

# Soluções em Energias Renováveis e Sustentáveis

Sprint 02

1CCR

## Integrantes

Leticia Giordani RM: 564028

Lívia Dos Santos RM: 562967

Luize Martinês RM: 564016

Sarah Sayako RM: 563841

## Ideia principal:

Usar um dispositivo para monitorar o consumo de energia e identificar os momentos de maior uso. Com base nesses dados, criar um algoritmo que automatize tarefas (como carregar o carro ou ligar o ar-condicionado) de acordo com a rotina da pessoa e o clima. O sistema também economizará energia nos períodos de baixo uso, como durante o horário de trabalho.

- definir padrões de consumo com base em histórico, clima e rotina do usuário
- Sugerir melhor horário ou melhor dia para usar eletrodomésticos (máquina de lavar, secadora) com base na geração solar
- calcular o consumo de energia de cada atividade e criar um plano de uso otimizado para dias nublados ou situações de queda de energia
- monitorar a eficiência dos painéis solares para alertar sobre necessidade de limpeza ou manutenção

## Objetivo do Protótipo

Desenvolver uma simulação funcional utilizando a plataforma TinkerCad, em que cargas elétricas (representadas por LEDs) sejam acionadas conforme a disponibilidade de energia solar, priorizando os dispositivos mais importantes.

## Justificativa

O uso de energia solar tem crescido significativamente, mas a sua disponibilidade pode variar ao longo do dia. Um sistema que priorize automaticamente o acionamento de cargas conforme a energia gerada contribui para a sustentabilidade, otimizando o consumo e garantindo o funcionamento de equipamentos essenciais mesmo com pouca geração.

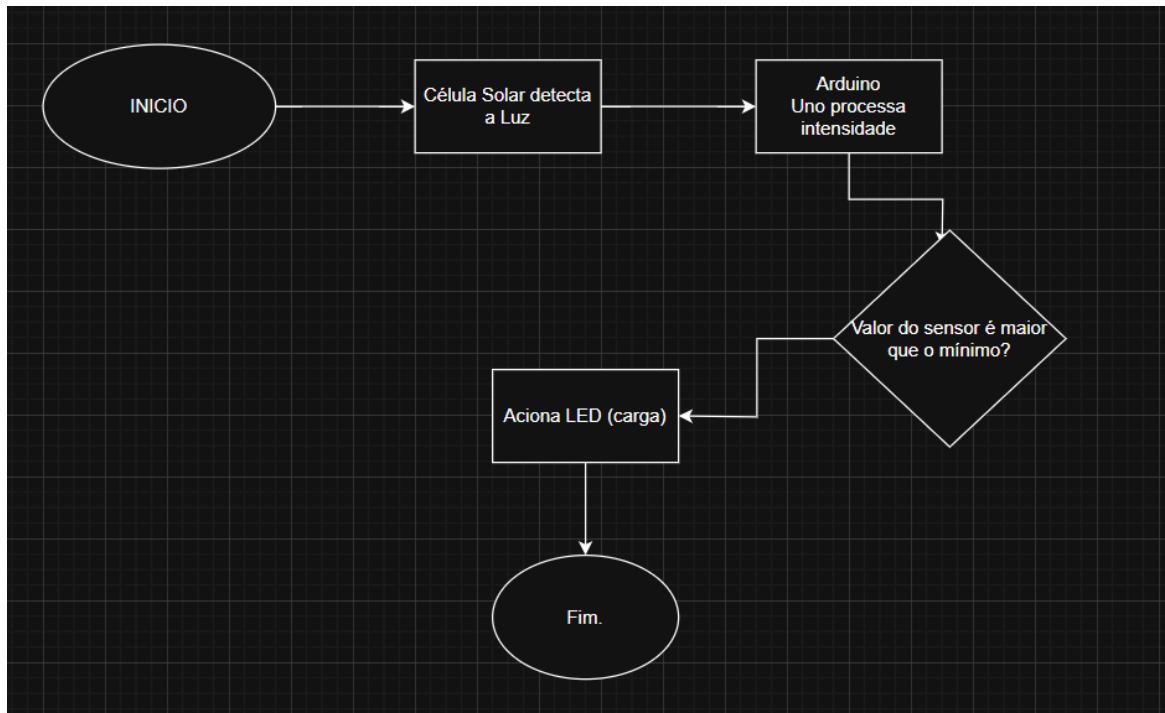
## Componentes Utilizados

- **Arduino Uno:** Microcontrolador responsável por ler a tensão simulada e controlar os LEDs.
- **Painéis Solares (simulados):** Fontes de tensão para representar a geração de energia.
- **Voltímetro:** Mostra a tensão gerada pelos painéis.
- **Sensor de tensão (A0):** Entrada analógica usada para ler a tensão.
- **LEDs (x4):** Representam cargas com diferentes níveis de prioridade.
- **Protoboard e resistores:** Conexão dos componentes.

