AI & SUSTAINABILITY IN VET EDUCATION

Nome do(s) autor(es): Sucessos Criativos, Lda

Título do WP: WP3 – IA-SAVES@ProjectHub

Referência de Atividade: 3.2 – Modelo de projeto para IA-SAVES@ProjectHub

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Índice

1. Título do Projeto: 4
2. Categoria do Projeto: 4
3. Descrição do Projeto: 4
4. Palavras-chave: 4
5. Público-alvo: 4
6. Objetivos do Projeto: 4
7. Cronologia do Projeto: 5
8. Recursos Fornecidos: 5
9. Ferramentas e Tecnologias Utilizadas: 6
10. Plano de Implementação Passo a Passo: 7
Passo 1: Prepare o Raspberry Pi para Interação de Hardware 7
Passo 2: Ligue o Sensor de LDR 7
Passo 3: Escreve o programa Python para Detetar Luz 8
11. Critérios de Avaliação: 12
12. Línguas Disponíveis: 12
13. Colaboradores: 12
17. Feedback e Melhoria Contínua: 13

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

1. Título do Projeto:

Primeiros Passos com o Kit AI-VET: MONITORIZAÇÃO DE DISTÂNCIAS E SISTEMA DE FEEDBACK VISUAL EFICIENTE EM ENERGIA

---

2. Categoria do Projeto:

Fundamentos da Microeletrónica

---

3. Descrição do Projeto:

Este projeto integra três elementos de hardware — um sensor de distância, um ecrã OLED e uma fita NeoPixel — num sistema inteligente e eficiente em termos energéticos. O sensor ultrassónico mede a distância a um objeto, o ecrã OLED mostra a leitura em tempo real e a faixa NeoPixel responde com feedback visual codificado por cores. Este sistema de feedback em tempo real apoia a sustentabilidade ao otimizar o consumo de energia através de hardware mínimo e comunicação clara. É ideal para salas inteligentes, sistemas de deteção de objetos ou alertas visuais que priorizam baixo consumo de energia.

---

4. Palavras-chave:

Raspberry Pi, Python, Sensor Ultrassónico, Ecrã OLED, NeoPixel Strip, SPI, I^2C, Monitorização de Distância, Feedback Visual, Sustentabilidade

---

5. Público-alvo:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Educadores, aprendizes e criadores, com ou sem experiência prévia, que estão a iniciar o seu percurso na microeletrónica e computação física.

---

6. Objetivos do Projeto:

- Sensor ultrassónico de interface, ecrã OLED e tira NeoPixel com Raspberry Pi 5
- Mede e mostra distâncias usando feedback em tempo real
- Utilizar a comunicação SPI e I^2C de forma eficiente
- Promover um design energeticamente eficiente com uma pegada mínima de hardware
- Pratique a integração de múltiplos sensores num único sistema controlado por Python

---

7. Cronologia do Projeto:

A duração estimada é de aproximadamente **2,5 a 3 horas**, incluindo a montagem do ambiente e a montagem física.

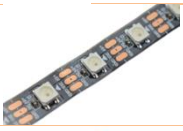
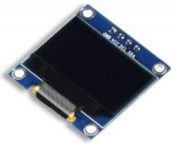
---

8. Recursos Fornecidos:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- Kit AI-VET (Raspberry Pi 5, LED, Breadboard, Jumpers, Ecrã OLED, Sensor Ultrassónico, Fita LED)

MATERIAL	
Raspberry Pi 5	
Fonte de alimentação USB Tipo-C de 27W para Raspberry Pi 5	
Cabos Flat Jumper F/F	
Breadboard	
FIO EM FILA ÚNICA PARA PLACAS DE ENSAIO	
WS2812 Fita LED Programável RGB 15x LED	
Ecrã OLED i2c de 0,95 polegadas Azul/Amarelo SSD1306	
HC-SR04P - SENSOR ULTRASSÓNICO 3V~5V	

