |
|---|---------------------|--|
| 57acdb99b59de48bd6f4448d2b97333641f41eb6 | 2025-11-30T18:49:51 | fabiobrasileiroo |
| 08a17315f2b4fe4c7908dada0b069fe732ea0be4 | 2025-11-30T14:41:56 | Mateus Mike |
| d1ab91f3956c3d143ac5c4326ebfe88c9967d659 | 2025-11-28T17:20:25 | fabiobrasileiroo |
| 8c223fef77a4f346b6b7179207a55b7ac5f06951 | 2025-11-27T16:23:20 | fabiobrasileiroo |
| 25608798d67417a86c30d07c4e92b93728a5ad4d | 2025-11-27T16:17:59 | fabiobrasileiroo |
| ef5b32a276575f9ef4f7895f3dde1a93eb27aae9 | 2025-11-27T16:17:40 | fabiobrasileiroo |
| b319f21f68fcf11cc45f1e8d4b2d21c49ccb173d | 2025-11-27T13:02:06 | fabiobrasileiroo |
| 67e28dd49ea4937751bf8be268abee3aeeec33f95 | 2025-11-27T02:55:58 | fabiobrasileiroo |
| fe70cbf83fee2f643d89b26c1ca0e937c29ec191 | 2025-11-26T16:38:37 | fabiobrasileiroo |
| 1544531a68a4bb191c7c742b34022b5cf816c3f2 | 2025-11-26T11:42:57 | fabiobrasileiroo |
| 4bce5ad27dafe76b26642dcc08ac6e993598f03f | 2025-11-24T18:26:48 | fabiobrasileiroo |
| 755ec7e061b01cd30ee0864876932ad9064e7b86 | 2025-11-23T18:51:16 | Fábio Henrique Mendes Brasileiro |
| bb5d1947a9c18ab2b348a0928c346233eфе2ад4а | 2025-11-23T18:30:13 | Fábio Henrique Mendes Brasileiro |
| 6a991bd14a74f4b24656425f7f68a679a1302b3c | 2025-11-23T05:37:51 | Fábio Henrique Mendes Brasileiro |
| 1441c5acf11600ba936d2c17ff7fa6e760d667e | 2025-11-23T05:07:31 | Fábio Henrique Mendes Brasileiro |
| 60cc9cd9af88a2dd07d38d17ef026627aba9026c | 2025-11-23T05:07:18 | Fábio Henrique Mendes Brasileiro |
| 7e44687ad490b8d484fdb94cb260b3d90babad6 | 2025-11-23T03:49:58 | Fábio Henrique Mendes Brasileiro |
| 55e3e61c8f1487605792271c4b2633e38a1dc4696 | 2025-11-21T17:49:59 | LevyAbreu |
| 412cc136937209228e02e2c8527d55cc13b5151b | 2025-11-19T15:34:17 | Mateus Mike |
| a1dd6d6989505ad8439ae2c729b7a8f8e25c04f3 | 2025-11-19T14:54:04 | Mateus Mike |
| 02b567969f0cf8b975f727ad21a34a7a073ebbbf | 2025-11-19T14:29:38 | Mateus Mike |
| 3f11ff0c23caf4d46c78f605f8c653e0fdb949de | 2025-11-19T14:16:09 | Mateus Mike |
| 039ec2e6c21e90eba24ae09de27e44535674f416 | 2025-11-19T14:11:41 | Mateus Mike |
| 1735661b112f57d8140843204c9197e9d097b7c5 | 2025-11-19T13:57:33 | Fábio Henrique Mendes Brasileiro |
| 212f16e70910661aaф1cb5e2acec45c773d57928 | 2025-11-19T13:55:48 | fabiobrasileiroo |
| 161407acb267754f144c75a957a599b722e53e08 | 2025-11-19T13:55:16 | fabiobrasileiroo |
| 50c5c0aa969c31632ad69f199b6c1d8f5a920930 | 2025-11-18T23:47:27 | LevyAbreu |
| 1c59bcea0d0bdbbf5d27db79a1a11e9b9bc5034 | 2025-11-18T23:40:45 | LevyAbreu |
| 825fe0efcc6e6af6cbf570c2f3c574172c64692e | 2025-11-18T22:22:42 | fabiobrasileiroo |
| c2955ead6c50dd7cfeec584947bd07b625c4970e | 2025-11-18T22:12:25 | fabiobrasileiroo |
| 0ad4c6b32c6f655b833bbca428bb3245cf346a6a | 2025-11-18T22:12:08 | fabiobrasileiroo |
| 1d182f9b2b4e5eb6d7afac455f1e4040f60dce51 | 2025-11-18T20:12:26 | Fábio Henrique Mendes Brasileiro |
| f0637ddeb6c466e03f06ba0823e75f9c1d29a11e | 2025-11-17T17:53:17 | fabiobrasileiroo |
| d029742ccb431e0b9f947bab82de0023cefb03a | 2025-11-17T01:30:10 | Fábio Henrique |
| 3ab29c0f1513ce250ae702ddf3160b5fc72beb7b | 2025-11-17T01:30:04 | Fábio Henrique |
| 071250df63a2c15c2582db34e92a79e203586881 | 2025-11-17T01:13:47 | Fábio Henrique Mendes Brasileiro |
| d0577221d1cd5d0085b9683fd69af224c607d37a | 2025-11-17T01:13:16 | Fábio Henrique Mendes Brasileiro |
| 6324c83715342e75c579afa38edfc5cc11d746c0 | 2025-11-17T00:01:09 | Fábio Henrique |
| 3ef875b243d257fc5555da27503992d6294e7a49 | 2025-11-16T23:53:17 | Fábio Henrique Mendes Brasileiro |
| d6c30cddd4dc24c62e76e15dfad3d3bb0c06d9bd | 2025-11-16T23:52:58 | Fábio Henrique Mendes Brasileiro |
| dd5ffc45e0437ffe7d7e19aec58f6cd7cf8fb5c | 2025-11-16T23:52:27 | Fábio Henrique Mendes Brasileiro |
| 2fdafaafc24540d2c5ff2cd6f2decdd60ba9b0cf | 2025-11-16T22:26:03 | Fábio Henrique Mendes Brasileiro |
| a4f1ce26a065473fee25334392ca75cf9a8f8f50 | 2025-11-16T17:52:54 | Fábio Henrique Mendes Brasileiro |
| 16f408809564082375dc840ce41e0699e7e2f56b | 2025-11-16T17:52:40 | Fábio Henrique Mendes Brasileiro |
| 2e801b89b368cc66e123a35534bb5d0639c34eba | 2025-11-16T16:12:11 | Fábio Henrique Mendes Brasileiro |
| 606b2089d3f683fa57b04c8d64958639df6ad838 | 2025-11-16T16:11:23 | Fábio Henrique Mendes Brasileiro |
| 22c0219feb4d6d85d348b264ea40a3d52abc0958 | 2025-11-16T15:56:06 | Fábio Henrique Mendes Brasileiro |
| 7198f126a0f633b61db0071dde3ceee53b12e97e | 2025-11-16T15:20:35 | Fábio Henrique feat: Add SSE and WebSocket test script |
| 646844be2d4ae02f9d6ce505166e87787cd8341b | 2025-11-16T01:10:29 | heloisacativo |
| a3559934a04d9b99862d7377a509ef2d75a24cc8 | 2025-11-15T18:40:59 | Fábio Henrique |
| 9d7d2aa20573ca9f48492fb5c106dbfb1c1dbdb3c | 2025-11-15T18:39:05 | Fábio Henrique |
| a2e0a3bac851cb5ad11e84c6561c5a959c219814 | 2025-11-14T21:28:12 | fabiobrasileiroo |
| 5a15e29e2f5e3de64b11dc50cac7862e5797ccb0 | 2025-11-13T15:15:34 | fabiobrasileiroo |
| 4dc58e828c2b81e6e3823a9072174d7ac5b847e | 2025-11-13T15:07:39 | fabiobrasileiroo |
| dbeed151791bc94cd9cac512c8bd4cd24b9d67ea | 2025-11-12T17:46:36 | Mateus Mike |
| c1875d3879a4a547050a1caa09856214815a5fa0 | 2025-11-12T17:43:47 | Mateus Mike |
| b2159067b8a7b8282ff0880ae47d73d0a0fa520d | 2025-11-12T17:14:07 | Mateus Mike |
| c7cb6b24fe99493c42905a30fc33e454f2bc3e7 | 2025-11-11T15:50:24 | Mateus Miranda |
| b4b37facdd2e5ba6ab136657a195346ebaebb674 | 2025-11-11T15:50:11 | Mateus Miranda |
| b84513e2276ac247fb5785747941c94ce8ff95c9 | 2025-11-11T15:49:25 | Mateus Miranda |
| 4fe048b5a287b7d68ecc42e0a402f336fad0fa61 | 2025-11-11T15:47:57 | Mateus Miranda |
| 4de820ecb2ea664389cf78b20b0d96cb65270a0 | 2025-11-11T15:32:23 | Mateus Miranda |
| 2bf3c754c376da0cd4f895983bc11522aed3656f | 2025-11-11T15:22:46 | Mateus Miranda |
| fe86091d910afac61f81b4400a221c982d549f6c | 2025-11-04T20:58:11 | Mateus Miranda |
| fc9b57b09765ce0b472e0c5a51fcc3e8ce957f1e | 2025-11-04T20:57:42 | Mateus Miranda |
| b7ceaf3e3bcab4a1a2b9f846695ee73514c9b4a8 | 2025-11-04T20:56:58 | Mateus Miranda |
| eedeb68dde8f915c300184830bb3d152552fb1ce0 | 2025-11-03T17:30:31 | Fábio Henrique Mendes Brasileiro |

