

FIAP – FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA

Alberto R. Peruchi – RM99706 - 2TDSR
Angelo Augusto Pelluci - RM98310 - 2TDSR
Gabrielle da Silva Stanguini RM98786 - 2TDSA
Maria Fernanda Ribeiro Melo RM98818 - 2TDSA
Nicolas Reis do Espírito Santo RM98819 - 2TDSA

**GLOBAL SOLUTION - DISRUPTIVE
ARCHITECTURES: IOT, IOB & GENERATIVE IA**
Documento da Global Solution 2ºSemestre

São Paulo
2024

SUMÁRIO

1. Descrição da ideia.....	03
2. Público Alvo.....	03
3. Benefícios da solução.....	04
4. Explicação com Imagens.....	05
4.1. Forma oficial, via circuito para o MQTT.....	07
4.2. Caso o código com o MQTT não rode.....	09
5. Link GitHub.....	11
6. Link do projeto no Wokwi.....	11
7. Link do vídeo explicação.....	11

1- Descrição da Ideia:

A **SunWise** é uma solução inovadora desenvolvida para otimizar o controle de irrigação e iluminação em estufas agrícolas, promovendo eficiência e sustentabilidade. Através de sensores de umidade e controle automatizado de iluminação, a plataforma monitora as condições de umidade do solo e o horário do dia, ajustando a irrigação e as luzes de forma inteligente. Com interface web, os usuários têm acesso fácil e preciso às condições das suas estufas, garantindo a melhor utilização dos recursos disponíveis.

A solução foi projetada para atender a produtores agrícolas, gestores de estufas e agricultores que buscam reduzir o consumo de energia, otimizar os processos de irrigação e melhorar o controle das condições ambientais dentro das estufas. Pode ser utilizada em estufas de diversos tamanhos, tanto por pequenos produtores quanto por grandes empresas agrícolas.

2- Público Alvo:

O público alvo da solução é voltado para produtores agrícolas, especialmente pequenos e médios produtores que cultivam hortaliças, flores e outros vegetais em estufas, permitindo que tenham controle automatizado e eficiente sobre a irrigação e iluminação de seus cultivos. Gestores de estufas comerciais também se beneficiam, pois, o sistema facilita o monitoramento e a manutenção dos recursos hídricos e energéticos, simplificando o gerenciamento das operações diárias. A solução é ideal para empresas agrícolas sustentáveis que buscam integrar práticas mais ecológicas e econômicas em seus processos de produção, alinhando-se aos princípios da agricultura sustentável. Além disso, a solução atende agricultores urbanos, que cultivam produtos em estufas em centros urbanos e estão em busca de sistemas que otimizem a irrigação e o controle de iluminação, contribuindo para a sustentabilidade e o uso eficiente de recursos.

3- Benefícios da Solução:

A **SunWise** oferece uma série de benefícios para os consumidores, principalmente ao reduzir custos com operação. O sistema automatiza a irrigação, ativando-a apenas quando necessário com base na umidade do solo, evitando o desperdício de água e também de energia que é usada para fazer a bomba de água funcionar durante a irrigação. Além disso, o controle de iluminação ajusta o uso das luzes dependendo da hora do dia, garantindo que a energia elétrica seja utilizada de forma eficiente, evitando o uso desnecessário.

