

# Prototipação em Engenharia



## Portfólio

**Aluno: Marco Antonio Monteiro Pedro**

<https://github.com/marcosamambaia/CamisaParaCego>

# Sumário

Acessibilidade e Independência para Pessoas Cegas: A Solução da Camisa Sensorial.....	3
Funcionamento da Camisa Sensorial .....	3
Detecção de Obstáculos .....	3
Ativação dos Motores de Vibração .....	4
Controle Inteligente via ESP32 .....	4
Projeto Técnico: Camisa Sensorial para Pessoas Cegas com ESP32 .....	5
Objetivo .....	5
Arquitetura do Sistema .....	5
Distribuição dos Sensores e Motores .....	5
Esquema no Protheus Schematic: .....	6
Esquema no Protheus PCB: .....	7
Código-Fonte .....	8
Viabilidade Técnica .....	12
Viabilidade Econômica .....	13

# Acessibilidade e Independência para Pessoas Cegas: A Solução da Camisa Sensorial

A vida de uma pessoa cega é repleta de desafios diários. A locomoção em ambientes urbanos pode ser complexa, exigindo atenção constante para evitar obstáculos como postes, paredes, veículos e até mesmo pedestres. A ausência de uma percepção visual limita a segurança e pode dificultar a independência. Muitas pessoas cegas dependem de bengalas, cães-guia e aplicativos de acessibilidade, mas nem sempre esses recursos são suficientes para garantir uma experiência fluida e intuitiva.

Pensando nisso, a **camisa sensorial baseada no ESP32** surge como uma solução inovadora para aumentar a autonomia das pessoas cegas. Utilizando **sensores ultrassônicos HC-SR04**, a camisa consegue detectar obstáculos próximos e alertar o usuário por meio de **motores de vibração** estrategicamente posicionados. O sistema conta com **quatro sensores**, localizados nos ombros, na frente e nas costas, proporcionando uma percepção espacial mais ampla.

## Funcionamento da Camisa Sensorial

A **camisa sensorial** utiliza **sensores ultrassônicos HC-SR04** e **motores de vibração** para detectar obstáculos e fornecer feedback tátil ao usuário. O sistema foi projetado para auxiliar pessoas cegas na locomoção, oferecendo uma maneira intuitiva de perceber o ambiente ao redor.

## Detecção de Obstáculos

Cada sensor ultrassônico emite **pulsos sonoros** imperceptíveis ao ouvido humano.

O som reflete nos objetos próximos e retorna ao sensor.

O tempo entre a emissão e o retorno do som é medido e convertido em **distância**.

Quanto menor a distância detectada, **mais próximo está o obstáculo**.

### Ativação dos Motores de Vibração

O sistema compara a **distância medida** com um limite predefinido.

Quando um obstáculo é identificado **a menos de 50 cm**, o **motor correspondente** é ativado.

Quanto **mais perto** o objeto estiver, **mais intensa será a vibração**, garantindo um feedback tátil proporcional à proximidade do obstáculo.

### Controle Inteligente via ESP32

O **ESP32** processa os dados dos sensores e ajusta a vibração dos motores em tempo real.

A intensidade da vibração é regulada através de **PWM (modulação por largura de pulso)**.

Isso permite que o usuário perceba objetos próximos com diferentes **níveis de alerta**.

**Facilidade de uso** – Basta vestir a camisa para que ela comece a auxiliar na navegação do ambiente.

**Maior segurança e independência** – Redução do risco de colisões e acidentes, permitindo que a pessoa se mova com mais confiança.

Essa tecnologia pode ser integrada a outros dispositivos assistivos, criando um sistema robusto para melhorar a qualidade de vida das pessoas cegas. Com mais desenvolvimento e testes, podemos aperfeiçoar essa inovação e torná-la acessível para um público maior.

A **camisa sensorial** representa um grande passo para a acessibilidade e autonomia, trazendo um mundo mais seguro e inclusivo para aqueles que enfrentam dificuldades diárias de locomoção.

# Projeto Técnico: Camisa Sensorial para Pessoas Cegas com ESP32

## Objetivo

Desenvolver um sistema vestível baseado em **ESP32 (NodeMCU-32S)**, utilizando **sensores ultrassônicos HC-SR04** e **motores de vibração**, para auxiliar pessoas cegas na detecção de obstáculos ao redor.