9ceef2c0de571a7d509394129d25ae759003b671 2025-11-03T17:30:00 Fábio Henrique Mendes Brasileiro
d915adcdfad71e39d07b33670b12c92970731f16 2025-11-03T17:29:20 Fábio Henrique Mendes Brasileiro
7209e8acaeea20c42614e5dc0929dcad94d1c56b 2025-11-03T11:48:06 Fábio Henrique Mendes Brasileiro
452ec6023e750968cad4423f77e88cb48e772b9c 2025-11-01T23:23:46 Fábio Henrique
1e44bb9cdac5df45bb1dc1af3e98c17e7eb6e34e 2025-11-01T23:20:18 Fábio Henrique
e206c7fe7a719dd0102f64193696586759fe35a3 2025-11-01T23:00:34 Fábio Henrique
7ef27155e9f023be7f763fda88379043b0c26408 2025-11-01T22:49:17 Fábio Henrique
605027acba54521b7b45b3fa7ee1495aa96106c6 2025-11-01T17:57:41 Fábio Henrique
7e5bb7f2d2e2e2a3aa44712bd2e772c1777889d1 2025-11-01T16:57:07 Fábio Henrique
ade8c69432a9538a96b7f01221e8bd86f21f75a0 2025-11-01T16:26:23 Fábio Henrique Mendes Brasileiro
4eda64d20b6a8f0df9f41e3037cc55c12f3d1e4e 2025-11-01T16:24:34 Fábio Henrique
5161a725d15eaff46346fdcff8bba89617be2238 2025-11-01T16:15:46 Fábio Henrique
f6b60ef9088d7b86aa4a5e7dba2e917b4ef38940 2025-10-31T20:03:34 fabiobrasileiroo
936b497523036181ca3882a087f5f50d5d3f65f0 2025-10-30T01:20:43 Fábio Henrique
58048dfdd47bfa7916b1a51e0943f7b35d7dab3f 2025-10-30T00:42:24 Fábio Henrique
f4ff1ccb4e27ebd382d17cf237fb80b3aba71f4 2025-10-29T23:40:54 Fábio Henrique
6e6b8d2845b41f4d47cb59a3558da36858ce1efb 2025-10-28T23:32:38 LevyAbreu
2a17d03107ac47c7ad0ae2e480d43c9b6c0eec5 2025-10-28T23:19:00 LevyAbreu
6015feb682728616b8dd99c067f645b4ce391304 2025-10-28T17:44:09 Fábio Henrique Mendes Brasileiro
4aa4aff2c938d99ffdad58774bf8fd691cddc10d 2025-10-28T17:06:51 fabiobrasileiroo
428e4c7a35e76d92afd28bac09b9177b89de9741 2025-10-28T16:57:39 fabiobrasileiroo
60e720dfc3294548f960f3c4e0c4dbc2037733b1 2025-10-28T16:26:50 fabiobrasileiroo
00f5113208f6da039d574682c783c214aefd73cf 2025-10-26T21:38:41 Fábio Henrique
ccbf669c51e175550149359797c5563afcc3007c 2025-10-26T21:36:12 Fábio Henrique
2e68b5ee7cad2281e06f9231d963e8cd29eba39 2025-10-23T17:35:46 Fábio Henrique Mendes Brasileiro
ad8db634b4fa033dcad569203a31f3c09fee4c41 2025-10-21T21:25:32 fabiobrasileiroo
2885dee100a8bacf2532bc609939a2f79d09d3a4 2025-10-21T15:32:20 fabiobrasileiroo
f34276c0183e55955b4cb7385eb44577ac5cc14c 2025-10-13T00:26:45 Fábio Henrique
2e4b8b9fabf20553abff752fc247f11be913fb43 2025-10-13T00:26:34 Fábio Henrique
fd18058311ed027af40d29c2793199090bb42888 2025-10-13T00:05:00 Fábio Henrique Mendes Brasileiro
1fb8f2198f212e132aa3c1d2dabafcfa9fcc4829 2025-10-12T23:58:55 Fábio Henrique
ed8cab6f8e8c9e86aef3169875d8acc5ed077522 2025-09-21T00:08:46 Fábio Henrique
37abf1dfabb48d902e001a2af7d67c60637e9ed5 2025-09-06T18:34:31 Fábio Henrique Mendes Brasileiro

TRIAGEM.xlsx - Triagem

TRIAGEM DE MUDANÇAS

NaN

Projeto:

NaN

COMMITS RECENTES PARA TRIAGEM

NaN

SHA

c2955ea

0ad4c6b

1d182f9

f0637dd

d029742

3ab29c0

071250d

d057722

6324c83

3ef875b

d6c30cd

dd5ffc4

2fdafaa

a4f1ce2

16f4088 Refactor Settings Screen: Replace emoji icons with Lucide icons, enhance loading state with SkeletonLoader, and up

TRIAGEM.xlsx - TRIAGEM1

Changed Files (30 days)

"docs/VALIDA\303\207\303\203O.xlsx"

"docs/change-management/VALIDA\303\207\303\203O.xlsx"

.vscode/settings.json

docs/DR-conciencia-espacial-pdc-visual.pdf

docs/DR.pdf

docs/ERS-conciencia-espacial-pdc-visual.pdf

docs/ERS.OculosEspecial.pdf

docs/ERS.pdf

docs/Manual_Equipe_OculosEspecial.pdf

docs/OculosEspecial_Relatorio-Consolidado_Final.pdf

docs/Organograma_OculosEspecial.PNG

docs/Organograma_OculosEspecial.pdf

docs/PLANEJAMENTO.xlsx

docs/PP-conciencia-espacial-pdc-visual.pdf

docs/PP.pdf

docs/Passo_a_Passo_OculosEspecial.pdf

docs/Relatorio-Consolidado_Final.pdf

docs/Relatorio_Consolidado_Final.docx

docs/TRIAGEM.xlsx

docs/change-management/REQUISITOS.xlsx

docs/change-management/TRIAGEM.xlsx

docs/requisitos.csv

docs/requisitos.xlsx

firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp

firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp

firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md

firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace

firmware/esp32-pai-mestre/src/main.cpp

firmware/modulo1-sensor/src/main.cpp

kaz-image-captioning/EXECUTAR.md

kaz-image-captioning/INSTALL.md

kaz-image-captioning/README.md

kaz-image-captioning/dep.sh

kaz-image-captioning/install.ps1

kaz-image-captioning/install.sh

kaz-image-captioning/launcher.py

kaz-image-captioning/requirements-unified.txt

kaz-image-captioning/requirements.txt

kaz-image-captioning/run.sh

kaz-image-captioning/run_esp.sh

kaz-image-captioning/run_webcam.sh

kaz-image-captioning/src/phone_camera_to_server.py

```
kaz-image-captioning/src/unified_camera_detection.py
    package.json
        pcd-visual-app/README.md
        pcd-visual-app/app/(tabs)/index.tsx
        pcd-visual-app/app/(tabs)/settings.tsx
    pcd-visual-app/components/history-item-card.tsx
        pcd-visual-app/components/ui/icon-symbol.tsx
        pcd-visual-app/contexts/AppContext.tsx
            pcd-visual-app/package.json
        pcd-visual-app/services/tts-service.ts
            pcd-visual-app/styles/styles.ts
                script_inicial.sh
```

TRIAGEM.xlsx - TRIAGEM2

```
    Changed Files (30 days)
        "docs/VALIDA\303\207\303\203O.xlsx"
    "docs/change-management/VALIDA\303\207\303\203O.xlsx"
        .vscode/settings.json
    docs/DR-conciencia-espacial-pdc-visual.pdf
        docs/DR.pdf
    docs/ERS-conciencia-espacial-pdc-visual.pdf
        docs/ERS.OculosEspecial.pdf
        docs/ERS.pdf
    docs/Manual_Equipe_OculosEspecial.pdf
    docs/OculosEspecial_Relatorio-Consolidado_Final.pdf
        docs/Organograma_OculosEspecial.PNG
        docs/Organograma_OculosEspecial.pdf
        docs/PLANEJAMENTO.xlsx
    docs/PP-conciencia-espacial-pdc-visual.pdf
        docs/PP.pdf
    docs/Passo_a_Passo_OculosEspecial.pdf
    docs/Relatorio-Consolidado_Final.pdf
    docs/Relatorio_Consolidado_Final.docx
        docs/TRIAGEM.xlsx
    docs/change-management/REQUISITOS.xlsx
    docs/change-management/TRIAGEM.xlsx
        docs/requisitos.csv
        docs/requisitos.xlsx
firmware/esp32-pai-mestre/INTEGRACAO_SERVIDOR_SSE.cpp
firmware/esp32-pai-mestre/INTEGRACAO_WEBSOCKET.cpp
firmware/esp32-pai-mestre/WEBSOCKET_VS_HTTP.md
firmware/esp32-pai-mestre/src/inovatech-2025.code-workspace
    firmware/esp32-pai-mestre/src/main.cpp
    firmware/modulo1-sensor/src/main.cpp
        kaz-image-captioning/EXECUTAR.md
        kaz-image-captioning/INSTALL.md
        kaz-image-captioning/README.md
        kaz-image-captioning/dep.sh
        kaz-image-captioning/install.ps1
        kaz-image-captioning/install.sh
        kaz-image-captioning/launcher.py
    kaz-image-captioning/requirements-unified.txt
    kaz-image-captioning/requirements.txt
        kaz-image-captioning/run.sh
        kaz-image-captioning/run_esp.sh
        kaz-image-captioning/run_webcam.sh
kaz-image-captioning/src/phone_camera_to_server.py
kaz-image-captioning/src/unified_camera_detection.py
    package.json
        pcd-visual-app/README.md
        pcd-visual-app/app/(tabs)/index.tsx
        pcd-visual-app/app/(tabs)/settings.tsx
    pcd-visual-app/components/history-item-card.tsx
        pcd-visual-app/components/ui/icon-symbol.tsx
        pcd-visual-app/contexts/AppContext.tsx
            pcd-visual-app/package.json
        pcd-visual-app/services/tts-service.ts
            pcd-visual-app/styles/styles.ts
                script_inicial.sh
```

VALIDAÇÃO.xlsx - Validação

| VALIDAÇÃO DE MUDANÇAS | | Unnamed: 1 | | Unnamed: 2 | | | Unnamed: 3 | Unnamed: 4 |
|--|--------------------------------|-------------------------------|-------------------------|------------------|------------------------------|-----|-------------|------------|
| NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| Projeto: consciencia-espacial-PCD-visual | | | | | | | | |
| Ambiente: | Produção | | NaN | NaN | NaN | | NaN | |
| NaN | NaN | NaN | NaN | NaN | NaN | | NaN | |
| TESTES DE VALIDAÇÃO | | NaN | | NaN | NaN | NaN | | NaN |
| NaN | NaN | | NaN | NaN | NaN | | NaN | |
| Módulo | Teste | Critério | Resultado | | | | Observações | |
| back-end | Testes de integração WebSocket | Conexões | ESP32 e App funcionando | Passou | API funcionando corretamente | | | |
| back-end | Teste de SSE | Eventos enviados corretamente | Passou | Streaming | de eventos OK | | | |
| pdc-visual-app | Testes E2E mobile | Todas as features | funcionando | Passou | App responsivo e funcional | | | |
| firmware | Teste de comunicação ESP-NOW | Módulos se comunicando | Passou | | Latência < 100ms | | | |
| firmware | Teste sensor HC-SR04 | Leitura de distância precisa | Passou | Precisão de ±2cm | | | | |
| kaz-image-captioning | Teste de inferência | Captions geradas corretamente | Passou | Modelo v2 | funcionando | | | |

VALIDAÇÃO.xlsx - VALIDAÇÃO1

| | |
|------------------------------|-----|
| Package HasTestScript | |
| pcd-vision-backend | yes |
| change-management-automation | no |
| nest-vision-api | yes |
| pdc-visual-app | yes |

VALIDAÇÃO.xlsx - VALIDAÇÃO2

| | |
|------------------------------|-----|
| Package HasTestScript | |
| pcd-vision-backend | yes |
| change-management-automation | no |
| nest-vision-api | yes |
| pdc-visual-app | yes |

README.md

Change Management Automation

Este script gera um arquivo Excel (CHANGE_MANAGEMENT_AUTOGENERATED.xlsx) na pasta docs/change-management com informações extraídas do repositório e do Git.