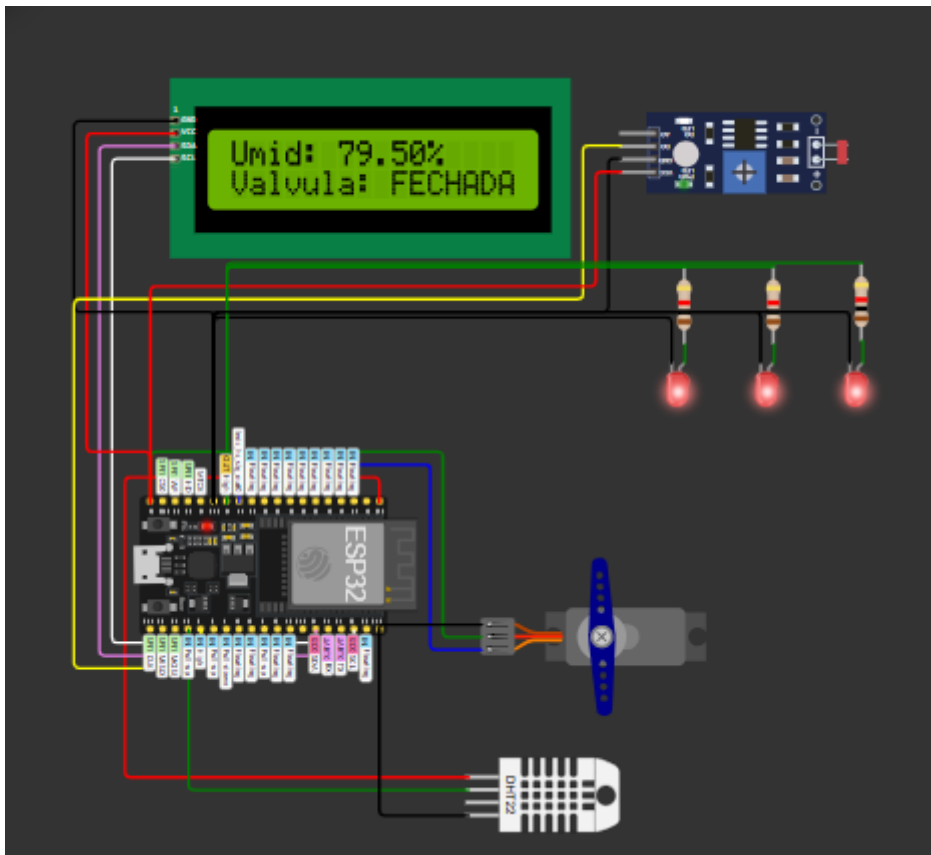
Outro benefício importante é a maior eficiência no uso de recursos. O sistema ajusta automaticamente a irrigação com base na umidade do solo, garantindo que as plantas recebam a quantidade ideal de água, o que promove um melhor crescimento e evita o uso excessivo de recursos. Com o monitoramento em tempo real, o usuário pode visualizar a umidade do solo, o status da irrigação e da iluminação, o que torna a gestão da estufa mais prática e eficiente. A solução oferece um dashboard fácil de usar, com gráficos de históricos de umidade, proporcionando aos usuários uma visão detalhada e atualizada de suas operações.

Além disso a solução possui um impacto significativo no meio ambiente, começando pela economia de água, ao controlar de forma inteligente a irrigação, evitando desperdícios. Também contribui para a redução das emissões de carbono por conta do controle da iluminação inteligente, que ajusta as luzes conforme a necessidade de luminosidade externa, reduzindo o consumo de energia elétrica e contribuindo para a diminuição das emissões de carbono associadas a produção de energia.

4- Explicação com imagens:

Para fazer esse projeto, nós utilizamos o simulador de circuito Wokwi.

Aqui está uma imagens do circuito montado:



No código do circuito, nós colocamos o Servo Motor (a válvula), para representar o sistema de irrigação e a passagem ou não da água na irrigação. Para fazer a condição da válvula abrir ou fechar, nós colocamos esse sensor DHT22 que simula um valor de umidade e temperatura. A partir dos valores inseridos nele, nós criamos uma condição para que quando a umidade atinja um valor determinado de 50 ou mais, a válvula fecha e mostra o status da umidade e da válvula no monitor, representando a pausa na irrigação e na passagem de água. Caso a umidade atinja um valor inferior a 50, a válvula se abre e o status de irrigação muda no monitor, representando a passagem de água ativa na irrigação. Também colocamos um sensor de luminosidade, para caso o sensor detecte a presença de LUZ no ambiente, representando a luz solar durante o dia, os leds se apagam. Já, caso o sensor de luminosidade não detecte a presença de luz (solar), ele ativa os leds, que ficam ligados até as condições de luminosidade mudarem de estado.

```
{"temperatura":24,"umidade":33,"luminosidade":4095}  
Dados enviados ao MQTT:  
{"temperatura":24,"umidade":33,"luminosidade":4095}  
{"temperatura":24,"umidade":33,"luminosidade":4095}  
Dados enviados ao MQTT:  
{"temperatura":24,"umidade":33,"luminosidade":4095}
```