## Arquitetura do Sistema

**Placa de controle:** ESP32 (NodeMCU-32S) – Responsável pelo processamento e comunicação entre os sensores e atuadores.  
**Sensores ultrassônicos** (HC-SR04) – Mede a distância de obstáculos.  
**Motores de vibração** – Alertam o usuário sobre obstáculos próximos.  
**Fonte de alimentação** – Bateria recarregável para mobilidade.

## Distribuição dos Sensores e Motores

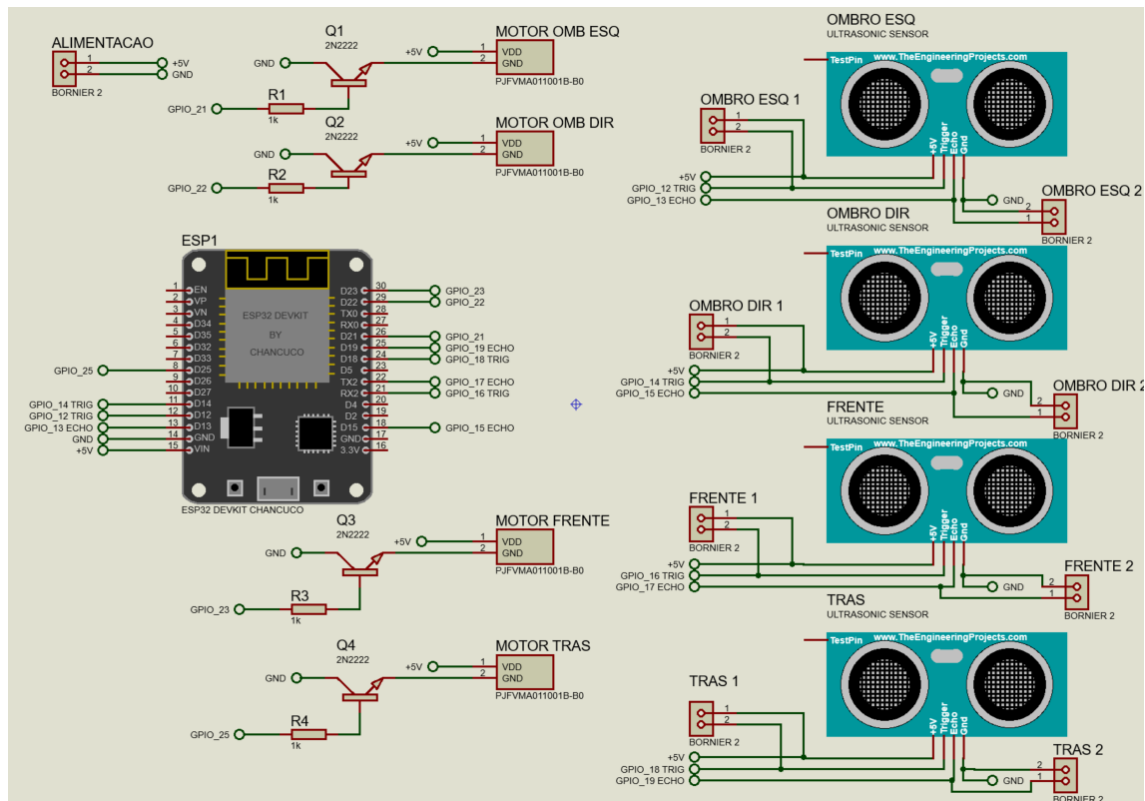
**Sensor Ombro Esquerdo** → GPIO 12 (TRIG) / GPIO 13 (ECHO) → Motor GPIO 21