Planilhas criadas:

- PLANEJAMENTO: Visão geral do repositório e branch
- REQUISITOS: Lista de pacotes (package.json encontrados)
- RFC REQUESTS: Lista dos últimos commits
- TRIAGEM: Arquivos modificados nos últimos 30 dias
- APROVAÇÃO CCB: Lista de contribuidores e e-mails
- ANÁLISE IMPACTO: Arquivos alterados e seus tamanhos
- VALIDAÇÃO: Indicador de existência do script test nos pacotes
- ENCERRAMENTO: Informações de fechamento (último commit)

Como usar:

- 1) Instalar dependências:

```
bash  
cd docs/change-management/automation  
npm install
```

- 2) Executar:

```
```bash  
npm run generate
opcional: sobrescrever os templates existentes com --overwrite
node generate_change_management.js --overwrite
```
```

Notas:

- O script deve ser executado a partir do repositório com git disponível e histórico de commits.
- A flag --overwrite irá sobrescrever os arquivos de planilhas existentes em docs/change-management com os dados gerados.

generate_change_management.js

```
#!/usr/bin/env node
const fs = require('fs');
const path = require('path');
const ExcelJS = require('exceljs');
const simpleGit = require('simple-git');

const repoRoot = path.resolve(__dirname, '..', '..'); // /docs/change-management/.. => repo root
const changeManagementFolder = path.resolve(__dirname, '..');
const git = simpleGit(repoRoot);

// Config
const outFile = path.join(changeManagementFolder, 'CHANGE_MANAGEMENT_AUTOGENERATED.xlsx');
const overwriteTemplates = process.argv.includes('--overwrite');

// Utilities
async function findPackageJsonFiles(root) {
  const results = [];
  const ignoreDirs = new Set(['node_modules', 'venv', '.git', '.venv', '.cache', 'dist', 'build']);
  async function walk(dir) {
    const list = await fs.promises.readdir(dir, { withFileTypes: true });
    for (const dirent of list) {
      const full = path.join(dir, dirent.name);
      if (dirent.isDirectory()) {
        if (!ignoreDirs.has(dirent.name)) {
          await walk(full);
        }
      } else if (dirent.isFile() && dirent.name === 'package.json') {
        // skip top-level 'docs/change-management/automation/package.json'
        if (!full.startsWith(changeManagementFolder)) {
          results.push(full);
        }
      }
    }
  }
  await walk(root);
  return results;
}

async function gatherData() {
  // Gather repo metadata
  const repoName = path.basename(repoRoot);
  const branch = (await git.revparse(['--abbrev-ref', 'HEAD'])).trim();
  const latestCommit = await git.log({ n: 1 });
  const latest = latestCommit.all[0] || {};
  const contributors = (await git.raw(['shortlog', '-sne'])).split('\n').filter(Boolean).map(line => {
    const parts = line.trim().split('t').filter(Boolean);
    const nameEmail = parts[1] || '';
    const m = nameEmail.match(/(.*) <(.*)>/);
    if (m) {
      return { commits: parts[0].trim(), name: m[1].trim(), email: m[2].trim() };
    }
    return { commits: parts[0].trim(), name: nameEmail, email: '' };
  });

  // Get commits (last 200)
  const commits = (await git.log({ n: 200 })).all.map(c => ({ hash: c.hash, date: c.date, message: c.message, author: c.author_name, email: c.author_email }));

  // Get tags
  const tags = (await git.tags()).all;

  // Find all package.json files
  const packageFiles = await findPackageJsonFiles(repoRoot);
  const packages = [];
  for (const p of packageFiles) {
    try {
      const raw = await fs.promises.readFile(p, 'utf8');
      const json = JSON.parse(raw);
      packages.push({ name: json.name || path.basename(path.dirname(p)), version: json.version || "", path: path.relative(repoRoot, p), dependencies: json.dependencies || {}, devDependencies: json.devDependencies || {} });
    } catch (e) {
      console.warn('Failed to parse', p, e.message);
    }
  }
}
```

```

}

// Get files changed in last 30 days
const since = new Date(Date.now() - 30 * 24 * 60 * 60 * 1000);
const recentCommits = (await git.log({ since: since.toISOString() })).all;
const changedFilesSet = new Set();
for (const c of recentCommits) {
  const diff = await git.show(['--name-only', '--pretty=', c.hash]);
  diff.split('\n').filter(Boolean).forEach(f => changedFilesSet.add(f));
}
const changedFiles = Array.from(changedFilesSet);

return { repoName, branch, latest, contributors, commits, tags, packages, changedFiles };
}

async function writeWorkbook(data) {
  const w = new ExcelJS.Workbook();
  w.creator = 'Change Management Automation';
  w.created = new Date();

  // PLANEJAMENTO: Project overview
  const planning = w.addWorksheet('PLANEJAMENTO');
  planning.addRow(['Repository', data.repoName]);
  planning.addRow(['Branch', data.branch]);
  planning.addRow(['Latest Commit', data.latest.hash || '']);
  planning.addRow(['Latest Commit Date', data.latest.date || '']);
  planning.addRow(['Latest Commit Author', data.latest.author || '']);
  planning.addRow(['Tags', data.tags.join(',')]);
  planning.addRow(['Generated On', new Date().toISOString()]);

  // REQUISITOS: We'll list package metadata and basic requirements
  const req = w.addWorksheet('REQUISITOS');
  req.addRow(['Package Name', 'Version', 'Path']);
  data.packages.forEach(p => req.addRow([p.name, p.version, p.path]));

  // RFC REQUESTS: list commits (as potential change requests)
  const rfc = w.addWorksheet('RFC REQUESTS');
  rfc.addRow(['Hash', 'Date', 'Author', 'Message']);
  data.commits.forEach(c => rfc.addRow([c.hash, c.date, c.author, c.message]));

  // TRIAGEM: changed files in last 30 days
  const triage = w.addWorksheet('TRIAGEM');
  triage.addRow(['Changed Files (30 days)']);
  data.changedFiles.forEach(f => triage.addRow([f]));

  // APROVAÇÃO CCB: contributors
  const ccb = w.addWorksheet('APROVAÇÃO CCB');
  ccb.addRow(['Name', 'Email', 'Commits']);
  data.contributors.forEach(c => ccb.addRow([c.name, c.email, c.commits]));

  // ANÁLISE IMPACTO: files with details and sizes
  const impact = w.addWorksheet('ANÁLISE IMPACTO');
  impact.addRow(['File', 'Size (bytes)', 'Path']);
  for (const f of data.changedFiles) {
    try {
      const st = await fs.promises.stat(path.join(repoRoot, f));
      impact.addRow([path.basename(f), st.size, f]);
    } catch (e) {
      impact.addRow([path.basename(f), 'n/a', f]);
    }
  }

  // VALIDAÇÃO: maybe tests and script existence
  const val = w.addWorksheet('VALIDAÇÃO');
  val.addRow(['Package', 'Has TestScript']);
  data.packages.forEach(p => {
    try {
      // read script from package.json
      const full = path.join(repoRoot, p.path);
      const pkg = JSON.parse(fs.readFileSync(full, 'utf8'));
      const hasTest = pkg.scripts && pkg.scripts.test ? 'yes' : 'no';
      val.addRow([p.name, hasTest]);
    } catch (e) {
      val.addRow([p.name, 'unknown']);
    }
  });
}

```

```
});

// ENCERRAMENTO: closure info like last commit message and summary
const close = w.addWorksheet('ENCERRAMENTO');
close.addRow(['Latest Commit', data.latest.hash || '']);
close.addRow(['Latest Commit Date', data.latest.date || '']);
close.addRow(['Latest Commit Author', data.latest.author || '']);
close.addRow(['Latest Commit Message', data.latest.message || '']);

await w.xlsx.writeFile(outFile);
console.log('Written to', outFile);

if (overwriteTemplates) {
  const templates = ['PLANEJAMENTO.xlsx', 'REQUISITOS.xlsx', 'RFC REQUESTS.xlsx', 'TRIAGEM.xlsx',
    'APROVAÇÃO CCB.xlsx', 'ANALISE IMPACTO.xlsx', 'VALIDAÇÃO.xlsx', 'ENCERRAMENTO.xlsx'];
  for (const t of templates) {
    const targetPath = path.join(changeManagementFolder, t);
    try {
      await w.xlsx.writeFile(targetPath);
      console.log('Overwrote template', targetPath);
    } catch (e) {
      console.warn('Failed to overwrite', t, e.message);
    }
  }
}

(async () => {
try {
  console.log('Gathering data...');
  const data = await gatherData();
  console.log('Writing workbook...');
  await writeWorkbook(data);
  console.log('Done.');
} catch (err) {
  console.error('Error:', err);
  process.exit(1);
}
})();
```

package-lock.json

```
{  
  "name": "change-management-automation",  
  "version": "0.1.0",  
  "lockfileVersion": 3,  
  "requires": true,  
  "packages": {  
    "": {  
      "name": "change-management-automation",  
      "version": "0.1.0",  
      "dependencies": {  
        "exceljs": "^4.3.0",  
        "simple-git": "^3.19.0"  
      }  
    },  
    "node_modules/@fast-csv/format": {  
      "version": "4.3.5",  
      "resolved": "https://registry.npmjs.org/@fast-csv/format/-/format-4.3.5.tgz",  
      "integrity": "sha512-8iRn6QF3I8Ak78lNAa+Gdl5MJJBM5vRHivFtMRUWINdevNo00K7OXxS2PshawLKTejVwiellP  
mK5YILu6w4u8A==",  
      "license": "MIT",  
      "dependencies": {  
        "@types/node": "^14.0.1",  
        "lodash.escaperegexp": "^4.1.2",  
        "lodash.isboolean": "^3.0.3",  
        "lodash.isequal": "^4.5.0",  
        "lodash.isfunction": "^3.0.9",  
        "lodash.isnil": "^4.0.0"  
      }  
    },  
    "node_modules/@fast-csv/parse": {  
      "version": "4.3.6",  
      "resolved": "https://registry.npmjs.org/@fast-csv/parse/-/parse-4.3.6.tgz",  
      "integrity": "sha512-uRsLYksqbDmWaSmzvJcuApSEe38+6NQZBUsuAyMZKqHxH0g1wcJgsKUVN3WC8tewaqFjBMMG  
rkHmC+T7k8LvA==",  
      "license": "MIT",  
      "dependencies": {  
        "@types/node": "^14.0.1",  
        "lodash.escaperegexp": "^4.1.2",  
        "lodash.groupby": "^4.6.0",  
        "lodash.isfunction": "^3.0.9",  
        "lodash.isnil": "^4.0.0",  
        "lodash.isundefined": "^3.0.1",  
        "lodash.uniq": "^4.5.0"  
      }  
    },  
    "node_modules/@kwsites/file-exists": {  
      "version": "1.1.1",  
      "resolved": "https://registry.npmjs.org/@kwsites/file-exists/-/file-exists-1.1.1.tgz",  
      "integrity": "sha512-m9/5YGR18llwxSFDwfE3oA7bWuq9kdau6ugN4H2rJeyhFQZcG9AgSHkQtSD15a8WvTgfz9aik  
ZMrKPHvbpqFiw==",  
      "license": "MIT",  
      "dependencies": {  
        "debug": "^4.1.1"  
      }  
    },  
    "node_modules/@kwsites/promise-deferred": {  
      "version": "1.1.1",  
      "resolved": "https://registry.npmjs.org/@kwsites/promise-deferred/-/promise-deferred-1.1.1.tgz",  
      "integrity": "sha512-GaHYm+c0O9MjZRu0ongGBRbinu8gVAMd2UZji6jVmqtZluZnptXGWhz1E8j8D2HJ3f/yMxK  
AUC0b+57wnclw==",  
      "license": "MIT"  
    },  
    "node_modules/@types/node": {  
      "version": "14.18.63",  
      "resolved": "https://registry.npmjs.org/@types/node/-/node-14.18.63.tgz",  
      "integrity": "sha512-fAtCfv4jJg+ExtXhvCkCqUKZ+4ok/JQk01qDKhL5BDDoS3AxKXhV5/MAVUZyQnSEd2GT92fkg  
ZI0pz0Q0AzclQ==",  
      "license": "MIT"  
    },  
    "node_modules/archiver": {  
      "version": "5.3.2",  
      "resolved": "https://registry.npmjs.org/archiver/-/archiver-5.3.2.tgz",  
      "integrity": "sha512-+25nxyyznAXF7Nef3y0EbBeqmGZgeN/BxHX29Rs39djAfaFalmQ89SE6CWyDCHzGL0yt/ycBt
```

