Detector de Motor Ocioso

O projeto consiste em um sistema capaz de detectar a ociosidade e o consumo de combustível de um motor por meio de um sensor analógico. Caso o motor permaneça parado por 10 minutos (10 segundos para simular), os buzzers emitirão um alerta sonoro. O sistema também conta com um LED RGB e um display OLED para indicar diferentes estados do motor, dados processados pelo programa, e um botão para ligar e desligar o motor.

OBJETIVOS

O principal objetivo do projeto é criar um sistema inteligente que monitore a atividade do motor em tempo real, identificando quando o motor entra em um estado de ociosidade prolongada. A detecção dessa inatividade é essencial, pois motores que permanecem ligados e inativos consomem combustível sem necessidade, contribuindo para o desperdício e aumento dos custos operacionais.

Ao alertar sobre a ociosidade prolongada por meio de sinais visuais (LED RGB) e sonoros (buzzer), o sistema oferece uma resposta imediata ao usuário, incentivando a tomada de ações corretivas, como desligar o motor quando não estiver em uso.

Além disso, o sistema contribui para a prevenção de desgaste desnecessário do motor. Motores que permanecem em funcionamento sem necessidade podem sofrer maior desgaste mecânico, reduzindo sua vida útil. Portanto, ao detectar e alertar para períodos de ociosidade, o sistema ajuda a prolongar a durabilidade do motor e reduzir custos com manutenção.

Dessa forma, o projeto visa não só a economia de combustível, mas também a eficiência operacional e a sustentabilidade no uso de máquinas e veículos, alinhando o desempenho com práticas mais conscientes e eficientes.

DESCRIÇÃO DO FUNCIONAMENTO

- **Sensor Analógico**: Mede o movimento do motor. Se o motor estiver em movimento, o sensor gera valores altos; se o motor estiver parado, o sensor gera valores baixos. Com base nisso, o sistema determina o estado do motor.
- **LED RGB**:
    - Verde: Quando o motor está ligado e em movimento.

    - Amarelo: Quando o motor está ligado, mas parado (detectado pelo sensor).
    - Vermelho: Quando o motor está ocioso por mais de 10 segundos.
    - Desligado: Quando o motor está completamente desligado.
- **Buzzer**: Emite um som de alerta quando o motor fica ocioso por mais de 10 segundos, chamando a atenção do usuário para o desperdício de combustível.
- **Botão A**: Controla o estado do motor, alternando entre ligado e desligado. Pressionando o botão A, o motor liga. Pressionando novamente, o motor desliga.
- **Botão B**: Inicia uma nova viagem e envia para o registro os dados da viagem finalizada. Resetando os demais contadores.
- **Display OLED**: Exibe informações em tempo real sobre o estado do motor (ligado ou desligado), tempo de ociosidade e o número de viagens realizadas. Também pode mostrar o consumo de combustível estimado.

JUSTIFICATIVA

A justificativa principal para este projeto é reduzir o desperdício de combustível e minimizar o impacto ambiental associado à queima desnecessária de combustíveis fósseis. Motores que permanecem ociosos por longos períodos continuam consumindo combustível e liberando emissões de gases poluentes, como dióxido de carbono (CO_2) e monóxido de carbono (CO), que contribuem para o aquecimento global e a poluição do ar. Ao alertar o usuário sobre a ociosidade do motor, o projeto incentiva práticas mais sustentáveis de uso de veículos e máquinas, ajudando a reduzir o consumo desnecessário de combustível e diminuindo a emissão de poluentes.

Além dos benefícios ambientais, o uso de logs de viagens gerados pelo sistema pode ser extremamente útil para empresas que utilizam uma frota de veículos ou equipamentos com motores. Ao registrar dados como o tempo de inatividade, o consumo de combustível e o número de viagens realizadas, o sistema oferece informações valiosas para o monitoramento de eficiência e a gestão de recursos. Esses logs permitem que a empresa faça um acompanhamento detalhado de cada motor, avaliando seu desempenho e identificando padrões de uso ou períodos de ociosidade excessiva.

A análise desses dados pode ajudar na otimização da operação, permitindo que a empresa tome decisões informadas sobre a manutenção de seus veículos e máquinas, evitando falhas inesperadas e prolongando a vida útil dos motores. Além disso, a empresa pode identificar oportunidades para reduzir

custos operacionais ao diminuir o tempo de inatividade dos motores, contribuindo para uma maior eficiência financeira e sustentabilidade.