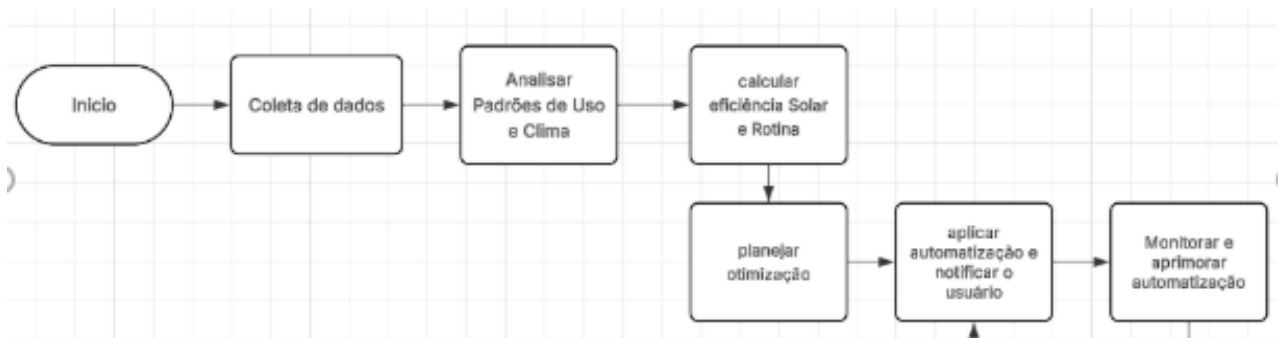
## Funcionamento do Sistema

A tensão gerada pelos painéis solares é lida pela entrada analógica A0 do Arduino. Com base nesse valor, o sistema ativa progressivamente os LEDs. Isso garante que, quando houver pouca energia, apenas as cargas mais essenciais sejam acionadas.

## Diagrama do Circuito



Este diagrama mostra a conexão dos painéis, protoboard, Arduino e LEDs. Ele representa claramente a lógica de priorização implementada no sistema.



## Considerações Finais

A prova de conceito apresentada demonstra, de forma simples e eficaz, como sistemas de energia solar podem ser otimizados com automação inteligente. Mesmo sendo uma simulação, o projeto evidencia a viabilidade técnica de soluções sustentáveis com uso de microcontroladores acessíveis e sensores de baixo custo.

## Repositório GitHub

<https://github.com/liviadantas11/Sprint2-EnergiaInteligente>

## Vídeo Demonstrativo

<https://youtu.be/udZc2jsmgb4>



## Código-Fonte Arduino

```
#define TENSAO A0
```

```
#define PRIORIDADE1 13
```

```
#define PRIORIDADE2 12
```

```
#define PRIORIDADE3 11
```

```
#define PRIORIDADE4 10
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(PRIORIDADE1, OUTPUT);
```

```
  pinMode(PRIORIDADE2, OUTPUT);
```

```
  pinMode(PRIORIDADE3, OUTPUT);
```

```
  pinMode(PRIORIDADE4, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
  int tensao = analogRead(TENSAO);
```

```
  Serial.println(tensao);
```

```
  if (tensao <= 255){
```

```
digitalWrite(PRIORIDADE1,HIGH);
```

```
  digitalWrite(PRIORIDADE2,LOW);
```

```
  digitalWrite(PRIORIDADE3,LOW);
```

```
  digitalWrite(PRIORIDADE4,LOW);
```

```
}
```

```
  else if(tensao <= 510){
```

```
digitalWrite(PRIORIDADE1,HIGH);
```

```
  digitalWrite(PRIORIDADE2,HIGH);
```

```
  digitalWrite(PRIORIDADE3,LOW);
```

```
  digitalWrite(PRIORIDADE4,LOW);
```

```
}  
else if(tensao <= 765){  
digitalWrite(PRIORIDADE1,HIGH);  
    digitalWrite(PRIORIDADE2,HIGH);  
    digitalWrite(PRIORIDADE3,HIGH);  
    digitalWrite(PRIORIDADE4,LOW);  
}  
else{  
digitalWrite(PRIORIDADE1,HIGH);  
    digitalWrite(PRIORIDADE2,HIGH);  
    digitalWrite(PRIORIDADE3,HIGH);  
    digitalWrite(PRIORIDADE4,HIGH);  
}  
  
delay(500);  
}
```