NOmGTW0FyGWNw==",
"license": "MIT",
"dependencies": {
"archiver-utils": "^2.1.0",
"async": "^3.2.4",
"buffer-crc32": "^0.2.1",
"readable-stream": "^3.6.0",
"readdir-glob": "^1.1.2",
"tar-stream": "^2.2.0",
"zip-stream": "^4.1.0"
},
"engines": {
"node": ">= 10"
}
,
"node_modules/archiver-utils": {
"version": "2.1.0",
"resolved": "https://registry.npmjs.org/archiver-utils/-/archiver-utils-2.1.0.tgz",
"integrity": "sha512-bEL/yUb/fNNiNTuUz979Z0Yg5L+LzLxGJz8x79iYmR54fmTlb6ob/hNQgkQnIUDW
Xs356l6BAMHfw==",
"license": "MIT",
"dependencies": {
"glob": "^7.1.4",
"graceful-fs": "^4.2.0",
"lazystream": "^1.0.0",
"lodash.defaults": "^4.2.0",
"lodash.difference": "^4.5.0",
"lodash.flatten": "^4.4.0",
"lodash.isplainobject": "^4.0.6",
"lodash.union": "^4.6.0",
"normalize-path": "^3.0.0",
"readable-stream": "^2.0.0"
},
"engines": {
"node": ">= 6"
}
,
"node_modules/archiver-utils/node_modules/readable-stream": {
"version": "2.3.8",
"resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
"integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrF
QrPmum45RTtdA==",
"license": "MIT",
"dependencies": {
"core-util-is": "~1.0.0",
"inherits": "~2.0.3",
"isarray": "~1.0.0",
"process-nextick-args": "~2.0.0",
"safe-buffer": "~5.1.1",
"string_decoder": "~1.1.1",
"util-deprecate": "~1.0.1"
}
,
"node_modules/archiver-utils/node_modules/safe-buffer": {
"version": "5.1.2",
"resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
"integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmM
/wPtojys4G6+g==",
"license": "MIT"
,
"node_modules/archiver-utils/node_modules/string_decoder": {
"version": "1.1.1",
"resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
"integrity": "sha512-n/ShnDi6FHbbVfviro+WojiFzv+s8MPMHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJ
HyPyTP/mzRfwg==",
"license": "MIT",
"dependencies": {
"safe-buffer": "~5.1.0"
}
,
"node_modules/async": {
"version": "3.2.6",
"resolved": "https://registry.npmjs.org/async/-/async-3.2.6.tgz",
"integrity": "sha512-htCUDlxxyCLMgaM3xXg0C0LW2xqfuQ6p05pCEIsXuyQ+a1koYKTuBMzRNwmy
Tn4+E0Ti6C2AA==",
"license": "MIT"

```
  },
  "node_modules/balanced-match": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",
    "integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0wIISEJhYU5Pw==",
    "license": "MIT"
  },
  "node_modules/base64-js": {
    "version": "1.5.1",
    "resolved": "https://registry.npmjs.org/base64-js/-/base64-js-1.5.1.tgz",
    "integrity": "sha512-AKpaYlHn8t4SVbOHCy+b5+KKgvR4vrsD8vbvrbiQJps7fKDTkjkdry6ji0rUJjC0kzbxq8iypo41qeWA==",
    "funding": [
      {
        "type": "github",
        "url": "https://github.com/sponsors/feross"
      },
      {
        "type": "patreon",
        "url": "https://www.patreon.com/feross"
      },
      {
        "type": "consulting",
        "url": "https://feross.org/support"
      }
    ],
    "license": "MIT"
  },
  "node_modules/big-integer": {
    "version": "1.6.52",
    "resolved": "https://registry.npmjs.org/big-integer/-/big-integer-1.6.52.tgz",
    "integrity": "sha512-QxD8cf2eVqJOz63z6JIN9BzvVs/dlySa5HGSBH5xtR8dPteIRQnBxxKqkNTiT6jd4oMcND9RGbQg==",
    "license": "Unlicense",
    "engines": {
      "node": ">=0.6"
    }
  },
  "node_modules/binary": {
    "version": "0.3.0",
    "resolved": "https://registry.npmjs.org/binary/-/binary-0.3.0.tgz",
    "integrity": "sha512-D4H1y5KYwpJgK8wk1Cue5LLPgmwHKYSChkbsPQg5JtVuR5ulGckxfR62H3AEYihuz3Aqq2XZq==",
    "license": "MIT",
    "dependencies": {
      "buffers": "~0.1.1",
      "chainsaw": "~0.1.0"
    },
    "engines": {
      "node": "*"
    }
  },
  "node_modules/bl": {
    "version": "4.1.0",
    "resolved": "https://registry.npmjs.org/bl/-/bl-4.1.0.tgz",
    "integrity": "sha512-1W07cM9gS6DcLperZfFSj+bWLtaPGSOHWhPiGzXmvVJbRLdG82sH/Kn8EtW1k5BbnlbwF3h6w==",
    "license": "MIT",
    "dependencies": {
      "buffer": "^5.5.0",
      "inherits": "^2.0.4",
      "readable-stream": "^3.4.0"
    }
  },
  "node_modules/bluebird": {
    "version": "3.4.7",
    "resolved": "https://registry.npmjs.org/bluebird/-/bluebird-3.4.7.tgz",
    "integrity": "sha512-iD3898SR7sWVRHbiQv+sHUtHnMvC1o3nW5rAcqnq3uOn07DSAppZYUkIGsIDYVBgoEJASPchA==",
    "license": "MIT"
  },
  "node_modules/brace-expansion": {
    "version": "1.1.12",
    "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.12.tgz",
    "integrity": "sha512-9T9UJW3r0UW5c1Q7GTwiptXwhvYmEzFhzMfZ9H7FQWt+uZePjZPjBP/W1ZEy
```