Dessa forma, a implementação do sistema não apenas melhora a qualidade do ar e reduz os impactos ambientais, mas também oferece uma ferramenta prática e estratégica para as empresas gerenciarem suas frotas de maneira mais eficiente, promovendo uma operação mais econômica e responsável.

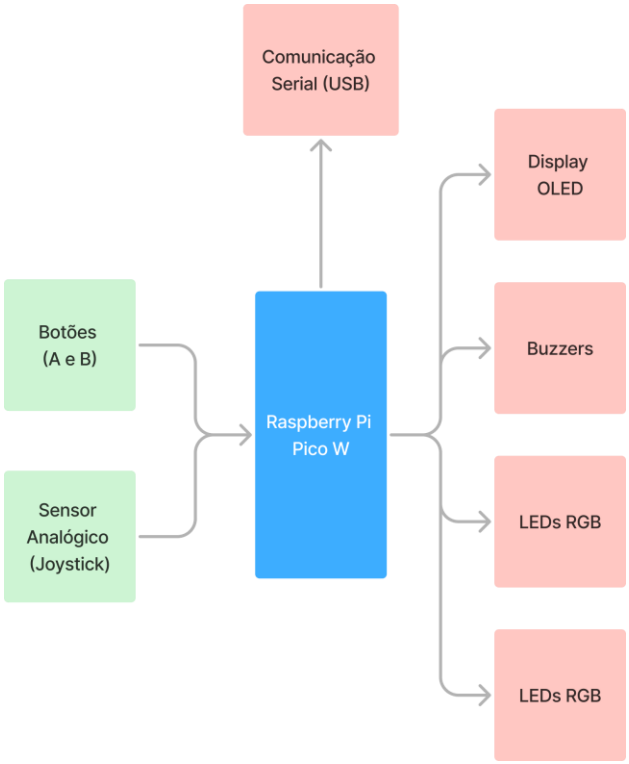
ORIGINALIDADE

Embora existam dispositivos de telemetria feitos por empresas que monitoram o desempenho e as condições dos motores, a proposta deste projeto se destaca pela simplicidade e dinamismo na implementação. Muitos sistemas de telemetria existentes são focados em registrar dados e transmiti-los para análise posterior, sem fornecer feedback imediato ao usuário. Isso pode resultar em uma resposta mais lenta a problemas como a ociosidade do motor, o que pode levar a um desperdício contínuo de combustível e maior impacto ambiental.

Neste projeto, a utilização de um display OLED permite que o usuário visualize, em tempo real, o estado do motor, incluindo informações sobre o consumo e tempo de inatividade. Isso torna o processo muito mais dinâmico, pois as informações são imediatamente acessíveis, possibilitando uma resposta rápida e eficaz.

Além disso, a emissão de um alerta sonoro via buzzer garante que o usuário seja notificado de maneira instantânea sobre a ociosidade do motor, evitando que o problema passe despercebido. Em dispositivos de telemetria mais tradicionais, essa comunicação pode ser feita de forma menos eficiente, sem proporcionar a mesma imediata interação com o usuário. A combinação de um display visual e uma resposta sonora imediata torna o sistema proposto não apenas mais acessível, mas também mais eficaz em promover práticas sustentáveis de uso do motor.

DIAGRAMA EM BLOCO



FUNÇÃO DE CADA BLOCO

1. **Sensor Analógico (Joystick)**:
    - Mede o movimento do motor se baseando nos eixos X e Y.
    - Gera valores analógicos que são convertidos para determinar se o motor está em movimento ou parado a partir do limiar definido (1940 a 2200).
2. **Botões (A e B)**:
    - **Botão A**: Liga e desliga o motor.
    - **Botão B**: Reseta os contadores de ociosidade e consumo, e inicia uma nova viagem.
3. **Microcontrolador (Raspberry Pi Pico)**:
    - Processa os dados dos sensores e controla os periféricos (LEDs, buzzer, display).
    - Gerencia o estado do motor e executa a lógica do sistema.
4. **Display OLED**:
    - Exibe informações em tempo real sobre o estado do motor, consumo de combustível, tempo ocioso e número de viagens.