**Sensor Ombro Direito** → GPIO 14 (TRIG) / GPIO 15 (ECHO) → Motor GPIO 22

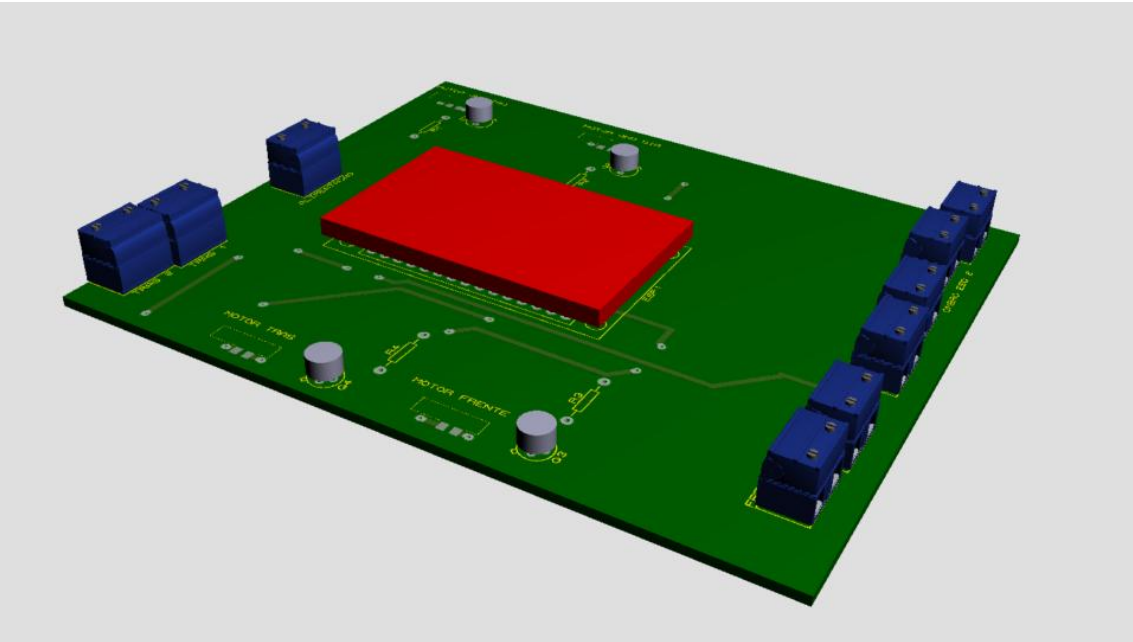
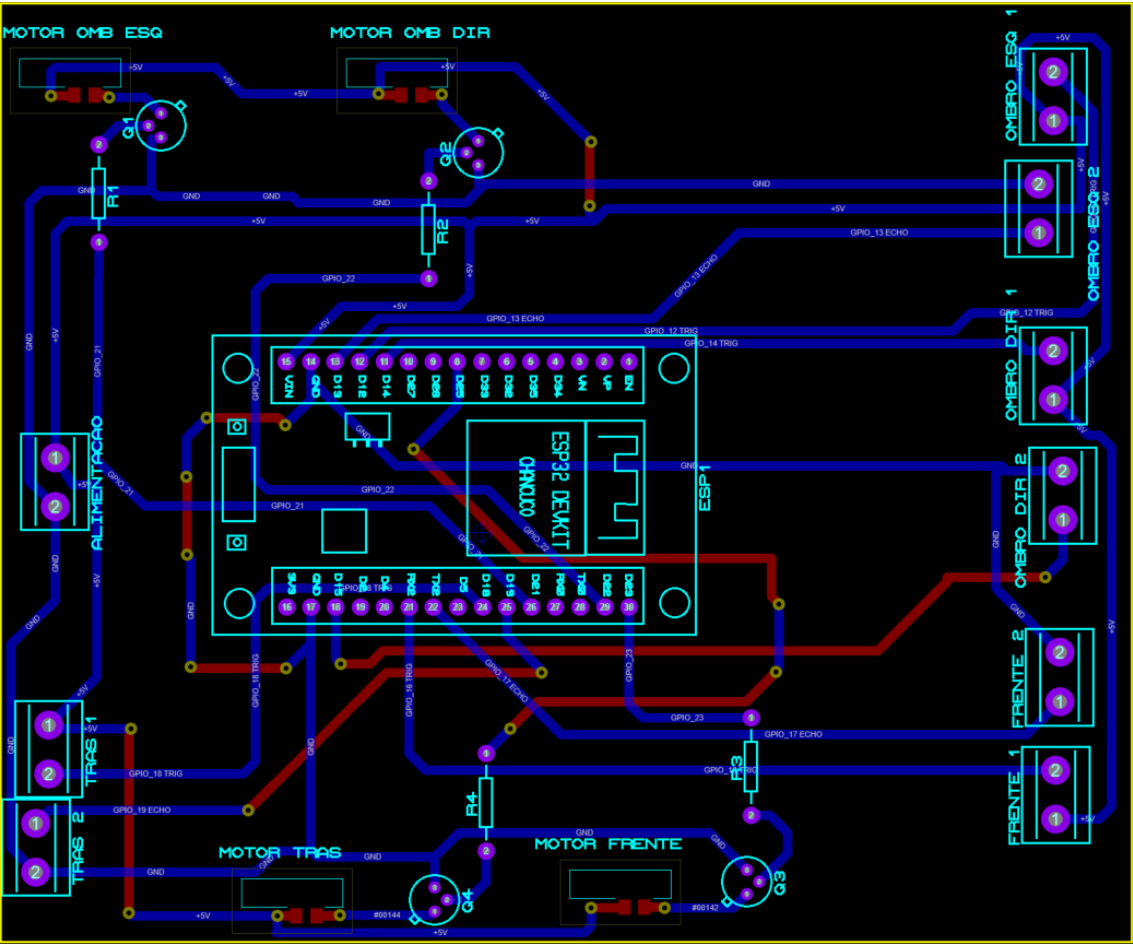
**Sensor Frontal** → GPIO 16 (TRIG) / GPIO 17 (ECHO) → Motor GPIO 23