TF4IPcqjnDHcg==",
"license": "MIT",
"dependencies": {
"balanced-match": "^1.0.0",
"concat-map": "0.0.1"
}
,
"node_modules/buffer": {
"version": "5.7.1",
"resolved": "https://registry.npmjs.org/buffer/-/buffer-5.7.1.tgz",
"integrity": "sha512-EHcyIPBQ4BSGlvjB16k5KgAJ27CIsHY2JBmCRReo48y9rQ3MaUzWX3KVIBa4K7C3WaB3ju7FQ==",
"funding": [
{
"type": "github",
"url": "https://github.com/sponsors/feross"
},
{
"type": "patreon",
"url": "https://www.patreon.com/feross"
},
{
"type": "consulting",
"url": "https://feross.org/support"
}
],
"license": "MIT",
"dependencies": {
"base64-js": "^1.3.1",
"ieee754": "^1.1.13"
}
,
"node_modules/buffer-crc32": {
"version": "0.2.13",
"resolved": "https://registry.npmjs.org/buffer-crc32/-/buffer-crc32-0.2.13.tgz",
"integrity": "sha512-VO9Ht/p3SN7SKWqcrgezjGbRSJYTx+Q1pTQC0wrWqHx0vpJraQ6GtHx8tvcaJcxgOgi7B0TDkQ==",
"license": "MIT",
"engines": {
"node": "*"
}
,
"node_modules/buffer-indexof-polyfill": {
"version": "1.0.2",
"resolved": "https://registry.npmjs.org/buffer-indexof-polyfill/-/buffer-indexof-polyfill-1.0.2.tgz",
"integrity": "sha512-I7wzHwA3t1/lwXQh+A5PbNvJxgfo5r3xulgYDB5zckTu/Z9oUK9biouBKQUjEqzaYoovmE+GqSf7A==",
"license": "MIT",
"engines": {
"node": ">=0.10"
}
,
"node_modules/buffers": {
"version": "0.1.1",
"resolved": "https://registry.npmjs.org/buffers/-/buffers-0.1.1.tgz",
"integrity": "sha512-9q/rDEGSb/Qsvv2qvzlzdluL5k7AaJOTrw23z9reQthrbF7is4CtlT0DXyO1oei2DCpQ7igaSHp1kAEQ==",
"engines": {
"node": ">=0.2.0"
}
,
"node_modules/chainsaw": {
"version": "0.1.0",
"resolved": "https://registry.npmjs.org/chainsaw/-/chainsaw-0.1.0.tgz",
"integrity": "sha512-75kWfWt6MEKNC8xYXIdRpDehRYY/tNSgwKaJq+dbbDcxORuVrrQ+SEHoWsniIeDk/4YNp1rNQ==",
"license": "MIT/X11",
"dependencies": {
"traverse": ">=0.3.0 <0.4"
}
,
"engines": {
"node": "*"
}
,
"node_modules/compress-commons": {

```
"version": "4.1.2",
"resolved": "https://registry.npmjs.org/compress-commons/-/compress-commons-4.1.2.tgz",
"integrity": "sha512-D3uMHtGc/fcO1Gt1/L7i1e33VOvD4A9hfQLP+6ewd+BvG/gQ84Yh4oftEhAdjSMgpT7JYNpo6MHHg==",
"license": "MIT",
"dependencies": {
  "buffer-crc32": "^0.2.13",
  "crc32-stream": "^4.0.2",
  "normalize-path": "^3.0.0",
  "readable-stream": "^3.6.0"
},
"engines": {
  "node": ">= 10"
}
},
"node_modules/concat-map": {
  "version": "0.0.1",
  "resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",
  "integrity": "sha512-Srv4dswyQNBFohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaHdodFvbkhKmqg==",
  "license": "MIT"
},
"node_modules/core-util-is": {
  "version": "1.0.3",
  "resolved": "https://registry.npmjs.org/core-util-is/-/core-util-is-1.0.3.tgz",
  "integrity": "sha512-ZQBvi1DcpJ4GDqanjucZ2Hj3wEO5pZDS89BWbkcrvdxksJorwUDDZamX9ldFkpObEA4DWNJ9FYQ==",
  "license": "MIT"
},
"node_modules/crc-32": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/crc-32/-/crc-32-1.2.2.tgz",
  "integrity": "sha512-ROmzCKrTnOwybPcJApAA6WBWij23HVfGVNKqqrZpuyZOHQK2CwHSvpGuyt/Gp4Mh+le1IHJQ==",
  "license": "Apache-2.0",
  "bin": {
    "crc32": "bin/crc32.njs"
  },
  "engines": {
    "node": ">=0.8"
  }
},
"node_modules/crc32-stream": {
  "version": "4.0.3",
  "resolved": "https://registry.npmjs.org/crc32-stream/-/crc32-stream-4.0.3.tgz",
  "integrity": "sha512-NT7w2JVU7DFroFdYkeq8cywxrgjPHWkdX1wjpRQXPX5Asews3tA+Ght6IddQO5Eqku3epj2tolw==",
  "license": "MIT",
  "dependencies": {
    "crc-32": "^1.2.0",
    "readable-stream": "^3.4.0"
  },
  "engines": {
    "node": ">= 10"
  }
},
"node_modules/dayjs": {
  "version": "1.11.19",
  "resolved": "https://registry.npmjs.org/dayjs/-/dayjs-1.11.19.tgz",
  "integrity": "sha512-t5EcLVS6QPBNqM2z8fakk/NKei+Xzshgt8FFKAn+qwID1pzZWxh0nVCrvFK7ZDkC7CBWTRIVApAw==",
  "license": "MIT"
},
"node_modules/debug": {
  "version": "4.4.3",
  "resolved": "https://registry.npmjs.org/debug/-/debug-4.4.3.tgz",
  "integrity": "sha512-RGwwWnwQvkVfavKVf22FGLw+xYSdzARwm0ru6DhTVA3umU5hZc28V3kO4sGijltAjNbCqA==",
  "license": "MIT",
  "dependencies": {
    "ms": "^2.1.3"
  },
  "engines": {
    "node": ">=6.0"
  }
},
"peerDependenciesMeta": {
```

```
"supports-color": {
  "optional": true
}
},
"node_modules/duplexer2": {
  "version": "0.1.4",
  "resolved": "https://registry.npmjs.org/duplexer2/-/duplexer2-0.1.4.tgz",
  "integrity": "sha512-asLFVfWWtJ90ZyOUEMqk7/S2w2guQKxUI2itj3d92ADHhxUSbCMGi1f1cBcJ7xM
wo1yuNbMEPKeA==",
  "license": "BSD-3-Clause",
  "dependencies": {
    "readable-stream": "^2.0.2"
  }
},
"node_modules/duplexer2/node_modules/readable-stream": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrF
QrPmum45RTtdA==",
  "license": "MIT",
  "dependencies": {
    "core-util-is": "~1.0.0",
    "inherits": "~2.0.3",
    "isarray": "~1.0.0",
    "process-nextick-args": "~2.0.0",
    "safe-buffer": "~5.1.1",
    "string_decoder": "~1.1.1",
    "util-deprecate": "~1.0.1"
  }
},
"node_modules/duplexer2/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKKUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmM
/wPtojs4G6+g==",
  "license": "MIT"
},
"node_modules/duplexer2/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnDi6FHbbVfviro+WojFzv+s8MPMFBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJ
HyPyTP/mzRfwg==",
  "license": "MIT",
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/end-of-stream": {
  "version": "1.4.5",
  "resolved": "https://registry.npmjs.org/end-of-stream/-/end-of-stream-1.4.5.tgz",
  "integrity": "sha512-ooEGc6HP26xXq/N+GCGOT0JKCLDGrq2bQUZrQ7gyiJiZANJ/8YDTxTpQBXGM
mm7KYVICQnyOg==",
  "license": "MIT",
  "dependencies": {
    "once": "^1.4.0"
  }
},
"node_modules/exceljs": {
  "version": "4.4.0",
  "resolved": "https://registry.npmjs.org/exceljs/-/exceljs-4.4.0.tgz",
  "integrity": "sha512-XctvKaEMaj1li9oDOqbW/6e1gXknSY4g/aLCDicOXqBE4M0nRWkUu0PTp++UP
s/VadKIS6Ilvg==",
  "license": "MIT",
  "dependencies": {
    "archiver": "^5.0.0",
    "dayjs": "^1.8.34",
    "fast-csv": "^4.3.1",
    "jszip": "^3.10.1",
    "readable-stream": "^3.6.0",
    "saxes": "^5.0.1",
    "tmp": "^0.2.0",
    "unzipper": "^0.10.11",
    "uuid": "^8.3.0"
  }
},
"engines": {
```

```
        "node": ">=8.3.0"
    },
    "node_modules/fast-csv": {
        "version": "4.3.6",
        "resolved": "https://registry.npmjs.org/fast-csv/-/fast-csv-4.3.6.tgz",
        "integrity": "sha512-2RNSpuwwsJGP0frGsOmTb9oUF+VkFSM4SyLTDgwf2ciHWTarN0lQTC+F2f/t5AAu85GsvMIusw==",
        "license": "MIT",
        "dependencies": {
            "@fast-csv/format": "4.3.5",
            "@fast-csv/parse": "4.3.6"
        },
        "engines": {
            "node": ">=10.0.0"
        }
    },
    "node_modules/fs-constants": {
        "version": "1.0.0",
        "resolved": "https://registry.npmjs.org/fs-constants/-/fs-constants-1.0.0.tgz",
        "integrity": "sha512-y6OAwoSlf7FyjMlv94u+b5rdheZEjzR63GTyZJm5qh4Bi+2YgwLCcl/fPFZkL5PSix+wHfp/Bciwow==",
        "license": "MIT"
    },
    "node_modules/fs.realpath": {
        "version": "1.0.0",
        "resolved": "https://registry.npmjs.org/fs.realpath/-/fs.realpath-1.0.0.tgz",
        "integrity": "sha512-OO0pH2IK6a0hZnAda5ltzHPI6pUlvi7jMVnxUQRtw4owF2wk8lOsabtGDCTP4G9X8K1t4+fGMDW==",
        "license": "ISC"
    },
    "node_modules/fstream": {
        "version": "1.0.12",
        "resolved": "https://registry.npmjs.org/fstream/-/fstream-1.0.12.tgz",
        "integrity": "sha512-WvJ193OHa0GHPEL+AycEJgqvBEwyfRkN1vhjca23OaPVMCaLCXTd5qAu82Ajb63Ypde7raDIg==",
        "deprecated": "This package is no longer supported.",
        "license": "ISC",
        "dependencies": {
            "graceful-fs": "^4.1.2",
            "inherits": "~2.0.0",
            "mkdirp": ">=0.5 0",
            "rimraf": "2"
        },
        "engines": {
            "node": ">=0.6"
        }
    },
    "node_modules/glob": {
        "version": "7.2.3",
        "resolved": "https://registry.npmjs.org/glob/-/glob-7.2.3.tgz",
        "integrity": "sha512-nFR0zLpU2YCaRxwoCJvL6UvCH2JFyFVIvwTLslf21AuHIMskA1hhTdk+LIYJtOI2BGRqBL+iQK9Q==",
        "deprecated": "Glob versions prior to v9 are no longer supported",
        "license": "ISC",
        "dependencies": {
            "fs.realpath": "^1.0.0",
            "inflight": "^1.0.4",
            "inherits": "2",
            "minimatch": "^3.1.1",
            "once": "^1.3.0",
            "path-is-absolute": "^1.0.0"
        },
        "engines": {
            "node": "*"
        }
    },
    "funding": {
        "url": "https://github.com/sponsors/isaacs"
    }
},
    "node_modules/graceful-fs": {
        "version": "4.2.11",
        "resolved": "https://registry.npmjs.org/graceful-fs/-/graceful-fs-4.2.11.tgz",
        "integrity": "sha512-RbJ5/jmFcNNCcDV5o9eTnBLJ/HszWV0P73bc+Ff4nS/rJj+YaS6IGyiOL0VoBYX-h/KrH+nhJ0XvQ==",
        "license": "ISC"
    }
}
```

```
},
  "node_modules/ieee754": {
    "version": "1.2.1",
    "resolved": "https://registry.npmjs.org/ieee754/-/ieee754-1.2.1.tgz",
    "integrity": "sha512-dcyqhDvX1C46IXZcVqCpK+FtMRQVdIMN6/Df5js2zouUsqG7I6sFxitIC+7KYK29zL4sFnoVQnqaA==",
    "funding": [
      {
        "type": "github",
        "url": "https://github.com/sponsors/feross"
      },
      {
        "type": "patreon",
        "url": "https://www.patreon.com/feross"
      },
      {
        "type": "consulting",
        "url": "https://feross.org/support"
      }
    ],
    "license": "BSD-3-Clause"
  },
  "node_modules/immediate": {
    "version": "3.0.6",
    "resolved": "https://registry.npmjs.org/immediate/-/immediate-3.0.6.tgz",
    "integrity": "sha512-XXOFtyqDjNDAQxVfYxuF7g9lI/lbWmmIQg2MYKOH8ExIT1qg6xc4zyS3HaEEATciUiY7gjSXRGQ==",
    "license": "MIT"
  },
  "node_modules/inflight": {
    "version": "1.0.6",
    "resolved": "https://registry.npmjs.org/inflight/-/inflight-1.0.6.tgz",
    "integrity": "sha512-k92l/b08q4wvFscXCLvqfsHCrjrF7yiXsQuIVvVE7N82W3+aqpzuUdBbfhWcy/FZRgKL0svPDrGCJA==",
    "deprecated": "This module is not supported, and leaks memory. Do not use it. Check out Iru-ca che if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.",
    "license": "ISC",
    "dependencies": {
      "once": "^1.3.0",
      "wrappy": "1"
    }
  },
  "node_modules/inherits": {
    "version": "2.0.4",
    "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",
    "integrity": "sha512-k/vGaX4/YIa3WzyMCvTQOXYelHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSMxa4dYeZIQgewQ==",
    "license": "ISC"
  },
  "node_modules/isarray": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/isarray/-/isarray-1.0.0.tgz",
    "integrity": "sha512-VLghIWNM6ELQzo7zwmcg0NmTVyWKYjvleM83yjp0wRDTmUnrM678fQbcKB0ytg+TKla89ALQ==",
    "license": "MIT"
  },
  "node_modules/jszip": {
    "version": "3.10.1",
    "resolved": "https://registry.npmjs.org/jszip/-/jszip-3.10.1.tgz",
    "integrity": "sha512-xXDvecyTpGLrqFrkrUSoxxfJl5AH7U8zxxtVclpsUtMCq4JQ290LY8AW5c7GgnrbqK2qmtk3pN4g==",
    "license": "(MIT OR GPL-3.0-or-later)",
    "dependencies": {
      "lie": "~3.3.0",
      "pako": "~1.0.2",
      "readable-stream": "~2.3.6",
      "setimmediate": "^1.0.5"
    }
  },
  "node_modules/jszip/node_modules/readable-stream": {
    "version": "2.3.8",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
    "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrFQrPmum45RTtdA==",
    "license": "MIT",
  }
}
```

```
"dependencies": {
  "core-util-is": "~1.0.0",
  "inherits": "~2.0.3",
  "isarray": "~1.0.0",
  "process-nextick-args": "~2.0.0",
  "safe-buffer": "~5.1.1",
  "string_decoder": "~1.1.1",
  "util-deprecate": "~1.0.1"
}
},
"node_modules/jszip/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpscSmM/wPtojys4G6+g==",
  "license": "MIT"
},
"node_modules/jszip/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnDi6FHbbVfviro+WojFzv+s8MPMFHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJHyPyTP/mzRfwg==",
  "license": "MIT",
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/lazystream": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/lazystream/-/lazystream-1.0.1.tgz",
  "integrity": "sha512-b94GiNHQNy6JNTrt5w6zNyffMrNkXZb3KTkCZJb2V1xaEGCk093vkZ2jk3tpaeP3K9AxUebnf5nbw==",
  "license": "MIT",
  "dependencies": {
    "readable-stream": "^2.0.5"
  },
  "engines": {
    "node": ">= 0.6.3"
  }
},
"node_modules/lazystream/node_modules/readable-stream": {
  "version": "2.3.8",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EvnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRFQrPmum45RTtdA==",
  "license": "MIT",
  "dependencies": {
    "core-util-is": "~1.0.0",
    "inherits": "~2.0.3",
    "isarray": "~1.0.0",
    "process-nextick-args": "~2.0.0",
    "safe-buffer": "~5.1.1",
    "string_decoder": "~1.1.1",
    "util-deprecate": "~1.0.1"
  }
},
"node_modules/lazystream/node_modules/safe-buffer": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
  "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpscSmM/wPtojys4G6+g==",
  "license": "MIT"
},
"node_modules/lazystream/node_modules/string_decoder": {
  "version": "1.1.1",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
  "integrity": "sha512-n/ShnDi6FHbbVfviro+WojFzv+s8MPMFHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJHyPyTP/mzRfwg==",
  "license": "MIT",
  "dependencies": {
    "safe-buffer": "~5.1.0"
  }
},
"node_modules/lie": {
  "version": "3.3.0",
  "resolved": "https://registry.npmjs.org/lie/-/lie-3.3.0.tgz",
  "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EvnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRFQrPmum45RTtdA=="
}
```

