**Logotipo

Descrição gerada automaticamente com confiança média**

**Amanda Fragnan de Oliveira**

**André Pessoa Cunha**

**Gabrielle Halasc**

**Introdução**

A revolução energética está em curso, impulsionada pela crescente demanda por eletricidade e pela necessidade urgente de sustentabilidade. Os modelos tradicionais de consumo de energia estão se tornando obsoletos diante das novas exigências ambientais e econômicas. A Inteligência Artificial (IA) e a Internet das Coisas (IoT) emergem como forças transformadoras, oferecendo soluções inovadoras para otimizar o uso de energia. Em ambientes residenciais e urbanos, a integração de fontes renováveis, é essencial para um futuro sustentável. No entanto, desafios como variações nos padrões de consumo, a intermitência das fontes renováveis e a complexidade de prever comportamentos humanos exigem abordagens avançadas e adaptativas.

Nossa proposta apresenta uma plataforma de gestão energética de última geração, capaz de operar em tempo real e otimizar o consumo de energia de forma inteligente. Utilizando o banco de dados Oracle, garantimos eficiência, precisão e escalabilidade, permitindo uma análise robusta e uma resposta rápida às mudanças nas condições de consumo e produção de energia. Esta solução não só aborda os desafios atuais, mas também prepara o caminho para um futuro em que a IA desempenha um papel central na criação de um mercado de energia mais resiliente, eficiente e sustentável.

**Objetivos**

* **Revolucionar o Consumo de Energia:** Implementar decisões automatizadas e inteligentes para reduzir significativamente o consumo de energia, transformando a maneira como utilizamos os recursos energéticos.
* **Promover a Sustentabilidade:** Integrar métodos que permitam ao sistema reconhecer quando a luz ambiente externa é adequada, indicando que a energia solar é suficiente, evitando assim o consumo desnecessário de energia elétrica, para criar um sistema energético mais sustentável e resiliente.
* **Otimização em Tempo Real:** Utilizar dados em tempo real para ajustar e otimizar o uso de energia, minimizando desperdícios e maximizando a eficiência.

**Desafios e Barreiras**

1. **Coleta e Integração de Dados:** A obtenção de dados precisos e em tempo real de diversas fontes (sensores IoT, medidores inteligentes, etc.) é essencial, mas desafiadora devido à diversidade de dispositivos e protocolos. Superar essa barreira é crucial para uma gestão energética eficaz.
2. **Análise de Grandes Volumes de Dados: Processar** e analisar vastas quantidades de dados de consumo de energia requer algoritmos avançados e uma infraestrutura robusta. A capacidade de transformar esses dados em insights acionáveis é fundamental para a otimização energética.
3. **Tomada de Decisões em Tempo Real: Desenvolver** sistemas que possam tomar decisões rápidas e precisas com base em dados dinâmicos é vital para ajustar o consumo de energia de maneira eficiente e responsiva.
4. **Integração de Fontes Renováveis:** Gerenciar a intermitência das fontes de energia renováveis e integrá-las de forma eficiente ao sistema energético é um desafio complexo, mas necessário para um futuro sustentável.
5. **Segurança e Privacidade:** Garantir que os dados coletados e processados estejam protegidos contra acessos não autorizados e que a privacidade dos usuários seja mantida é imperativo para a confiança e a adoção generalizada da solução.

**Solução**

**Infraestrutura IoT Integrada com Oracle**

* **ESP32 e Sensores:**
  + Monitorar temperatura, movimento, luminosidade e consumo de dispositivos com sensores distribuídos.
* **Coleta de Dados:**
  + Simular a coleta de dados dos sensores do ESP32 e armazená-los em um arquivo Python.

**Sistema de Alternância Inteligente de Energia**

* **Desenvolvimento do Sistema:**
  + Criar um sistema em Python que utiliza as informações dos sensores para alternar inteligentemente entre a energia da rede e a gerada por painéis solares.
* **Algoritmo de Previsão:**
  + Implementar um algoritmo que antecipa a produção de energia solar em dias úmidos e carrega dispositivos de armazenamento antes de picos de consumo.