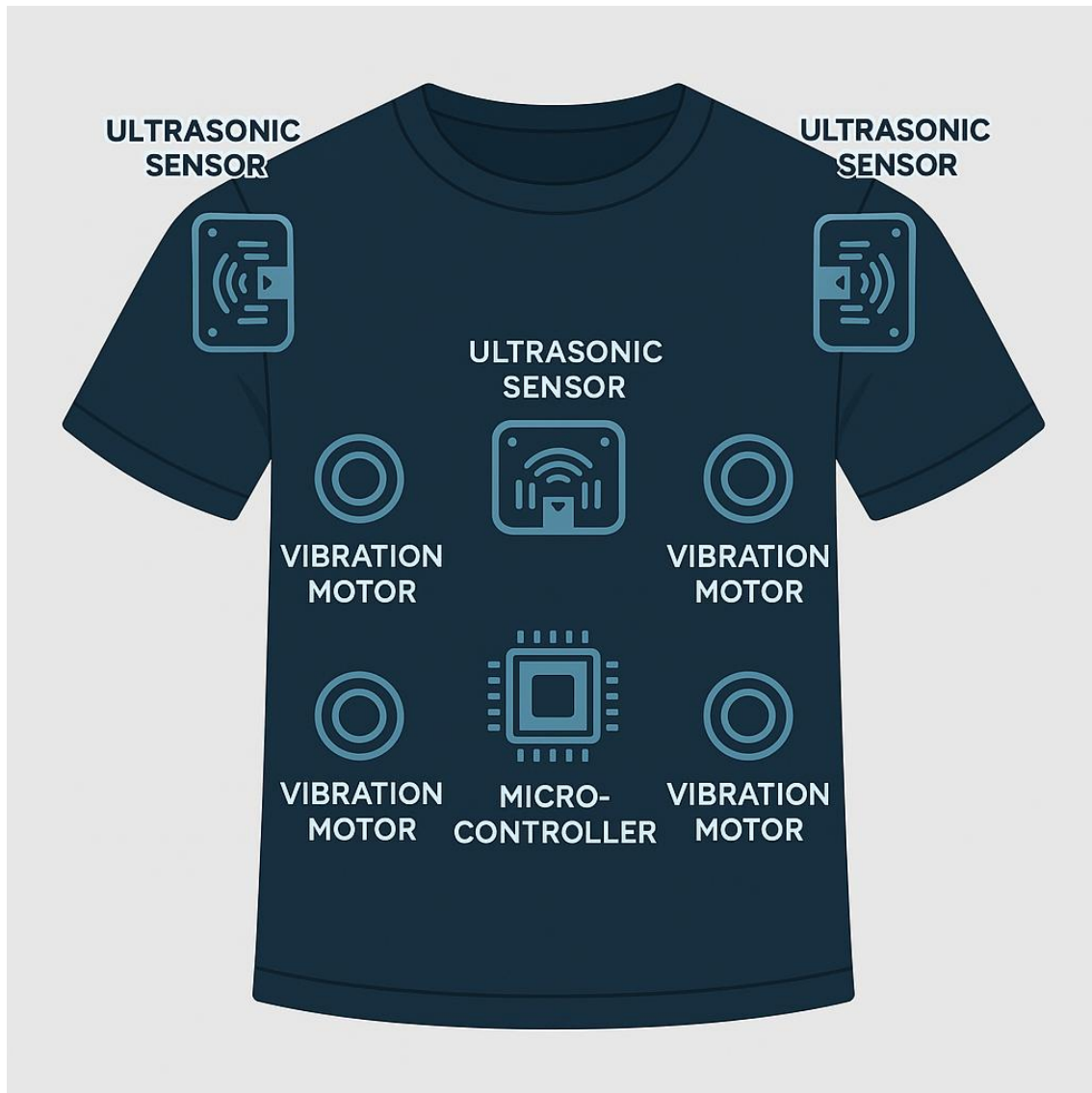
**Sensor Traseiro** → GPIO 18 (TRIG) / GPIO 19 (ECHO) → Motor GPIO 25