5. **Buzzer (A e B)**:
    - Emite alertas sonoros quando o motor está ocioso por mais de 10 segundos.
6. **LEDs RGB**:
    - Indica o estado do motor:
        - Verde: Motor ligado e em movimento.
        - Amarelo: Motor ligado, mas parado.
        - Vermelho: Motor ocioso por mais de 10 segundos.
7. **Comunicação Serial (USB)**:
    - Envia logs de viagens e consumo de combustível para o PC.
8. **Alimentação**:
    - Fornece energia para todos os componentes do sistema (3.3V ou 5V).

CONFIGURAÇÃO DE CADA BLOCO

1. **Sensor Analógico**:
    - Conectado aos pinos ADC do Raspberry Pi Pico (GPIO 26 e 27).
    - Configurado para leitura contínua dos valores analógicos.
2. **Botões A e B**:
    - Configurados como entradas com resistores de pull-up internos (GPIO 5 e 6).
    - Usam interrupções para detectar pressionamentos.
3. **Microcontrolador**:
    - Configura GPIOs, ADC, PWM e I2C.
    - Gerencia o estado do motor e controla os periféricos.
4. **Display OLED**:
    - Conectado via I2C (pinos GPIO 14 e 15).
    - Configurado para exibir texto e gráficos.
5. **Buzzer**:
    - Controlado por PWM para gerar tons de alerta.

    - Configurado nos pinos GPIO 10 e 21.
6. **LEDs RGB**:
    - Configurados como saídas digitais (GPIO 11, 12 e 13).
    - Controlados para indicar o estado do motor.
7. **Comunicação Serial**:
    - Configurada via USB para envio de logs.

COMANDOS E REGISTROS UTILIZADOS

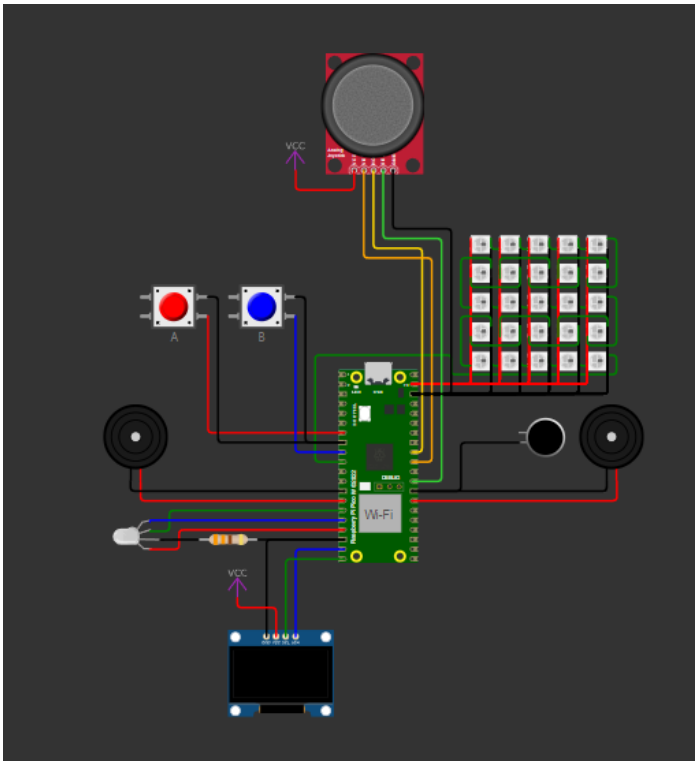
- **GPIO**:
    - gpio_init(), gpio_set_dir(), gpio_put(), gpio_get(): Configuração e controle dos pinos.
- **ADC**:
    - adc_init(), adc_gpio_init(), adc_select_input(), adc_read(): Leitura dos valores analógicos.
- **PWM**:
    - pwm_gpio_to_slice_num(), pwm_set_wrap(), pwm_set_gpio_level(), pwm_set_enabled(): Controle do buzzer.
- **I2C**:
    - i2c_init(), i2c_write_blocking(), i2c_read_blocking(): Comunicação com o display OLED.
- **Interrupções**:
    - gpio_set_irq_enabled_with_callback(): Configura interrupções para os botões.