**Análise de Dados em R**

* **Coleta de Dados do Mercado de Energia:**
  + Utilizar dados oficiais do Governo Federal sobre consumo energético dos últimos anos, armazenados no banco Oracle.
* **Análises Exploratórias:**
  + Realizar análises exploratórias em R para identificar padrões de consumo de energia.

**Automação e Visualização**

* **Desenvolvimento da Interface:**
  + Criar uma interface de usuário em Python para visualização dos dados e relatórios de eficiência energética.
* **Monitoramento em Tempo Real:**
  + Monitorar, em tempo real, as tarifas de energia e o consumo interno da residência, permitindo a seleção automática da fonte de energia mais econômica e sustentável.

**Diagrama da Solução**

Interface gráfica do usuário

Descrição gerada automaticamente

**Desenvolvimento / Entregáveis**

* **Artificial Intelligence with Computer Systems and Sensors (AICSS)**

Dentro das disciplinas a serem exploradas, temos o tema de *Artificial Intelligence with Computer Systems and Sensors (AICSS)*. O objetivo desta seção é desenvolver um circuito com o ESP32 que otimize a iluminação interna e externa de uma residência.

Diagrama, Esquemático

Descrição gerada automaticamente

Conforme ilustrado na figura acima, o projeto foi desenvolvido com o ESP32 como componente central. Para facilitar o entendimento, o projeto foi dividido em duas partes: uma para o ambiente externo (lado direito) e outra para o ambiente interno (lado esquerdo). Foram adicionados dois sensores LDR para medição de luminosidade, um para cada ambiente, assim como dois LEDs que representam a iluminação de cada ambiente, respectivamente. Adicionalmente, foi incluído um sensor HC-SR04 no ambiente interno para medição de distâncias, utilizado como um sensor eficaz para detecção de presença.

O código foi desenvolvido para implementar a seguinte lógica:

1. **Ambiente Interno:** Caso haja boa luminosidade no ambiente interno, a luz não será acionada. Entretanto, ao escurecer, a luz será acesa somente se for detectado algum movimento. Para isso, o sensor HC-SR04 está configurado com um valor padrão de 400 cm. Caso seja detectada uma diminuição dessa distância para menos de 200 cm (quando alguém passar na frente do sensor, por exemplo), o sistema permitirá que a luz seja acesa, desde que a luminosidade identificada tenha um valor analógico maior que 100 (aproximadamente 2.000 LUX - a sensibilidade do sensor LDR pode ser ajustada conforme necessário).
2. **Ambiente Externo:** Para o ambiente externo, não utilizamos técnicas de detecção de movimento, pois a luz precisa estar sempre acesa quando a luminosidade estiver baixa, por questões de segurança, especialmente em locais que utilizam câmeras sem infravermelho. Neste caso, o sistema acende a luz quando a luminosidade identificada tiver um valor analógico maior que 100 (aproximadamente 2.000 LUX - a sensibilidade do sensor LDR pode ser ajustada conforme necessário).

**Código fonte**

**Projeto.ino**

int LDRSensor\_Interno = 33; // Pino do sensor de luminosidade interno

int LDRSensor\_Externo = 25; // Pino do sensor de luminosidade externo

int Luz\_Interna = 32; // Pino da luz interna

int Luz\_Externa = 5; // Pino da luz externa

int Trig\_pin = 17; // Pino TRIG do Sensor HC-SR04

int Echo\_pin = 18; // Pino ECHO do Sensor HC-SR04

// Variáveis para cálculo de distância do Sensor HC-SR04

long duration;

float Speed\_of\_sound = 0.034;

float dist\_in\_cm;

void setup() {

  Serial.begin(115200);

  pinMode(LDRSensor\_Interno, INPUT);

  pinMode(LDRSensor\_Externo, INPUT);

  pinMode(Luz\_Interna, OUTPUT);

  pinMode(Luz\_Externa, OUTPUT);

  pinMode(Trig\_pin, OUTPUT);

  pinMode(Echo\_pin, INPUT);