"integrity": "sha512-UaiMJzeWRIEujAuw5LokY1L5ecNQYZKfmyZ9L7wDHb/p5etKaxXhohBcrw0EY
SN4N39dxHAIwQ==",
"license": "MIT",
"dependencies": {
"immediate": "~3.0.5"
}
,
"node_modules/listenercount": {
"version": "1.0.1",
"resolved": "https://registry.npmjs.org/listenercount/-/listenercount-1.0.1.tgz",
"integrity": "sha512-3mk/Zag0+lJxeDrxSgaDPy4zZ3w05PRZeJNnlWhzFz5OkX49J4krc+A8X2d2M69
8M+W+8gH+kBqQ==",
"license": "ISC"
},
"node_modules/lodash.defaults": {
"version": "4.2.0",
"resolved": "https://registry.npmjs.org/lodash.defaults/-/lodash.defaults-4.2.0.tgz",
"integrity": "sha512-qjxPLHd3r5DnsdGacqOMU6pb/avJzdh9tFX2ymgoZE27BmjXrNy/y4LoaiTeAb+C
tqfX/ae2leYYQ==",
"license": "MIT"
},
"node_modules/lodash.difference": {
"version": "4.5.0",
"resolved": "https://registry.npmjs.org/lodash.difference/-/lodash.difference-4.5.0.tgz",
"integrity": "sha512-dS2j+W26TQ7taQBGN8Lbbq04ssV3emRw4NY58WErIT029plqS0HmoT5aJ9+T
4eugsWwGp9yPA==",
"license": "MIT"
},
"node_modules/lodash.escaperegexp": {
"version": "4.1.2",
"resolved": "https://registry.npmjs.org/lodash.escaperegexp/-/lodash.escaperegexp-4.1.2.tgz",
"integrity": "sha512-TM9YBvyC84ZxE3rgfefxUWiQKLlstd6k7PTGt6wfbtXF8ixJLOL3VYyV/z+ZiPLsV
VwWIWeb2Y8Yyw==",
"license": "MIT"
},
"node_modules/lodash.flatten": {
"version": "4.4.0",
"resolved": "https://registry.npmjs.org/lodash.flatten/-/lodash.flatten-4.4.0.tgz",
"integrity": "sha512-C5N2Z3DgnnKr0LOpv/hKCgKdb7ZZwaflrsesve6lmzvZIRZRGaZ/I6Q8+2W7NaT
zCzrDCFdJfZ4q==",
"license": "MIT"
},
"node_modules/lodash.groupby": {
"version": "4.6.0",
"resolved": "https://registry.npmjs.org/lodash.groupby/-/lodash.groupby-4.6.0.tgz",
"integrity": "sha512-5dcWxm23+VAoz+awKmBaiBvzox8+RqMgFhi7UvX9DHZr2HdxHXM/Wrf8cfKps
SwNgurSILbmJw==",
"license": "MIT"
},
"node_modules/lodash.isboolean": {
"version": "3.0.3",
"resolved": "https://registry.npmjs.org/lodash.isboolean/-/lodash.isboolean-3.0.3.tgz",
"integrity": "sha512-Bz5mupy2SVbPHURB98VAcw+aHh4vRV5IPNhILUCsOzRmsTmSQ17jluqopAer
hbIXq980/1QJg==",
"license": "MIT"
},
"node_modules/lodash.isequal": {
"version": "4.5.0",
"resolved": "https://registry.npmjs.org/lodash.isequal/-/lodash.isequal-4.5.0.tgz",
"integrity": "sha512-pDo3lu8Jhfqls6GkMgpahsF9kCyayhgykjyLMNFTKWrpVdAQTyB4muAMWozB
MxLEI8wuz+DYQ==",
"deprecated": "This package is deprecated. Use require('node:util').isDeepStrictEqual instead.
",
"license": "MIT"
},
"node_modules/lodash.isfunction": {
"version": "3.0.9",
"resolved": "https://registry.npmjs.org/lodash.isfunction/-/lodash.isfunction-3.0.9.tgz",
"integrity": "sha512-AirXNj15uRIMMPihnklnB4i3NHeb4iBtNg9WRWuK2o31S+ePwwNmDPaTL3o7d
s4rxN4YU1oUJw==",
"license": "MIT"
},
"node_modules/lodash.isnil": {
"version": "4.0.0",
"resolved": "https://registry.npmjs.org/lodash.isnil/-/lodash.isnil-4.0.0.tgz",
"integrity": "sha512-up2Mzq3545mwvnMhTDMdfoG1OurpA/s5t88JmQX809eH3C8491iu2sfKhTfhQt