DESCRIÇÃO DA PINAGEM USADA

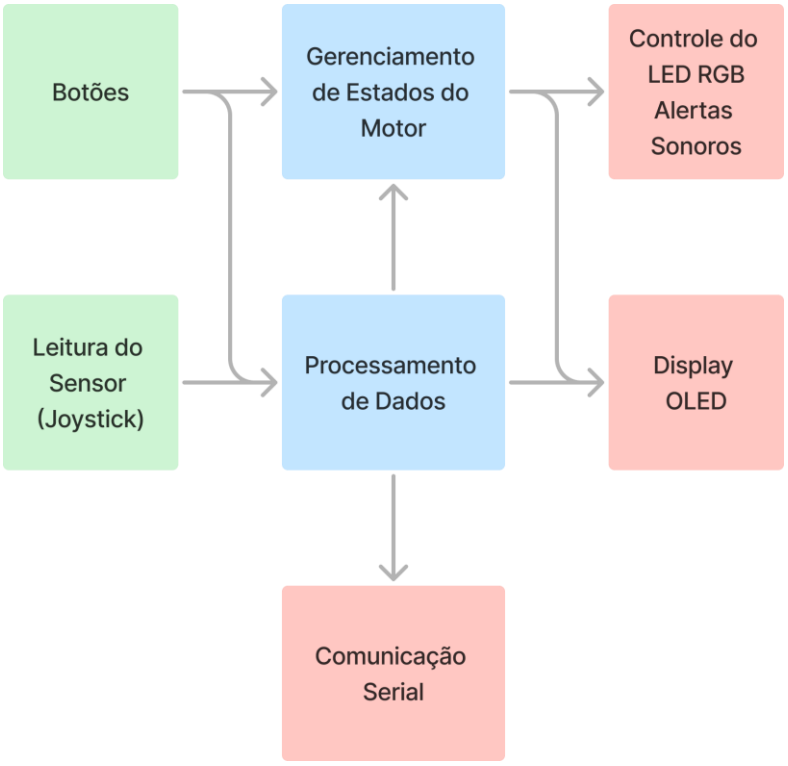
- **GPIO 5**: Botão A – Liga/desliga o motor.
- **GPIO 6**: Botão B – Reseta o contador de viagens.
- **GPIO 10**: Buzzer A – Emite alertas sonoros (controlado por PWM).
- **GPIO 11**: LED Verde – Indica motor em movimento.
- **GPIO 12**: LED Azul – Não utilizado.
- **GPIO 13**: LED Vermelho – Indica motor parado ou ocioso.

- **GPIO 14**: I2C SDA – Comunicação com o display OLED.
- **GPIO 15**: I2C SCL – Comunicação com o display OLED.
- **GPIO 21**: Buzzer B – Emite alertas sonoros (controlado por PWM).
- **GPIO 22**: Botão do Joystick – Detecta pressionamento.
- **GPIO 26**: Sensor Analógico (Eixo Y) – Mede movimento no eixo Y.
- **GPIO 27**: Sensor Analógico (Eixo X) – Mede movimento no eixo X.