## Esquema no Protheus Schematic:



Esquema no Protheus PCB:





## Código-Fonte

O código é implementado em **C++**, utilizando **PlatformIO no VS Code**. A estrutura inclui:

**Biblioteca Arduino.h** para controle do ESP32

**Função medirDistancia(trigPin, echoPin)** para calcular a distância dos sensores

**Condicionais digitalWrite(motor, HIGH/LOW)** para ativação dos motores.



Segue o código:

```
#include <Arduino.h> // Biblioteca necessária para usar funções do Arduino
```

```
// Definição dos pinos dos sensores ultrassônicos HC-SR04
```

```
#define TRIG_ESQUERDO 12 // Sensor no ombro esquerdo
```

```
#define ECHO_ESQUERDO 13
```

```
#define TRIG_DIREITO 14 // Sensor no ombro direito
```

```
#define ECHO_DIREITO 15
```

```
#define TRIG_FRENTE 16 // Sensor frontal
```

```
#define ECHO_FRENTE 17
```

```
#define TRIG_TRAS 18 // Sensor traseiro
```

```
#define ECHO_TRAS 19
```

```
// Definição dos pinos dos motores de vibração
```

```
#define MOTOR_ESQUERDO 21 // Motor do ombro esquerdo
```

```
#define MOTOR_DIREITO 22 // Motor do ombro direito
```

```
#define MOTOR_FRENTE 23 // Motor frontal
```

```
#define MOTOR_TRAS 25 // Motor traseiro
```

```
// Configuração dos canais PWM no ESP32
```

```
#define PWM_FREQUENCIA 5000 // Frequência de PWM
```

```
#define PWM_RESOLUCAO 8 // Resolução de 8 bits (valores de 0 a 255)
```

```
// Função para medir distância dos sensores ultrassônicos
```

```
int medirDistancia(int trigPin, int echoPin) {
```

```

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);


long duration = pulseIn(echoPin, HIGH); // Captura o tempo do pulso do sensor

return duration * 0.034 / 2; // Retorna a distância calculada em cm
}


void setup() {
    Serial.begin(115200); // Inicializa a comunicação serial


    // Configuração dos sensores ultrassônicos como entrada e saída
    pinMode(TRIG_ESQUERDO, OUTPUT);
    pinMode(ECHO_ESQUERDO, INPUT);


    pinMode(TRIG_DIREITO, OUTPUT);
    pinMode(ECHO_DIREITO, INPUT);


    pinMode(TRIG_FRENTE, OUTPUT);
    pinMode(ECHO_FRENTE, INPUT);


    pinMode(TRIG_TRAS, OUTPUT);
    pinMode(ECHO_TRAS, INPUT);


    // Configuração dos motores de vibração como saída PWM
    ledcSetup(0, PWM_FREQUENCIA, PWM_RESOLUCAO);
    ledcSetup(1, PWM_FREQUENCIA, PWM_RESOLUCAO);