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

9. Ferramentas e Tecnologias Utilizadas:

- Raspberry Pi OS
- Python 3
- Geany IDE
- Terminal (bash)
- Bibliotecas: `lgpio`, `adafruit-circuitpython-ssd1306`, `adafruit-circuitpython-neopixel-spi`, `Pillow`, `placa`

---

10. Plano de Implementação Passo a Passo:

Passo 1: Prepare o Raspberry Pi para Interação de Hardware

Este projeto baseia-se diretamente em experiências anteriores em kits de IA-VET, incluindo:

- **"Começar com o Raspberry Pi 5: Configuração, Configuração e Hello World em Python"**
- **"Medição de Distância com um Sensor Ultrassónico"**
- **"A trabalhar com o ecrã OLED"**
- **"Controlar a NeoPixel Strip"**

Certifica-te de que já completaste esses procedimentos para uma configuração mais suave.

Neste passo, irá:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- Instale o Raspberry Pi OS através do Raspberry Pi Imager
- Configure definições do sistema como Wi-Fi e fuso horário
- Ative as seguintes interfaces usando `raspi-config`:
    - **I²C** para o ecrã OLED
    - **SPI** para a faixa NeoPixel
- Identifique o IP do Raspberry Pi usando `ifconfig`
- Ligue-se remotamente via TigerVNC, se necessário
- Instalar bibliotecas obrigatórias:

```
pip3 install adafruit-circuitpython-ssd1306

pip3 install adafruit-circuitpython-neopixel-spi

Sudo apt install python3-rpi-lgpio
```

Passo 2: Construir o Circuito

Neste passo, irá ligar três módulos-chave — **Sensor Ultrassónico**, **Ecrã OLED** e **NeoPixel Strip** — ao Raspberry Pi usando diferentes protocolos de comunicação: GPIO, I²C e SPI. Cada módulo desempenha um papel único no sistema e está ligado a pinos específicos do Raspberry Pi para garantir comunicação e funcionalidade adequadas.

Ecrã OLED (Saída Visual – Comunicação I²C)

O ecrã OLED recebe texto em direto do Raspberry Pi e mostra-o em tempo real. Utiliza o protocolo **I²C (Inter-Integrated Circuit),** permitindo que múltiplos dispositivos partilhem as mesmas duas linhas de dados, o que simplifica a cablagem e suporta a transmissão de dados energeticamente eficiente.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Ligações:

- **VCC5V** → Raspberry Pi (pino 2)
  Fornece energia ao módulo de ecrã OLED

- **GNDGND** → Raspberry Pi (pino 6)
  Liga a terra do ecrã à terra do sistema

- **SDA (Linha de Dados Serial) GPIO 2** → (pino 3)
  Transmite dados do Raspberry Pi para o ecrã OLED

- **SCL (Linha de Relógio Serial) GPIO 3** → (pino 5)
  Envia sinais de temporização que coordenam a comunicação I^2C

Estas quatro ligações são tudo o que precisas para comunicar com o ecrã OLED. Uma vez ligado, o Raspberry Pi envia dados de píxeis por I^2C para controlar a saída do ecrã.

Fita LED NeoPixel (Feedback Visual – Comunicação SPI)

A fita NeoPixel fornece feedback codificado por cores usando SPI (Serial Peripheral Interface), que proporciona uma comunicação fiável e rápida para efeitos de iluminação.

Ligações:

- **DIN (Data In)10** → GPIO (MOSI) (pino 19)
  Transfere dados de cor do LED do Raspberry Pi

- **VCC5V** → Raspberry Pi (pino 4)
  Alimenta a fita NeoPixel

- **GNDGND** → Raspberry Pi (pino 14)
  Partilha um terreno comum para sinalização precisa

Sensor Ultrassónico (Medição de Distância – Controlo GPIO)

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

O sensor ultrassónico mede a distância usando o tempo de eco. Requer dois pinos GPIO para transmissão digital de sinal.