CIRCUITO COMPLETO DO HARDWARE



BLOCOS FUNCIONAIS



DESCRIÇÃO DAS FUNCIONALIDADES

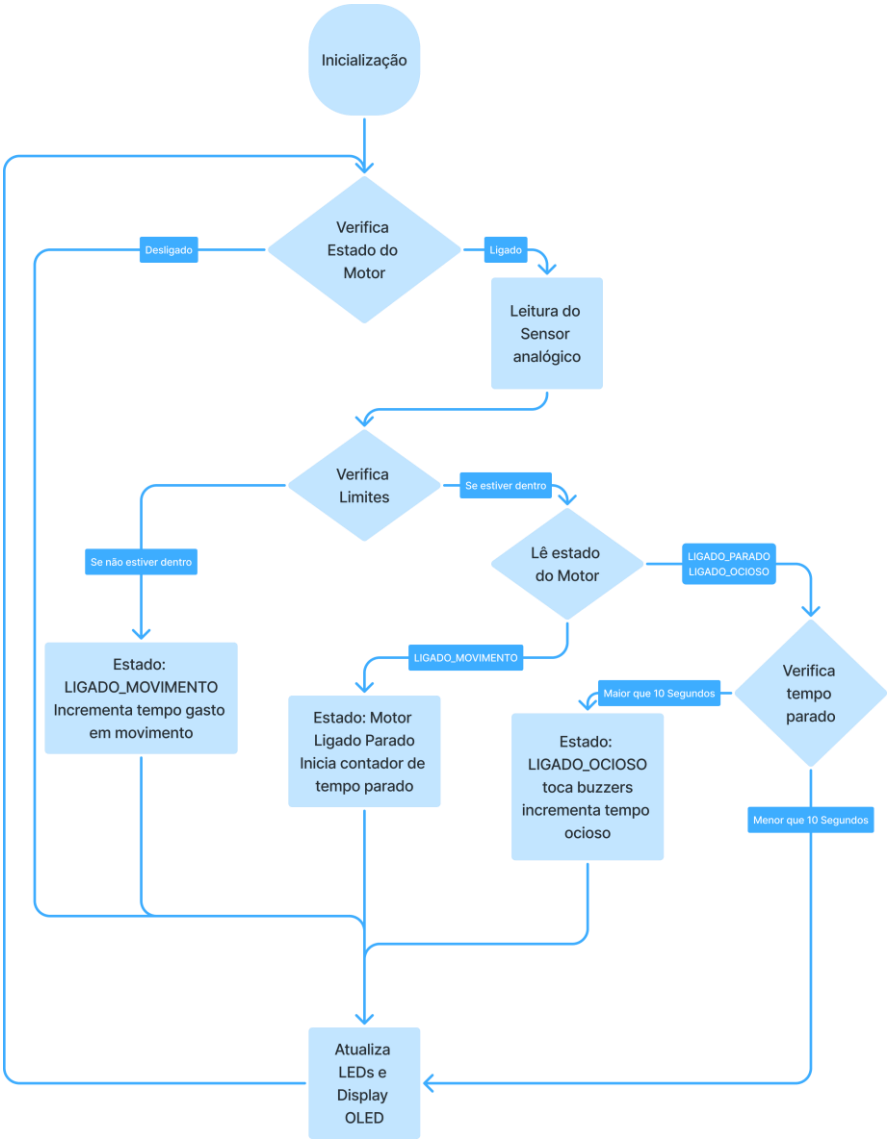
- **Leitura do Sensor**:
  - Captura valores analógicos do joystick (eixos X e Y) para determinar o estado do motor.
- **Controle do LED RGB**:
  - Atualiza os LEDs conforme o estado do motor (verde: movimento; vermelho: parado/ocioso).
- **Alertas Sonoros**:
  - Emite sons pelo buzzer quando o motor está ocioso por mais de 10 segundos.
- **Display OLED**:
  - Exibe informações em tempo real sobre o estado do motor, consumo de combustível e tempo ocioso.
- **Botões**:
  - Botão A: Liga/desliga o motor.
  - Botão B: Reseta o contador de viagens.

- **Comunicação Serial**:
    - Envia logs de viagens e consumo de combustível para um PC.

DEFINIÇÃO DAS VARIÁVEIS

- **estado_motor**: Armazena o estado atual do motor (desligado, ligado/movimento, ligado/parado, ligado/ocioso).
- **tempo_inicio_parado**: Registra o tempo em que o motor ficou parado.
- **gasto_tempo**: Contabiliza o tempo de movimento do motor.
- **viagem**: Contador de viagens realizadas.
- **tempo_ocioso**: Armazena o tempo total de ociosidade do motor.
- **leitura_x**, **leitura_y**: Valores analógicos lidos do joystick (eixos X e Y).

FLUXOGRAMA



INICIALIZAÇÃO

- Configura GPIOs, ADC, PWM e I2C.
- Define estados iniciais do motor (desligado).
- Inicializa o display OLED e limpa a tela.
- Configura interrupções para os botões.

CONFIGURAÇÃO DOS REGISTROS

- **GPIO**:
    - Configura pinos como entrada (botões) ou saída (LEDs, buzzer).

    - Habilita resistores de pull-up para os botões.
- **ADC**:
    - Configura os pinos 26 e 27 para leitura analógica.
    - Define o canal de leitura (0 para eixo X, 1 para eixo Y).
- **PWM**:
    - Configura os pinos 10 e 21 para controle do buzzer.
    - Define frequência e duty cycle para gerar tons.
- **I2C**:
    - Configura os pinos 14 (SDA) e 15 (SCL) para comunicação com o display OLED.
    - Define a frequência de comunicação (400 kHz).