JUpT/dUDAAtng==",
 "license": "MIT"
},
 "node_modules/lodash.isplainobject": {
 "version": "4.0.6",
 "resolved": "https://registry.npmjs.org/lodash.isplainobject/-/lodash.isplainobject-4.0.6.tgz"
 "integrity": "sha512-oSXzaWypCMHkPC3NvBEaPHf0KsA5mvPrOPgQWDsb8n7orZ290M0BmC/jgF
YgdsW/F+MFOBA==",
 "license": "MIT"
 },
 "node_modules/lodash.isundefined": {
 "version": "3.0.1",
 "resolved": "https://registry.npmjs.org/lodash.isundefined/-/lodash.isundefined-3.0.1.tgz",
 "integrity": "sha512-MXB1is3s899/cD8jheYYE2V9qTHwKvt+npCwpD+1Sxm3Q3cECXCiYHjeHWXN
xendrO6goVTEA==",
 "license": "MIT"
 },
 "node_modules/lodash.union": {
 "version": "4.6.0",
 "resolved": "https://registry.npmjs.org/lodash.union/-/lodash.union-4.6.0.tgz",
 "integrity": "sha512-c4pB2CdGrGdjMKYLA+XiRDO7Y0PRQbm/Gzg8qMj+QH+pFVAoTp5sBpO0odL
sP+yQoiLoOBcw==",
 "license": "MIT"
 },
 "node_modules/lodash.uniq": {
 "version": "4.5.0",
 "resolved": "https://registry.npmjs.org/lodash.uniq/-/lodash.uniq-4.5.0.tgz",
 "integrity": "sha512-xfBaXQd9ryd9dlSDvnl0lxvfLJIYAZzXomUYzLKtUeOQvOP5piqAWuGtrhWeqaX
Jc5AV+XfsX3HQ==",
 "license": "MIT"
 },
 "node_modules/minimatch": {
 "version": "3.1.2",
 "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",
 "integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+BylrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFA
UO+K5aqBQOIHw==",
 "license": "ISC",
 "dependencies": {
 "brace-expansion": "^1.1.7"
 },
 "engines": {
 "node": "*"
 }
 },
 "node_modules/minimist": {
 "version": "1.2.8",
 "resolved": "https://registry.npmjs.org/minimist/-/minimist-1.2.8.tgz",
 "integrity": "sha512-2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVIDwDHbocAJveqqj1u8+S
pWqqWqAxb0zCA==",
 "license": "MIT",
 "funding": {
 "url": "https://github.com/sponsors/ljharb"
 }
 },
 "node_modules/mkdirp": {
 "version": "0.5.6",
 "resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-0.5.6.tgz",
 "integrity": "sha512-FP+p8RB8OWpF3YZBCrP5gtADmtXApB5AMLn+vdyA+PyxCjrCs00mjyUozssO3
xJ5M/YqtHAJw==",
 "license": "MIT",
 "dependencies": {
 "minimist": "^1.2.6"
 },
 "bin": {
 "mkdirp": "bin/cmd.js"
 }
 },
 "node_modules/ms": {
 "version": "2.1.3",
 "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
 "integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7I
SO/tXtF3WRtIA==",
 "license": "MIT"
 },
 "node_modules/normalize-path": {

```
"version": "3.0.0",
"resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz",
"integrity": "sha512-6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6D4WYojCRwcwLA==",
"license": "MIT",
"engines": {
  "node": ">=0.10.0"
}
},
"node_modules/once": {
  "version": "1.4.0",
  "resolved": "https://registry.npmjs.org/once/-/once-1.4.0.tgz",
  "integrity": "sha512-lNaJg1+2Q5URQBkccEKHTQOPaXdUxnZZEIQTZY0MFUAuaEqe1E+Nyvgdz/alyH8n58/GELp3+w==",
  "license": "ISC",
  "dependencies": {
    "wrappy": "1"
  }
},
"node_modules/pako": {
  "version": "1.0.11",
  "resolved": "https://registry.npmjs.org/pako/-/pako-1.0.11.tgz",
  "integrity": "sha512-4hLB8Py4zZce5s4yd9XzopqwVV/yGNhV1BI8NTmCq1763HeK2+EwVTv+leGeL1VPXCIO0z4taYW==",
  "license": "(MIT AND Zlib)"
},
"node_modules/path-is-absolute": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/path-is-absolute/-/path-is-absolute-1.0.1.tgz",
  "integrity": "sha512-AVbw3UJ2e9bq64vSaS9Am0fje1Pa8pbGqTTsmXfaliMpnri5DIDhfJOuLj9Sf95ZPk88MPmPe7UCQq==",
  "license": "MIT",
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/process-nextick-args": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-2.0.1.tgz"
},
"integrity": "sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbxuScTKiKJFFn1ag==",
"license": "MIT"
},
"node_modules/readable-stream": {
  "version": "3.6.2",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-3.6.2.tgz",
  "integrity": "sha512-9u/sniCrY3D5WdsERHzHE4G2YCXqoG5FTHUiCC4Slbr6XcLZBY05ya9EKjYekJdGBAS7Q9ScoA==",
  "license": "MIT",
  "dependencies": {
    "inherits": "^2.0.3",
    "string_decoder": "^1.1.1",
    "util-deprecate": "^1.0.1"
  },
  "engines": {
    "node": ">= 6"
  }
},
"node_modules/readdir-glob": {
  "version": "1.1.3",
  "resolved": "https://registry.npmjs.org/readdir-glob/-/readdir-glob-1.1.3.tgz",
  "integrity": "sha512-v05l2k7xN8zXvPD9N+z/uXPaj0sUFCe2rcWZIpBsqxP7xFQ0tipAd/wjj1YxWyVpOG82JKt2EAVA==",
  "license": "Apache-2.0",
  "dependencies": {
    "minimatch": "^5.1.0"
  }
},
"node_modules/readdir-glob/node_modules/brace-expansion": {
  "version": "2.0.2",
  "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.2.tgz",
  "integrity": "sha512-Jt0vHyM+jmUBqojB7E1NIYadt0vI0Qxjxd2TErW94wDz+E2LAm5vKMXXwg6ZZBXvjGBdfcF1ZDQ==",
  "license": "MIT",
  "dependencies": {
    "loose-regex": "1.0.2"
  }
}
```

```
        "balanced-match": "^1.0.0"
    },
    "node_modules/readaddr-glob/node_modules/minimatch": {
        "version": "5.1.6",
        "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-5.1.6.tgz",
        "integrity": "sha512-LKwV1brpG6mBUFtb7NUmtABCb2WZZmm2wNiOA5hAb8VdCS4B3dtMWyvccLt0zP1zXUN26g==",
        "license": "ISC",
        "dependencies": {
            "brace-expansion": "^2.0.1"
        },
        "engines": {
            "node": ">=10"
        }
    },
    "node_modules/rimraf": {
        "version": "2.7.1",
        "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-2.7.1.tgz",
        "integrity": "sha512-uWjbAK3T1OSVptzX7NI6PvQ3qAGtKEtVRjRuazjfL3Bx5eI409VZSqqND+4UNqFJNPqBZFve4w==",
        "deprecated": "Rimraf versions prior to v4 are no longer supported",
        "license": "ISC",
        "dependencies": {
            "glob": "^7.1.3"
        },
        "bin": {
            "rimraf": "bin.js"
        }
    },
    "node_modules/safe-buffer": {
        "version": "5.2.1",
        "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
        "integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPFuCXmpuYvmCXQ==",
        "funding": [
            {
                "type": "github",
                "url": "https://github.com/sponsors/feross"
            },
            {
                "type": "patreon",
                "url": "https://www.patreon.com/feross"
            },
            {
                "type": "consulting",
                "url": "https://feross.org/support"
            }
        ],
        "license": "MIT"
    },
    "node_modules/saxes": {
        "version": "5.0.1",
        "resolved": "https://registry.npmjs.org/saxes/-/saxes-5.0.1.tgz",
        "integrity": "sha512-5LBh1Tls8c9xgGjw3QrMwETmTMVk0oFgvrFSvWx62lIR2hcEInrKNZ2GZCCuuypeqc5hRYKF0Cw==",
        "license": "ISC",
        "dependencies": {
            "xmlchars": "^2.2.0"
        },
        "engines": {
            "node": ">=10"
        }
    },
    "node_modules/setimmediate": {
        "version": "1.0.5",
        "resolved": "https://registry.npmjs.org/setimmediate/-/setimmediate-1.0.5.tgz",
        "integrity": "sha512-MATJdZp8sLqDI/68LfQmbP8zKPLQNV6BIZolgrscFDQ+RsvK/BxeDQOgyxKKohCqQ/gd+reiXA==",
        "license": "MIT"
    },
    "node_modules/simple-git": {
        "version": "3.30.0",
        "resolved": "https://registry.npmjs.org/simple-git/-/simple-git-3.30.0.tgz",
        "integrity": "sha512-q6lxyDsCmEal/MEGhP1aVyQ3oxnagGIBDOVStB4XUVLI1iZh0Pah6ebC9V4xBaEKs0WS0rMEJHg==",
        "license": "MIT"
    }
}
```

```
"license": "MIT",
"dependencies": {
  "@kwsites/file-exists": "^1.1.1",
  "@kwsites/promise-deferred": "^1.1.1",
  "debug": "^4.4.0"
},
"funding": {
  "type": "github",
  "url": "https://github.com/steveukx/git-js?sponsor=1"
}
},
"node_modules/string_decoder": {
  "version": "1.3.0",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.3.0.tgz",
  "integrity": "sha512-hKRX8U1WjJFd8LsDJ2yQ/wWWxaopEsABU1XfkM8A+j0+85JAGppt16cr1Whg6
6BOj+up/wKeA==",
  "license": "MIT",
  "dependencies": {
    "safe-buffer": "~5.2.0"
  }
},
"node_modules/tar-stream": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/tar-stream/-/tar-stream-2.2.0.tgz",
  "integrity": "sha512-ujeqbceABgwMZxEJnk2HDY2DInUZ+9oEcb1KzTVfYHio0UE6dG71n60d8D2I4q
pt7vZeF1LnMZQ==",
  "license": "MIT",
  "dependencies": {
    "bl": "^4.0.3",
    "end-of-stream": "^1.4.1",
    "fs-constants": "^1.0.0",
    "inherits": "^2.0.3",
    "readable-stream": "^3.1.1"
  },
  "engines": {
    "node": ">=6"
  }
},
"node_modules/tmp": {
  "version": "0.2.5",
  "resolved": "https://registry.npmjs.org/tmp/-/tmp-0.2.5.tgz",
  "integrity": "sha512-voyz6MApa1rQGUxT3E+BK7/ROe8itEx7vD8/HEvt4xwXucvQ5G5oeEiHkmHZJu
m9MOivj709jow==",
  "license": "MIT",
  "engines": {
    "node": ">=14.14"
  }
},
"node_modules/traverse": {
  "version": "0.3.9",
  "resolved": "https://registry.npmjs.org/traverse/-/traverse-0.3.9.tgz",
  "integrity": "sha512-iawgk0hLP3SxGKDfnDJf8wTz4p2qlmnyihM5Hh/sGvQ3K37dPi/w8sRhndNIxYA1T
Jq+O0RsvXBKdQ==",
  "license": "MIT/X11",
  "engines": {
    "node": "*"
  }
},
"node_modules/unzipper": {
  "version": "0.10.14",
  "resolved": "https://registry.npmjs.org/unzipper/-/unzipper-0.10.14.tgz",
  "integrity": "sha512-ti4wZJ+0bQTiX2KmKWuwj7lhV+2n//uXEotUmGuQqrBvZSEGFMBI68+c6JCQ8aA
A8K6wXvneR/6g==",
  "license": "MIT",
  "dependencies": {
    "big-integer": "^1.6.17",
    "binary": "~0.3.0",
    "bluebird": "~3.4.1",
    "buffer-indexof-polyfill": "~1.0.0",
    "duplexer2": "~0.1.4",
    "fstream": "^1.0.12",
    "graceful-fs": "^4.2.2",
    "listenercount": "~1.0.1",
    "readable-stream": "~2.3.6",
    "setimmediate": "~1.0.4"
  }
}
```

```
},
  "node_modules/unzipper/node_modules/readable-stream": {
    "version": "2.3.8",
    "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",
    "integrity": "sha512-8p0AUk4XODglewSi0l8Epjs+EvnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5IRrFQrPmum45RTtdA==",
    "license": "MIT",
    "dependencies": {
      "core-util-is": "^1.0.0",
      "inherits": "~2.0.3",
      "isarray": "~1.0.0",
      "process-nextick-args": "~2.0.0",
      "safe-buffer": "~5.1.1",
      "string_decoder": "~1.1.1",
      "util-deprecate": "~1.0.1"
    }
  },
  "node_modules/unzipper/node_modules/safe-buffer": {
    "version": "5.1.2",
    "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",
    "integrity": "sha512-Gd2UZBJDkXIY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpSCSmM/wPtojs4G6+g==",
    "license": "MIT"
  },
  "node_modules/unzipper/node_modules/string_decoder": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.1.1.tgz",
    "integrity": "sha512-n/ShnDi6FHbbVfviro+WojIFzv+s8MPMHBCzVePfUpDJLwoLT0ht1I4YwBCbi8pJHyPyTP/mzRfwg==",
    "license": "MIT",
    "dependencies": {
      "safe-buffer": "~5.1.0"
    }
  },
  "node_modules/util-deprecate": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
    "integrity": "sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAlil2TaSCA7VCJ1UJDMagCzlIkXNTEQ8xilbrHtcw==",
    "license": "MIT"
  },
  "node_modules/uuid": {
    "version": "8.3.2",
    "resolved": "https://registry.npmjs.org/uuid/-/uuid-8.3.2.tgz",
    "integrity": "sha512-+NYs2QeMWy+GWFOEm9xnn6HCDp0I7QBD7ml8zLUmJ+93Q5NF0NocErnwkS8Dhb/gz7R7eg==",
    "license": "MIT",
    "bin": {
      "uuid": "dist/bin/uuid"
    }
  },
  "node_modules/wrappy": {
    "version": "1.0.2",
    "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
    "integrity": "sha512-l4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KIPS1xUsZ3u7/IQO4wPRcR3QCvezPcQ==",
    "license": "ISC"
  },
  "node_modules/xmlchars": {
    "version": "2.2.0",
    "resolved": "https://registry.npmjs.org/xmlchars/-/xmlchars-2.2.0.tgz",
    "integrity": "sha512-JZnDKK8B0RCdw84FNdDAIpZK+JuJw+s7Lz8nksI7SluU3UXJJslUthsi+uWBUYLRFUKpxjtjFCw==",
    "license": "MIT"
  },
  "node_modules/zip-stream": {
    "version": "4.1.1",
    "resolved": "https://registry.npmjs.org/zip-stream/-/zip-stream-4.1.1.tgz",
    "integrity": "sha512-9qv4riDiopXg4E69k+vMHjNN63YFMe9sZMrdivKnCjlCRWeCBswPPMPUfx+ipsAcbahdJjpS6hyQ==",
    "license": "MIT",
    "dependencies": {
      "archiver-utils": "^3.0.4",
      "compress-commons": "^4.1.2",
      "readable-stream": "^3.6.0"
    }
  }
```

```
"engines": {
  "node": ">= 10"
}
},
"node_modules/zip-stream/node_modules/archiver-utils": {
  "version": "3.0.4",
  "resolved": "https://registry.npmjs.org/archiver-utils/-/archiver-utils-3.0.4.tgz",
  "integrity": "sha512-KVgf4XQVrTjhyWmx6cte4RxonPLR9onExufI1jhvw/MQ4BB6lsZD5gT8Lq+u/+pR
piQioaqFP5Rzw==",
  "license": "MIT",
  "dependencies": {
    "glob": "^7.2.3",
    "graceful-fs": "^4.2.0",
    "lazystream": "^1.0.0",
    "lodash.defaults": "^4.2.0",
    "lodash.difference": "^4.5.0",
    "lodash.flatten": "^4.4.0",
    "lodash.isplainobject": "^4.0.6",
    "lodash.union": "^4.6.0",
    "normalize-path": "^3.0.0",
    "readable-stream": "^3.6.0"
  },
  "engines": {
    "node": ">= 10"
  }
}
}
```