}

void loop() {

  // Leitura dos sensores de luminosidade

  int luminosidadeInterna = analogRead(LDRSensor\_Interno);

  Serial.println("LUX Analog Read Interno: " + String(luminosidadeInterna));

  int luminosidadeExterna = analogRead(LDRSensor\_Externo);

  Serial.println("LUX Analog Read Externo: " + String(luminosidadeExterna));

  // Leitura do sensor de distância HC-SR04

  digitalWrite(Trig\_pin, LOW);

  delayMicroseconds(2);

  digitalWrite(Trig\_pin, HIGH);

  delayMicroseconds(10);

  digitalWrite(Trig\_pin, LOW);

  duration = pulseIn(Echo\_pin, HIGH);

  dist\_in\_cm = duration \* Speed\_of\_sound / 2;

  Serial.println("Distancia em cm: " + String(dist\_in\_cm));

  //Tabela de Parametros do Sensor de Luminosidade

  //--------------------------------------------------------------------

  //Condição                  Nível de iluminância (lux)    analogRead()

  //Lua cheia                 0.1                           1016

  //Crepúsculo profundo       1                             985

  //Crepúsculo                10                            853

  //Monitor do computador\*\*   50                            633

  //Iluminação da escada      100                           511

  //Iluminação do escritório  400                           281

  //Dia nublado               1,000                         170

  //Luz do dia                10,000                        39

  //Luz direta do sol         100,000                       8

  // Controle da luz externa

  if (luminosidadeExterna > 100) {

    digitalWrite(Luz\_Externa, HIGH); // Liga a luz externa

  } else {

    digitalWrite(Luz\_Externa, LOW); // Desliga a luz externa

  }

  // Controle da luz interna

  if (luminosidadeInterna > 100 && dist\_in\_cm < 200) {

    digitalWrite(Luz\_Interna, HIGH); // Liga a luz interna

    delay(10000);

  } else {

    digitalWrite(Luz\_Interna, LOW); // Desliga a luz interna

  }

  delay(500);

}

**Diagrama JSON**

{

  "version": 1,

  "author": "André Cunha",

  "editor": "wokwi",

  "parts": [

    { "type": "board-esp32-devkit-c-v4", "id": "esp", "top": 0, "left": 0, "attrs": {} },