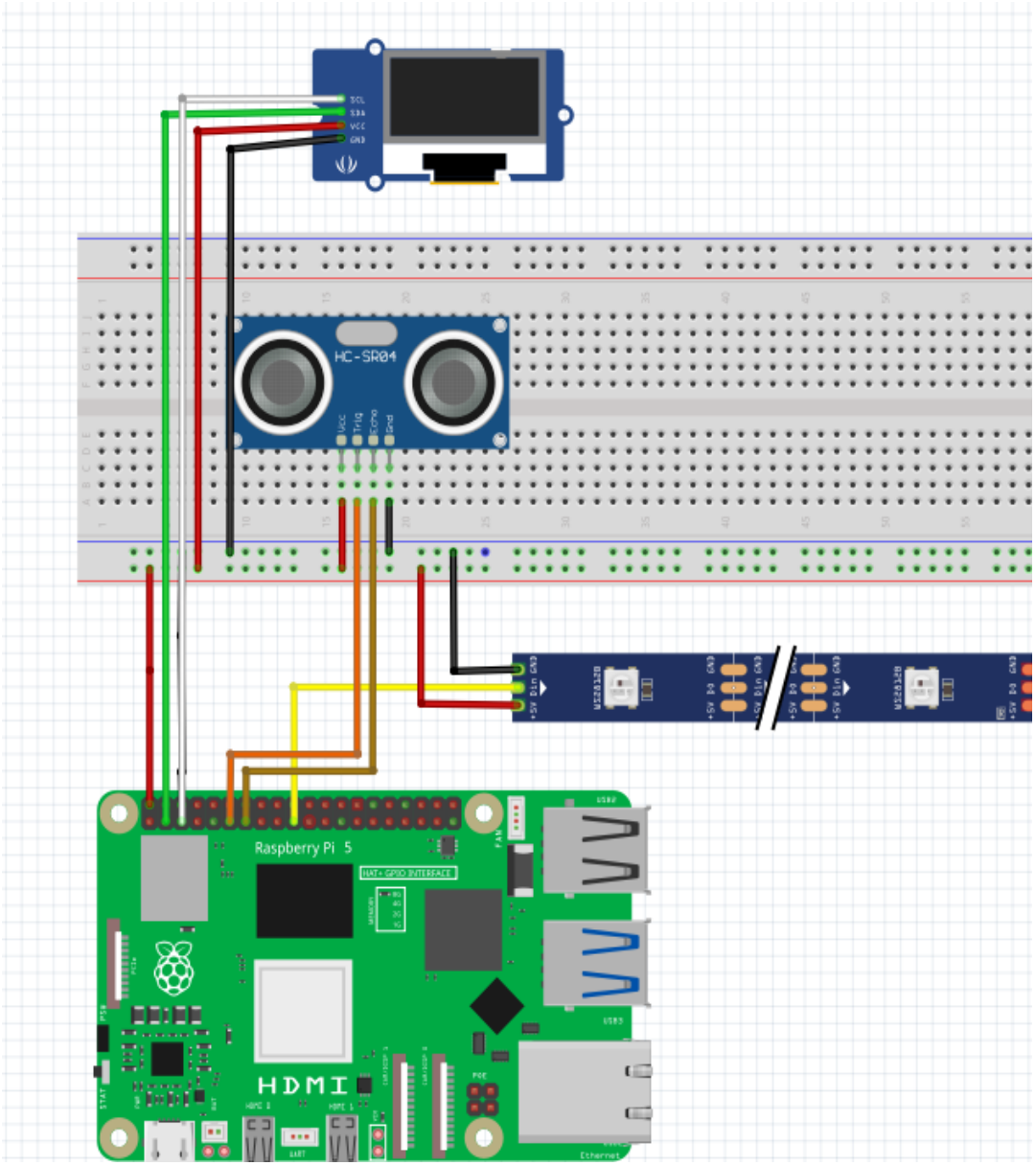
Ligações:

