Para um projeto simples, funcional e alinhado com o ODS 6 (Água Potável e Saneamento), podemos desenvolver uma solução que ajude no monitoramento de consumo de água em tempo real, com foco em conservação e uso eficiente desse recurso essencial. A ideia é criar um sistema de monitoramento de vazamentos ou medição de consumo de água, integrando um protótipo com sensor e atuador, comunicação via MQTT, e fácil implementação, podendo ser feito tanto com um protótipo físico quanto com simuladores.

**Título do Projeto: Sistema de Monitoramento de Consumo de Água com Alerta de Vazamento**

**Objetivo:**

Desenvolver um sistema simples e prático que monitore o consumo de água e detecte vazamentos em tempo real, com comunicação via MQTT para alertar o usuário sobre possíveis desperdícios de água.

**Componentes:**

1. **Sensor de Fluxo de Água (Sensor de Vazão)**:
   * Esse sensor irá medir a quantidade de água que passa por ele em um período de tempo. Pode ser utilizado um sensor de fluxo como o **YF-S201**.
   * Ele detecta o fluxo de água e gera pulsos, que são convertidos em dados que informam o consumo de água.
2. **Atuador:**
   * **Válvula Solenoide**: O sistema pode ser configurado para acionar uma válvula solenoide, interrompendo o fluxo de água quando um vazamento ou consumo excessivo for detectado.
   * Quando o sensor de fluxo detecta um consumo anômalo (vazamento ou consumo excessivo), a válvula é acionada para fechar o fluxo de água, evitando desperdícios.
3. **Plataforma de Comunicação MQTT**:
   * Para a comunicação via Internet, utilizaremos um microcontrolador como o **NodeMCU (ESP8266)** ou **ESP32**, que se conectará à rede Wi-Fi e enviará os dados para um broker MQTT.
   * Quando o sensor detectar um vazamento ou consumo elevado, o sistema enviará uma mensagem MQTT com a informação de alerta (por exemplo, "Vazamento Detectado") para um aplicativo ou servidor na nuvem.
4. **Dashboard ou Aplicativo**:
   * A interface pode ser um aplicativo simples ou uma página web que exibe os dados de consumo de água em tempo real.
   * Usando um serviço de MQTT como **ThingSpeak** ou **Blynk**, o usuário poderá visualizar o status do consumo de água, histórico de dados e receber notificações quando o sistema detectar problemas.

**Como Funciona:**

1. O **sensor de fluxo de água** é instalado na tubulação de água (pode ser em um ponto específico da casa, como no ponto de entrada de água).
2. O **NodeMCU** coleta os dados do sensor e, com base na quantidade de água que passa pelo sensor, calcula o consumo.
3. Se o consumo de água ultrapassar um limite predeterminado ou se o sensor detectar um fluxo de água contínuo (indicando vazamento), o **NodeMCU** envia uma mensagem de alerta via **MQTT** para a plataforma de monitoramento.
4. O **atuador (válvula solenoide)** pode ser ativado para cortar o fluxo de água automaticamente em caso de vazamento.
5. O usuário pode monitorar em tempo real e receber notificações via o **aplicativo ou dashboard**.

**Tecnologias e Ferramentas:**

* **Sensor de Fluxo de Água (YF-S201)**
* **NodeMCU ESP8266 ou ESP32** (plataforma de protótipo com Wi-Fi)
* **MQTT** (protocolo para comunicação de mensagens)
* **Válvula Solenoide** (atuador para controle do fluxo de água)
* **Plataforma de Dashboard** (Blynk, ThingSpeak, ou criação própria)
* **Arduino IDE** (para programação do microcontrolador)
* **Wokwi ou Tinkercad** (para simulação do projeto)

**Desenvolvimento do Projeto:**

1. **Sensor e Leitura dos Dados**:
   * O código no NodeMCU será responsável por ler os pulsos do sensor de fluxo de água.
   * A partir do número de pulsos, o microcontrolador calculará a quantidade de água consumida.
2. **Comunicação via MQTT**:
   * A comunicação via MQTT será configurada para enviar os dados do consumo de água para um broker na internet, que pode ser acessado pelo usuário através de um aplicativo.
3. **Atuação da Válvula**:
   * Em caso de vazamento ou consumo excessivo de água, o NodeMCU pode controlar a válvula solenoide para interromper o fornecimento de água até que o problema seja resolvido.
4. **Visualização e Notificação**:
   * O usuário poderá visualizar em tempo real o consumo de água, histórico de consumo, e receber alertas de vazamentos ou consumo elevado via o dashboard ou aplicativo.

**Exemplo de Código Simples (NodeMCU + Sensor de Fluxo):**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

// Defina as credenciais Wi-Fi

const char\* ssid = "seuSSID";

const char\* password = "suaSenha";

// Defina o broker MQTT

const char\* mqtt\_server = "broker.mqtt.com";

WiFiClient espClient;

PubSubClient client(espClient);

// Defina o pino do sensor de fluxo

const int flowSensorPin = D2;

volatile int flowCount = 0;

void flowSensorInterrupt() {

flowCount++;

}

void setup() {

Serial.begin(115200);

pinMode(flowSensorPin, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(flowSensorPin), flowSensorInterrupt, FALLING);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Conectando ao Wi-Fi...");

}

Serial.println("Conectado ao Wi-Fi");

client.setServer(mqtt\_server, 1883);

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

// Enviar dados via MQTT

if (flowCount > 0) {

float flowRate = flowCount \* 2.25; // Ajuste conforme o sensor

flowCount = 0;

String message = "Consumo de água: " + String(flowRate) + " L/h";

client.publish("casa/agua", message.c\_str());

}

}

void reconnect() {

while (!client.connected()) {

if (client.connect("ESP8266Client")) {

client.subscribe("casa/agua");

} else {

delay(5000);

}

}

}

**Considerações Finais:**

* O projeto é simples e prático, podendo ser desenvolvido com facilidade em plataformas como Wokwi (simulador) ou usando protótipos físicos.
* A integração com MQTT permite que o sistema seja escalável, podendo ser acessado e monitorado remotamente.
* Atende diretamente ao ODS 6, ao ajudar na redução do desperdício de água e no controle do consumo de água de maneira eficiente e inteligente.