    {

      "type": "wokwi-photoresistor-sensor",

      "id": "ldr1",

      "top": 89.6,

      "left": -402.4,

      "attrs": {}

    },

    {

      "type": "wokwi-led",

      "id": "led1",

      "top": -166.8,

      "left": -447.4,

      "attrs": { "color": "white", "lightColor": "orange" }

    },

    {

      "type": "wokwi-resistor",

      "id": "r1",

      "top": -91.2,

      "left": -451.75,

      "rotate": 90,

      "attrs": { "value": "1000" }

    },

    {

      "type": "wokwi-hc-sr04",

      "id": "ultrasonic1",

      "top": -190.5,

      "left": -224.9,

      "attrs": { "distance": "400" }

    },

    {

      "type": "wokwi-photoresistor-sensor",

      "id": "ldr2",

      "top": 99,

      "left": 324.4,

      "rotate": 180,

      "attrs": {}

    },

    {

      "type": "wokwi-led",

      "id": "led2",

      "top": -80.4,

      "left": 445.4,

      "attrs": { "color": "white", "lightColor": "orange" }

    },

    {

      "type": "wokwi-resistor",

      "id": "r2",

      "top": 14.4,

      "left": 441.05,

      "rotate": 90,

      "attrs": { "value": "1000" }

    }

  ],

  "connections": [

    [ "esp:TX", "$serialMonitor:RX", "", [] ],

    [ "esp:RX", "$serialMonitor:TX", "", [] ],

    [ "ldr1:VCC", "esp:3V3", "red", [ "h144", "v-76.8" ] ],

    [ "ldr1:GND", "esp:GND.1", "black", [ "h86.4", "v38" ] ],

    [ "ldr1:AO", "esp:33", "green", [ "h67.2", "v172.1" ] ],

    [ "esp:TX0", "$serialMonitor:RX", "", [] ],

    [ "esp:RX0", "$serialMonitor:TX", "", [] ],

    [ "led1:A", "r1:1", "green", [ "v0" ] ],

    [ "esp:GND.2", "led1:C", "black", [ "v-38.4", "h-437.16" ] ],

    [ "ultrasonic1:GND", "esp:GND.1", "black", [ "v57.6", "h-318", "v192" ] ],

    [ "ultrasonic1:ECHO", "esp:18", "green", [ "v67.2", "h277.6", "v134.4" ] ],

    [ "ultrasonic1:TRIG", "esp:17", "orange", [ "v355.2", "h287.6", "v-134.4" ] ],

    [ "ultrasonic1:VCC", "esp:3V3", "red", [ "v38.4", "h-110.59" ] ],

    [ "ldr2:VCC", "esp:3V3", "red", [ "h-38.4", "v-163.2", "h-283.39" ] ],

    [ "ldr2:GND", "esp:GND.1", "black", [ "h-48", "v96.4", "h-288", "v-76.8" ] ],

    [ "ldr2:AO", "esp:25", "green", [ "h-19.2", "v154.3", "h-336", "v-163.2" ] ],

    [ "r2:1", "led2:A", "green", [ "h0" ] ],

    [ "esp:GND.2", "led2:C", "black", [ "v0" ] ],

    [ "r2:2", "esp:5", "green", [ "h-268.8", "v66" ] ],

    [ "r1:2", "esp:32", "green", [ "v85.2", "h417.41" ] ]

  ],

  "dependencies": {}

}

**Link para Vídeo Explicativo postado no YouTube**

Para uma demonstração prática da solução no Wokwi, veja:

<https://www.youtube.com/watch?v=OnoLvKlGCsQ>

O projeto também pode ser acessado em:

<https://wokwi.com/projects/414376808765722625>

* **Computational Thinking with Python (CTWP)**

     O objetivo desta seção é desenvolver um sistema automatizado em Python que gerencie e otimize o consumo energético em residências. O sistema simula a coleta de dados dos sensores do ESP32, armazenando informações sobre luminosidade, distância, tarifas de energia e consumo interno em um arquivo JSON. Com base nesses dados, o sistema alterna inteligentemente entre a energia da rede e a gerada por painéis solares, priorizando a fonte mais econômica e sustentável. Além disso, um algoritmo antecipa a produção de energia solar em dias úmidos, carregando dispositivos de armazenamento antes de picos de consumo. A interface de usuário, desenvolvida com a biblioteca tkinter, permite a visualização dos dados e relatórios de eficiência energética, facilitando o monitoramento em tempo real e a tomada de decisões para reduzir custos e aumentar a eficiência energética.

**Código python**

import random

import time

import json

# Função para simular a coleta de dados dos sensores do ESP32

def simulate\_sensor\_data():

    data = {

        "luminosity": random.randint(0, 3000),  # Valores de luminosidade

        "distance": random.randint(0, 400),     # Valores de distância

        "energy\_price": random.uniform(0.1, 0.5),  # Tarifas de energia

        "internal\_consumption": random.uniform(0, 5)  # Consumo interno em kWh

    }

    return data

# Função para salvar os dados simulados em um arquivo JSON

def save\_data\_to\_file(data, filename="sensor\_data.json"):

    with open(filename, 'w') as file:

        json.dump(data, file)

# Função principal para simular a coleta de dados e salvar em arquivo

def main():

    sensor\_data = []

    for \_ in range(10):  # Simulando 10 leituras de sensores

        data = simulate\_sensor\_data()

        sensor\_data.append(data)

        time.sleep(1)

    save\_data\_to\_file(sensor\_data)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