- **VCC5V** → Raspberry Pi (pino 1)
  Fornece energia ao sensor
- **GNDGND** → Raspberry Pi (pino 9)
  Estabelece terreno comum com o Pi
- **TRIG17** → GPIO (pino 11)
  Envia o pulso ultrassónico
- **ECHO27** → GPIO (pino 13)
  Recebe o eco de retorno para calcular a distância

Em conjunto, estas ligações permitem ao Raspberry Pi recolher dados de distância, mostrar valores no ecrã OLED e desencadear alterações de cor na fita NeoPixel — formando um sistema de feedback visual completo e energeticamente eficiente.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.



Passo 3: Escreva o programa Python para controlar a janela

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Abra Geany e introduza o script seguinte. Este código inicializa os três componentes e liga-os através de comportamentos sincronizados:

```
Tempo de importação
Tabuleiro de importação
Import Digitalio
a partir de PIL importar Image, ImageDraw, ImageFont
montante adafruit_ssd1306
Importação da LGPIO
importar neopixel_spi como Neopixel
# Defina pinos GPIO para o sensor ultrassónico
TRIGGER_PIN = pino GPIO 17  # para o gatilho do sensor ultrassónico
ECHO_PIN = pino GPIO 27     # para o eco do sensor ultrassónico
# Defina as dimensões do ecrã OLED
LARGURA = 128   # largura OLED em pixels
ALTURA = 64   # Altura OLED em pixels
# Número de LEDs na faixa NeoPixel
NUM_PIXELS = 12  # Número de LEDs NeoPixel na fita
# Defina a ordem das cores dos NeoPixels (Verde, Vermelho, Azul)
PIXEL_ORDER = neopixel. GRB # Ordem de cor GRB para NeoPixel
# Defina três cores (Vermelho, Verde, Azul) em formato hexadecimal
CORES = (0xFF0000, 0x00FF00, 0x0000FF) # Cores Vermelho, Verde, Azul
# Inicializar o chip GPIO (o chip GPIO padrão é "/dev/gpiochip0")
h = lgpio.gpiochip_open(0)
lgpio.gpio_claim_output(h, TRIGGER_PIN)  # Definir TRIGGER_PIN como saída
lgpio.gpio_claim_input(h, ECHO_PIN)      # Definir ECHO_PIN como entrada
# Inicializar o ecrã OLED usando comunicação I2C
i2c = tabuleiro. I2C()
OLED = adafruit_ssd1306. SSD1306_I2C(LARGURA, ALTURA, i2c, addr=0x3C)
# Inicializar a fita NeoPixel com comunicação SPI
SPI = Conselho. SPI()
píxeis  =  neopíxel.  NeoPixel_SPI(spi,  NUM_PIXELS,  pixel_order=PIXEL_ORDER,
auto_write=Falso)
# Função para medir distâncias usando o sensor ultrassónico
def measure_distance():
    # Enviar um pulso de 10 microssegundos para o pino de gatilho
    lgpio.gpio_write(h, TRIGGER_PIN, 1)
    time.sleep(0.00001) # Espera 10 microssegundos
    lgpio.gpio_write(h, TRIGGER_PIN, 0)
    # Mede o tempo em que o sinal de eco começa (sobe alto)
    start_time = tempo.tempo()
    enquanto lgpio.gpio_read(h, ECHO_PIN) == 0:
        start_time = tempo.tempo()
    # Mede o tempo em que o sinal de eco termina (fica baixo)
    end_time = tempo.tempo()
    enquanto lgpio.gpio_read(h, ECHO_PIN) == 1:
        end_time = tempo.tempo()
    # Calcula a duração do pulso e converte-a em distância
    pulse_duration = end_time - start_time
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
 distância = (pulse_duration * 34300) / 2 # A velocidade do som é 34300 cm/s
     Distância de retorno
# Função para atualizar o ecrã OLED com a medição de distância
def update_display(distância):
    oled.fill(0) # Limpar o ecrã OLED
    oled.show()

    # Criar uma nova imagem para o ecrã OLED
 imagem = Image.new("1", (oled.width, oled.height))
    draw = ImageDraw.Draw(image)
    # Desenha um retângulo de borda para estética
    FRONTEIRA = 5
    draw.rectangle(
        (BORDA, BORDA, oled.largura - FRONTEIRA - 1, oled.altura - FRONTEIRA -
1),
        resumo=0,
        preencher=0,
    )
    # Carrega a fonte padrão e prepara o texto
    fonte = ImageFont.load_default()
    texto = f"Distância: {distância:.2f} cm"
    # Calcula o tamanho do texto para o centrar no ecrã
    font_width, font_height = font.getbbox(texto)[2:4]
    # Desenha o texto centrado no ecrã OLED
    draw.text(
        (oled.width // 2 - font_width // 2, oled.height // 2 - font_height //
2),
        texto,
        fonte=fonte,
        preenchido=255, # Cor de texto branco
    )
    # Enviar a imagem para o ecrã OLED
    oled.imagem(imagem)
    oled.show()
# Função para atualizar a faixa NeoPixel com base na distância
def update_neopixels(distância):
    Se a distância < 10:
        cor = COLORS[0] # Definir para Vermelho se a distância for inferior a
10 cm
    Distância ELIF < 20:
        cor = CORES[1] # Definido para Verde se a distância for entre 10 cm e
20 cm
    Caso mais:
        cor = CORES[2] # Definido para Azul se a distância for superior a 20 cm
    # Aplicar a cor selecionada a todos os LEDs da tira
    para i no intervalo(NUM_PIXELS):
        pixels[i] = cor
    pixels.mostrar()
# Loop principal para medir continuamente distâncias, atualizar o ecrã OLED e a
faixa NeoPixel
Experimente:
```