ESTRUTURA E FORMATO DE DADOS

- **Valores Analógicos**:
    - Lidos do ADC (0 a 4095).
    - Usados para determinar movimento ou parada do motor.
- **Dados do Display**:
    - Strings formatadas para exibir estado do motor, consumo e tempo ocioso.
- **Logs de Viagens**:
    - Mensagens enviadas via comunicação serial:
        - Formato: "Viagem X: Consumo = Y L Tempo ocioso = Z min".
- **Comandos PWM**:
    - Frequência e duty cycle para controle do buzzer.

PROTOCOLO DE COMUNICAÇÃO

- Utiliza comunicação serial via USB para logs.

FORMATO DO PACOTE DE DADOS

- Mensagens enviadas via printf() para o PC:
    - "Viagem X: Consumo = Y L Tempo ocioso = Z min"

METODOLOGIA

A execução seguiu as etapas abaixo:

1. **Pesquisas realizadas**:
    - Pesquisas sobre telemetria em ônibus.
    - Estudo dos sensores analógicos.
    - Análise para integração com display OLED.
2. **Escolha do hardware**:
    - Raspberry Pi Pico.
    - Joystick analógico para simular movimento do motor.
    - Display OLED para exibição de dados.
3. **Definição das funcionalidades do software**:
    - Lógica para detectar ociosidade com base em limiares analógicos.
    - Controle de LEDs, buzzer e display OLED.
4. **Inicialização da IDE**:
    - Bibliotecas para ADC, PWM e I2C.
5. **Depuração**:
    - Uso de *prints seriais* para monitorar valores do sensor.
    - Ajuste de limiares e correção de erros de configuração.

TESTES DE VALIDAÇÃO

- Teste de leitura do sensor analógico
    - Ajuste do limiar de ativação para o movimento.
- Teste da mudança de estados do motor
    - Troca para resetar timer de ociosidade
- Teste da resposta do buzzer ao estado de ociosidade.
    - Acionar o buzzer sem afetar o timer

DISCUSSSÃO DOS RESULTADOS

O sistema se mostrou eficaz na detecção do estado do motor e na emissão de alertas. Pequenas variações na leitura do sensor foram ajustadas com o uso de limiares bem definidos.

- **Eficiência do sistema**:
    - O sistema detectou estados de movimento e ociosidade com precisão, emitindo alertas visuais e sonoros conforme esperado.
    - O display OLED proporcionou monitoramento intuitivo em tempo real.
- **Confiabilidade**:
    - Testes comprovaram resposta imediata a mudanças de estado, com ajustes robustos para variações no sensor.
- **Limitações**:
    - O joystick foi usado apenas para simulação. Em aplicações reais, sensores de vibração seriam mais adequados.
    - A estimativa de consumo de combustível foi simplificada (baseada em tempo).
- **Conclusão**:
    - O projeto cumpriu seus objetivos, demonstrando viabilidade técnica e potencial para reduzir desperdício de combustível em cenários reais.

REFERÊNCIAS

1. Documentação Oficial do Raspberry Pi Pico
   RASPBERRY PI FOUNDATION. Raspberry Pi Pico Python SDK. [S. l.], 2024. Disponível em: https://datasheets.raspberrypi.com/pico/raspberry-pi-pico-python-sdk.pdf. Acesso em: [26 de fevereiro de 2025].
2. Repositório BitDogLab no GitHub
   BITDOGLAB. BitDogLab GitHub Repository. [S. l.], 2024. Disponível em: https://github.com/BitDogLab/BitDogLab. Acesso em: [26 de fevereiro de 2025].
3. Wokwi - Simulador Online
   WOKWI. Wokwi - Simulador Online para Projetos de Eletrônica e IoT. [S.

l.], 2024. Disponível em: https://wokwi.com/. Acesso em: [26 de fevereiro de 2025].

4. Artigo da Delta Global
   DELTA GLOBAL. Alertas Inteligentes em Rastreadores de Frota: Motor Ocioso e Outras Notificações. Blog Delta Global, [S. l.]. Disponível em: https://blog.deltaglobal.com.br/alertas-inteligentes-rastreadores-frotas/#:~:text =Motor%20Ocioso%3A%20Notifica%20quando%20o,em%20um%20n%C3%ADvel%20muito%20baixo. Acesso em: [26 de fevereiro de 2025].

REPOSITÓRIO

https://github.com/luizzrosario/projeto-U7

VÍDEO

https://youtu.be/Wm3yKWKy5ml