

1. Plano de projeto

Projeto:

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

Registro de Alterações

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

1.1 Introdução

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

1.2 Escopo do projeto

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

Justificativa:

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

Produto:

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

Descrição Preliminar do Sistema (breve):

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

1.3 Descrição Preliminar do Sistema

- Hardware disponível: ESP32 (com/câmera), ESP32-S3/ESP32-CAM, Raspberry Pi Pico, HC-SR04, motores de vibração, estrutura wearable, smartphone Android.
- Performance: latência ≤ 400 ms do evento ao alerta; autonomia mínima de prova ≥ 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: $\text{Total de funcionalidades concluídas} / \text{Total previstas na sprint}$
- Procedimento: Verificação dos tickets finalizados no Github
- Momento: Ao fim de cada sprint
- Responsável pela Medição: Scrum Master
- Procedimento de Análise: Comparação com metas da sprint
- Responsável pela Análise: Gerente de Projeto

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