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

```
    embora Verdadeiro:
        # Mede a distância do sensor ultrassónico
        distance_cm = measure_distance()
        print(f'Distância: {distance_cm:.2f} cm')  # Distância de impressão até
à consola
        # Atualizar o ecrã OLED com a distância medida
        update_display(distance_cm)
        # Atualize a faixa NeoPixel com base na distância medida
        update_neopixels(distance_cm)
        # Atraso entre cada iteração de ciclo
        tempo.sono(0.3)
# Captura Ctrl+C para sair do programa com elegância
exceto o KeyboardInterrupt:
    print("\nA sair do programa.")
# Limpar os recursos do GPIO e encerrar o programa
Finalmente:
    lgpio.gpiochip_close(h)
    print("GPIO limpo!")
```

Guarde o ficheiro como: NeoDistOled.py

Passo 4: Análise do Código

Configuração e Importações:

- Board, `Digitalio` e `LGPIO` para interação de hardware
- `Almofada` e `adafruit_ssd1306` para controlo OLED
- `neopixel_spi` para iluminação de fitas LED
- Tempo para atrasos de loop

Inicialização do Sensor e do Ecrã:

- Os pinos GPIO 17 e 27 são reivindicados para leitura ultrassónica
- Os objetos OLED e NeoPixel são criados usando protocolos I^2C e SPI

Funções Principais:

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- `measure_distance()` utiliza TRIG e ECHO para calcular a distância em cm
- `update_display()` traça a distância no OLED
- `update_neopixels()` define as cores dos LEDs com base nos limiares de distância:
    - <10 cm → Vermelho
    - 10–20 cm → Verde
    - 20 cm → Azul

Circuito Principal:

- Mede continuamente a distância, apresenta-a no ecrã e atualiza a faixa de LEDs
- Sai limpo com Ctrl + C, libertando recursos GPIO

Passo 5: Execute o programa e observe a saída

1. **Guardar** o ficheiro como `NeoDistOled.py`
2. **Abra** o **terminal** e execute:

```
Python3 NeoDistOled.py
```