package.json

```
{  
  "name": "change-management-automation",  
  "version": "0.1.0",  
  "description": "Automação para gerar planilhas de Change Management a partir dos dados do repo e git",  
  "main": "generate_change_management.js",  
  "scripts": {  
    "generate": "node generate_change_management.js"  
  },  
  "dependencies": {  
    "exceljs": "^4.3.0",  
    "simple-git": "^3.19.0"  
}
```

compile_docs_to_pdf.py

```
#!/usr/bin/env python3
"""
compile_docs_to_pdf.py
Recursivamente percorre `docs/` e converte todos os arquivos para PDF e mescla em um único PDF (Project
Documentation.pdf).
Suporta: .md, .txt, .pdf, .xlsx, imagens (.png, .jpg).
"""

import argparse
import io
import os
import re
import sys
import tempfile
from pathlib import Path
from datetime import datetime
import pandas as pd
from pypdf import PdfReader, PdfWriter
from reportlab.lib.pagesizes import A4
from reportlab.lib.units import mm
from reportlab.lib import utils
from reportlab.pdfgen import canvas
from PIL import Image
import markdown as md

PAGE_WIDTH, PAGE_HEIGHT = A4

def convert_text_to_pdf(text: str, out_path: Path, title: str = None):
    c = canvas.Canvas(str(out_path), pagesize=A4)
    margin = 15 * mm
    x = margin
    y = PAGE_HEIGHT - margin
    if title:
        c.setFont('Helvetica-Bold', 14)
        c.drawString(x, y, title)
        y -= 14 + 6
    c.setFont('Helvetica', 10)
    lines = text.splitlines()
    for line in lines:
        # Simple wrap
        if not line:
            continue
        if y < margin:
            c.showPage()
            y = PAGE_HEIGHT - margin
        continue
    while line:
        # fit in width
        max_chars = 100
        part = line[:max_chars]
        c.drawString(x, y, part)
        line = line[max_chars:]
        y -= 10
        if y < margin:
            c.showPage()
            y = PAGE_HEIGHT - margin
    c.save()

def convert_df_to_pdf(df: pd.DataFrame, out_path: Path, title: str = None):
    text = ""
    if title:
        text += f'{title}\n\n'
    text += df.to_string(index=False)
    return convert_text_to_pdf(text, out_path)

def convert_image_to_pdf(img_path: Path, out_path: Path):
    image = Image.open(img_path)
    image = image.convert('RGB')
    # Fit to A4
    width, height = image.size
    ratio = min(PAGE_WIDTH / width, PAGE_HEIGHT / height)
```

```

new_w = int(width * ratio)
new_h = int(height * ratio)
image = image.resize((new_w, new_h), Image.Resampling.LANCZOS)
image.save(out_path, format='PDF')

def convert_md_to_pdf(md_path: Path, out_path: Path):
    text = md.markdown(md_path.read_text(encoding='utf8'))
    # Strip HTML to plaintext (simple)
    plain = re.sub(r'<[^>]+>', "", text)
    return convert_text_to_pdf(plain, out_path, title=md_path.name)

def convert_xlsx_to_pdf(xlsx_path: Path, out_path: Path):
    # read all sheets and write each as a string
    xls = pd.read_excel(xlsx_path, sheet_name=None)
    # write a temporary multi-page pdf
    c = canvas.Canvas(str(out_path), pagesize=A4)
    margin = 15 * mm
    x = margin
    y = PAGE_HEIGHT - margin
    for sheet_name, df in xls.items():
        c.setFont('Helvetica-Bold', 12)
        c.drawString(x, y, f'{xlsx_path.name} - {sheet_name}')
        y -= 16
        c.setFont('Helvetica', 9)
        table_text = df.to_string(index=False)
        for line in table_text.splitlines():
            # Basic wrapping
            c.drawString(x, y, line[:150])
            y -= 11
            if y < margin:
                c.showPage()
                y = PAGE_HEIGHT - margin
    # Spacer between sheets
    y -= 10
    if y < margin:
        c.showPage()
        y = PAGE_HEIGHT - margin
    c.save()

def collect_docs(root: Path):
    files = []
    ignore_dirs = {'node_modules', '.venv', 'venv', '.git'}
    for p in root.rglob('*'):
        if any(x in p.parts for x in ignore_dirs):
            continue
        if p.is_file():
            files.append(p)
    # sort for stable order
    files.sort()
    return files

def build_project_pdf(docs_root: Path, out_file: Path, temp_dir: Path):
    files = collect_docs(docs_root)
    print(f'Found {len(files)} files under {docs_root}')
    tmp_pdfs = []
    for p in files:
        ext = p.suffix.lower()
        tmp_pdf = temp_dir / (p.name + '.pdf')
        try:
            if ext == '.pdf':
                # copy directly
                import shutil
                shutil.copy2(p, tmp_pdf)
            elif ext in ['.md']:
                convert_md_to_pdf(p, tmp_pdf)
            elif ext in ['.txt']:
                convert_text_to_pdf(p.read_text(encoding='utf8'), tmp_pdf, title=p.name)
            elif ext in ['.png', '.jpg', '.jpeg']:
                convert_image_to_pdf(p, tmp_pdf)
            elif ext in ['.xlsx', '.xls']:
                convert_xlsx_to_pdf(p, tmp_pdf)
            else:

```

```

# fallback: plain text
try:
    text = p.read_text(encoding='utf8')
    convert_text_to_pdf(text, tmp_pdf, title=p.name)
except Exception:
    print('Unsupported file type for', p)
    continue
tmp_pdfs.append(tmp_pdf)
print('Converted', p, '->', tmp_pdf)
except Exception as e:
    print('Failed to convert', p, e)

# Merge all temp pdfs
writer = PdfWriter()
for t in tmp_pdfs:
    try:
        reader = PdfReader(str(t))
        for page in reader.pages:
            writer.add_page(page)
    except Exception as e:
        print('Failed to append', t, e)
with open(out_file, 'wb') as f:
    writer.write(f)
print('Wrote merged PDF to', out_file)

def main():
    parser = argparse.ArgumentParser(description='Compile docs/ files into a single PDF')
    parser.add_argument('--out', '-o', default='PROJECT_DOCUMENTATION.pdf')
    parser.add_argument('--docs-root', default='docs')
    args = parser.parse_args()
    docs_root = Path(args.docs_root).resolve()
    out_file = Path(args.out).resolve()
    temp_dir = Path(tempfile.mkdtemp(prefix='docs_pdf_'))
    build_project_pdf(docs_root, out_file, temp_dir)

if __name__ == '__main__':
    main()

```

merge_docs_to_pdf.py

```
#!/usr/bin/env python3
"""
merge_docs_to_pdf.py
Merge DOCS folder files into a single PDF (existing PDFs, images, short summaries of non-PDFs)
"""

import sys
from pathlib import Path
from typing import List

from PIL import Image
from pypdf import PdfReader, PdfWriter
import pandas as pd

def is_image(path: Path) -> bool:
    return path.suffix.lower() in ['.png', '.jpg', '.jpeg', '.gif', '.bmp']

def is_pdf(path: Path) -> bool:
    return path.suffix.lower() == '.pdf'

def summarize_xlsx(path: Path) -> str:
    try:
        xls = pd.read_excel(path, sheet_name=None)
        out = []
        for sname, df in xls.items():
            out.append(f"Sheet: {sname}")
            out.append(df.head(5).to_csv(index=False))
            out.append('\n')
        return '\n'.join(out)
    except Exception as e:
        return f"(Failed to read xlsx: {e})"

def textpage_to_pdf(text: str, out: Path):
    # Create a simple PDF page from text using Pillow
    # Compute size
    lines = text.splitlines()
    width = 1200
    line_height = 20
    height = max(1200, line_height * (len(lines) + 5))
    img = Image.new('RGB', (width, height), color='white')
    from PIL import ImageDraw, ImageFont
    draw = ImageDraw.Draw(img)
    try:
        font = ImageFont.truetype('DejaVuSans.ttf', 14)
    except Exception:
        font = ImageFont.load_default()
    y = 10
    for l in lines:
        draw.text((10, y), l[:2000], fill='black', font=font)
        y += line_height
    img.save(out, 'PDF')

def main():
    repo_root = Path(__file__).resolve().parent
    docs_dir = repo_root
    out_file = docs_dir / 'CONSOLIDATED_DOCS_FOR_AI.pdf'
    writer = PdfWriter()

    # Find pdfs and images
    all_files: List[Path] = sorted([p for p in docs_dir.iterdir() if p.is_file()], key=lambda p: p.name.lower())

    # merge pdfs first that already exist
    for f in all_files:
        if is_pdf(f):
            try:
                r = PdfReader(str(f))
                for page in r.pages:
                    writer.add_page(page)
                print('Appended PDF', f)
```

```

except Exception as e:
    print('Failed append pdf', f, e)

# then convert images into pdf and append
tmp_dir = docs_dir / '.tmp_merge_pdf'
tmp_dir.mkdir(exist_ok=True)
for f in all_files:
    if is_image(f):
        # convert image to temporary PDF
        tmp_pdf = tmp_dir / (f.stem + '.pdf')
        try:
            image.open(f).convert('RGB').save(tmp_pdf, 'PDF', resolution=100.0)
            r = PdfReader(str(tmp_pdf))
            for page in r.pages:
                writer.add_page(page)
            print('Appended image as pdf', f)
        except Exception as e:
            print('Failed convert image', f, e)

# non-pdf, non-image files: create summarized text pdfs (xlsx -> summary, md/txt -> include)
for f in all_files:
    if not (is_pdf(f) or is_image(f)):
        if f.suffix.lower() in ['.md', '.txt']:
            text = f"File: {f.name}\n\n" + f.read_text(encoding='utf8')
        elif f.suffix.lower() in ['.xlsx', '.xls']:
            text = f"File: {f.name}\n\n" + summarize_xlsx(f)
        else:
            # other types: just show file name and size
            text = f"File: {f.name}\n\nType: {f.suffix}\nSize: {f.stat().st_size} bytes\nPath: {f}\n"
        tmp_pdf = tmp_dir / (f.stem + '_summary.pdf')
        try:
            textpage_to_pdf(text, tmp_pdf)
            r = PdfReader(str(tmp_pdf))
            for page in r.pages:
                writer.add_page(page)
            print('Appended summary for', f)
        except Exception as e:
            print('Failed create textpdf', f, e)

# write merged output
try:
    with open(out_file, 'wb') as f:
        writer.write(f)
    print('Wrote consolidated PDF to', out_file)
finally:
    # cleanup tmp files
    import shutil
    try:
        shutil.rmtree(tmp_dir)
    except Exception:
        pass

if __name__ == '__main__':
    main()

```

requirements-pdf.txt

```
pypdf>=3.0.0  
Pillow>=10.0.0  
pandas>=2.2.2  
openpyxl>=3.1.2
```