```

```
ledcSetup(2, PWM_FREQUENCIA, PWM_RESOLUCAO);
ledcSetup(3, PWM_FREQUENCIA, PWM_RESOLUCAO);

// Associação dos motores aos canais PWM
ledcAttachPin(MOTOR_ESQUERDO, 0);
ledcAttachPin(MOTOR_DIREITO, 1);
ledcAttachPin(MOTOR_FRENTE, 2);
ledcAttachPin(MOTOR_TRAS, 3);
}

void loop() {
    // Medindo distância de cada sensor
    int distanciaEsquerdo = medirDistancia(TRIG_ESQUERDO, ECHO_ESQUERDO);
    int distanciaDireito = medirDistancia(TRIG_DIREITO, ECHO_DIREITO);
    int distanciaFrente = medirDistancia(TRIG_FRENTE, ECHO_FRENTE);
    int distanciaTras = medirDistancia(TRIG_TRAS, ECHO_TRAS);

    // Mapeando intensidade da vibração (quanto mais perto, maior a vibração)
    int intensidadeEsquerdo = map(distanciaEsquerdo, 50, 5, 0, 255);
    int intensidadeDireito = map(distanciaDireito, 50, 5, 0, 255);
    int intensidadeFrente = map(distanciaFrente, 50, 5, 0, 255);
    int intensidadeTras = map(distanciaTras, 50, 5, 0, 255);

    // Limitando intensidade mínima e máxima da vibração para conforto do usuário
    intensidadeEsquerdo = constrain(intensidadeEsquerdo, 0, 255);
    intensidadeDireito = constrain(intensidadeDireito, 0, 255);
    intensidadeFrente = constrain(intensidadeFrente, 0, 255);
    intensidadeTras = constrain(intensidadeTras, 0, 255);
}
```

```
// Aplicando intensidade nos motores usando PWM (LEDC no ESP32)

ledcWrite(0, intensidadeEsquerdo);

ledcWrite(1, intensidadeDireito);

ledcWrite(2, intensidadeFrente);

ledcWrite(3, intensidadeTras);


Serial.print("Esq: "); Serial.print(distanciaEsquerdo);

Serial.print(" | Dir: "); Serial.print(distanciaDireito);

Serial.print(" | Frontal: "); Serial.print(distanciaFrente);

Serial.print(" | Traseiro: "); Serial.println(distanciaTras);


delay(100); // Pequena pausa para estabilizar leituras

}
```

## Viabilidade Técnica

O projeto da **camisa sensorial** é tecnicamente viável devido à acessibilidade dos componentes e à sua fácil integração com sistemas embarcados. Os principais pontos técnicos a serem considerados incluem:

**Hardware compatível** – O ESP32 oferece conectividade e processamento adequado para controle de sensores ultrassônicos e motores de vibração.

**Sensores eficientes** – O HC-SR04 realiza medições precisas de distância, permitindo a detecção de obstáculos em tempo real.

**Atuação confiável** – Os motores de vibração garantem um feedback tátil intuitivo para o usuário, aumentando a segurança na locomoção.

**Facilidade de programação** – O desenvolvimento do firmware é viável utilizando **PlatformIO** e bibliotecas **Arduino**, permitindo ajustes rápidos no código.

**Montagem modular** – O projeto pode ser montado com **placas PCB**

**customizadas**, facilitando a produção e manutenção.

**Baixo consumo energético** – O sistema pode ser alimentado por **baterias recarregáveis**, garantindo autonomia sem necessidade de fios.

**Desafios técnicos:** Ajustar a alimentação dos motores para evitar sobrecarga no ESP32, melhorar a eficiência da comunicação entre sensores e explorar alternativas como **Bluetooth** para futuras expansões.

## Viabilidade Econômica

A implementação do projeto apresenta um **custo acessível** e pode ser viabilizada comercialmente. Os principais aspectos econômicos são:

**Baixo custo de componentes** – Os sensores **HC-SR04**, motores de vibração e ESP32 são relativamente baratos e amplamente disponíveis no mercado.

**Produção escalável** – A fabricação da PCB personalizada pode reduzir custos de montagem e simplificar a produção em larga escala.

**Alta demanda no mercado** – Tecnologias assistivas são um **setor em expansão**, tornando o projeto comercialmente promissor para inclusão social e acessibilidade.

**Alternativas de financiamento** – O projeto pode ser viável por meio de **parcerias com instituições de acessibilidade**, crowdfunding ou incentivos governamentais.

**Custo-benefício positivo** – A solução pode oferecer um valor agregado significativo para usuários cegos, tornando-se uma ferramenta de apoio indispensável.

**Desafios econômicos:** Investimentos iniciais podem ser necessários para pesquisa de materiais, testes de usabilidade e certificação do produto para comercialização.

## Conclusão

A **camisa sensorial** apresenta **alta viabilidade técnica e econômica**, sendo um projeto acessível, funcional e escalável. Com ajustes na fabricação e otimização do código, a implementação pode **reduzir**

**custos e ampliar seu impacto**, tornando-se uma ferramenta valiosa para a inclusão social de pessoas cegas.