# Função para selecionar automaticamente a fonte de energia mais econômica e sustentável

def switch\_energy\_source(sensor\_data):

    for data in sensor\_data:

        if data["luminosity"] > 2000 and data["energy\_price"] > 0.3:

            print("Usando energia solar")

        else:

            print("Usando energia da rede")

if \_\_name\_\_ == "\_\_main\_\_":

    with open("sensor\_data.json", 'r') as file:

        sensor\_data = json.load(file)

    switch\_energy\_source(sensor\_data)

# Interface de usuário para visualização dos dados dos sensores

import tkinter as tk

from tkinter import ttk

def display\_data(sensor\_data):

    root = tk.Tk()

    root.title("Dados dos Sensores")

    tree = ttk.Treeview(root, columns=("Luminosidade", "Distância", "Tarifa de Energia", "Consumo Interno"), show='headings')

    tree.heading("Luminosidade", text="Luminosidade")

    tree.heading("Distância", text="Distância")

    tree.heading("Tarifa de Energia", text="Tarifa de Energia (R$/kWh)")

    tree.heading("Consumo Interno", text="Consumo Interno (kWh)")

    for data in sensor\_data:

        tree.insert("", tk.END, values=(data["luminosity"], data["distance"], data["energy\_price"], data["internal\_consumption"]))

    tree.pack()

    root.mainloop()

if \_\_name\_\_ == "\_\_main\_\_":

    with open("sensor\_data.json", 'r') as file:

        sensor\_data = json.load(file)

    display\_data(sensor\_data)

# Função para gerenciamento e otimização do consumo energético

def manage\_energy\_consumption(sensor\_data):

    for data in sensor\_data:

        if data["luminosity"] > 2000 and data["energy\_price"] > 0.3:

            print("Usando energia solar")

        else:

            print("Usando energia da rede")

        if data["luminosity"] < 1000:

            print("Carregando dispositivos de armazenamento")

if \_\_name\_\_ == "\_\_main\_\_":

    with open("sensor\_data.json", 'r') as file:

        sensor\_data = json.load(file)

    manage\_energy\_consumption(sensor\_data)

**Saída de dados**

Tabela

Descrição gerada automaticamente com confiança média

Texto

Descrição gerada automaticamente

**Resultados Esperados**

A solução promete economia de até 45% do consumo energético com um impacto mínimo no conforto. Usando um ar-condicionado como exemplo, o sistema é capaz de ajustar a temperatura com base na presença e no clima externo, diminuindo o uso desnecessário. Além disso, o sistema pode prolongar a vida útil dos aparelhos ao usá-los de forma otimizada. A integração com Oracle permite uma análise detalhada do histórico de consumo, facilitando uma gestão energética precisa e previsível.

**Conclusão**

A implementação desta solução representa um avanço significativo na gestão energética, destacando o potencial transformador da Inteligência Artificial no setor de energia. Ao combinar machine learning, IoT e a robustez do Oracle, criamos uma plataforma que não apenas otimiza o consumo de energia, mas também se adapta continuamente às mudanças ambientais e comportamentais.

No contexto de um mercado de energia em rápida evolução, onde a sustentabilidade e a eficiência são imperativos, a integração de IA oferece uma vantagem competitiva crucial. Esta solução demonstra como a tecnologia pode transformar a maneira como gerenciamos e consumimos energia, promovendo um uso mais inteligente e sustentável dos recursos.

À medida que avançamos para um futuro em que as fontes de energia renováveis se tornam cada vez mais predominantes, a capacidade de prever e ajustar o consumo em tempo real será essencial. A IA facilita essa transição, garantindo que ela seja feita de maneira eficiente e econômica.

Este projeto não é apenas uma resposta aos desafios atuais, mas também um modelo para o futuro da gestão energética. Ele destaca o papel crucial da IA em criar um mercado de energia mais resiliente e adaptável, promovendo um futuro mais sustentável e eficiente para todos.

**Vídeo Explicativo**

* **(inserir link depois que gravarmos)**