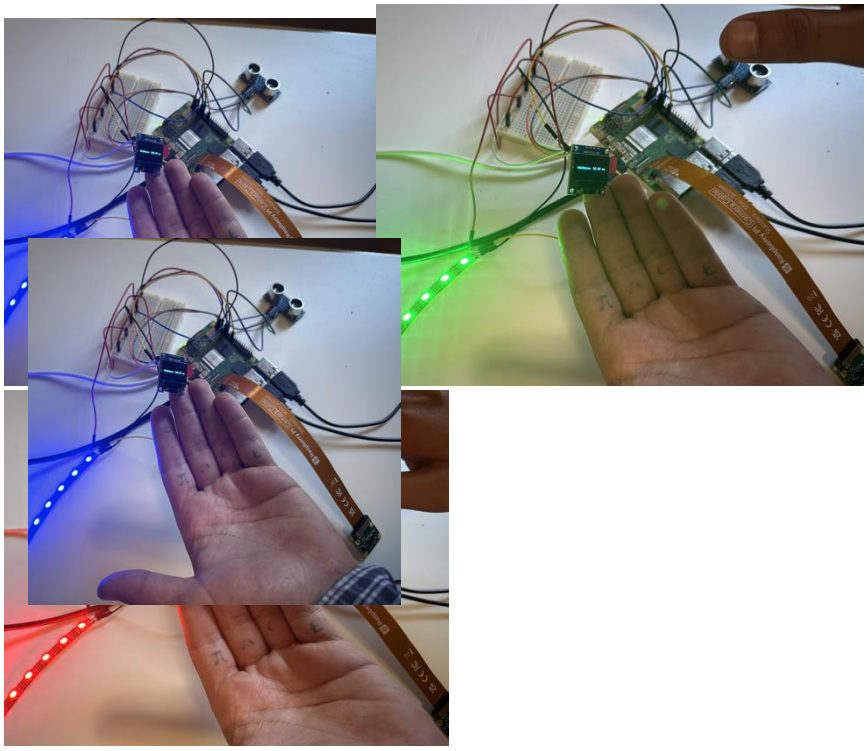
Deve observar:

- O ecrã OLED a mostrar distância em tempo real

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- A faixa NeoPixel ilumina-se em cores diferentes consoante a proximidade
- Impressão terminal das distâncias medidas



---

11. Critérios de Avaliação:

- **Precisão do Sensor:** O sensor ultrassónico mede corretamente distâncias dentro de uma margem de ±1 cm
- **Feedback Visual:** As cores dos NeoPixel mudam com precisão dependendo do limiar de distância

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

- **Ecrã OLED:** As leituras em tempo real são visíveis, legíveis e atualizadas em tempo real

- **Estabilidade do Código:** O script corre de forma fluida e sem erros, lida com interrupções de forma elegante

- **Competências de Integração:** Os três componentes trabalham em conjunto de forma síncrona para fornecer feedback responsivo

---

12. Línguas Disponíveis:

Inglês, **Português** (traduções em progresso)

---

13. Colaboradores:

Fátima Pais, Luís Soares – Designers e Facilitadores de Projetos

---

17. Feedback e Melhoria Contínua:

- **Inquéritos**: No final deste marco, alunos e educadores irão preencher inquéritos de feedback para avaliar a eficácia do projeto.
- **Revisões de Projetos**: Após a conclusão do projeto, será realizada uma reunião final de revisão para discutir potenciais melhorias para futuros projetos baseados em IA.
- **Análise de Dados**: Os dados de feedback serão analisados para identificar pontos fortes e áreas a melhorar, informando projetos futuros.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.

Co-funded by the European Union

Este projeto foi financiado com o apoio da Comissão Europeia. Esta comunicação reflete apenas as opiniões do autor, e a Comissão não pode ser responsabilizada por qualquer uso que possa ser feito da informação nela contida.