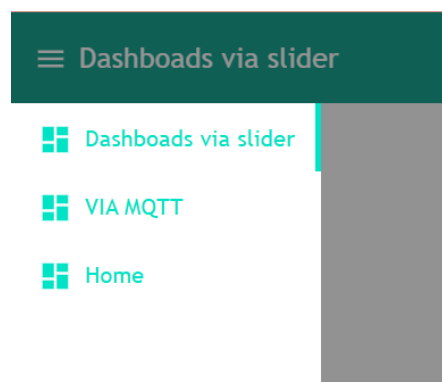
No Monitor Serial, printamos um json com os números em tempo real, que se ajusta a cada mudança nos sensores, da temperatura, umidade e luminosidade (sobre esses valores, o número 4095 indica a presença de luz, então os leds devem aparecer desligados. Caso o número esteja como 0 na luminosidade, isso representa a ausência de luz, logo os leds devem aparecer ligados no circuito).

Esse json é enviado como mensagem para o mqtt para ser utilizado no node-red, para geração de gráficos. Tivemos alguns problemas na compilação desse código, por inconsistências no servidor do Wokwi. Caso haja algum problema no dia e não rode o código, deixamos 2 alternativas para visualização dos dados no node-red.

No dashboard do Node-Red, criamos 3 páginas:

-A página Dashboards via slider, é a página de demonstração dos dados, sem nenhuma integração com MQTT, somente construída para visualização e demonstração dos dados.

-A página VIA MQTT, é o flow com a conexão do MQTT. Ou seja, nessa página é onde os dados são enviados via MQTT.

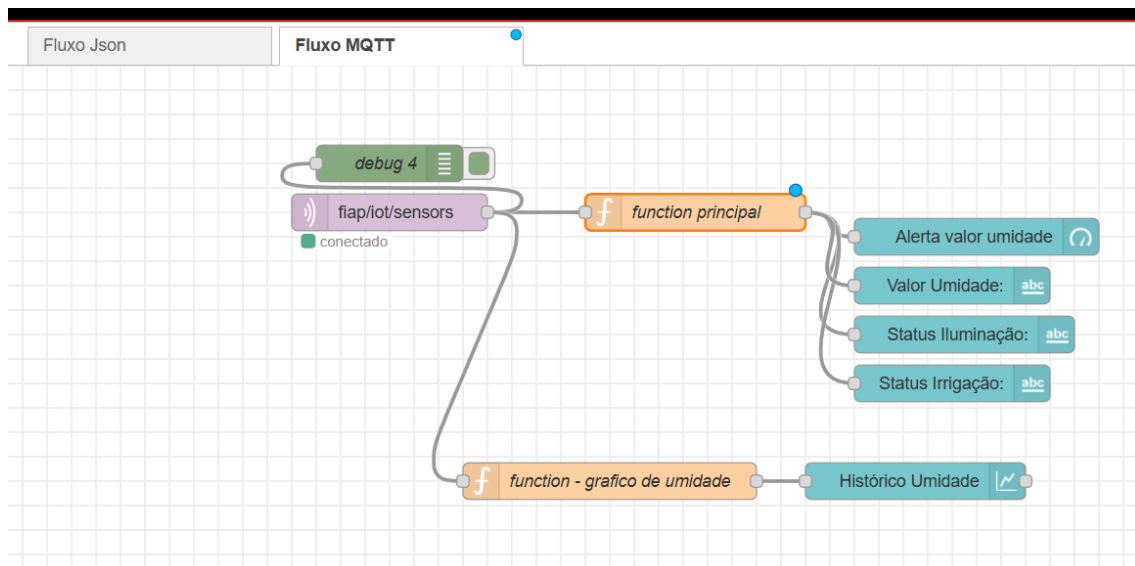


- A página HOME está mostrando um pouco sobre o projeto de forma simples:



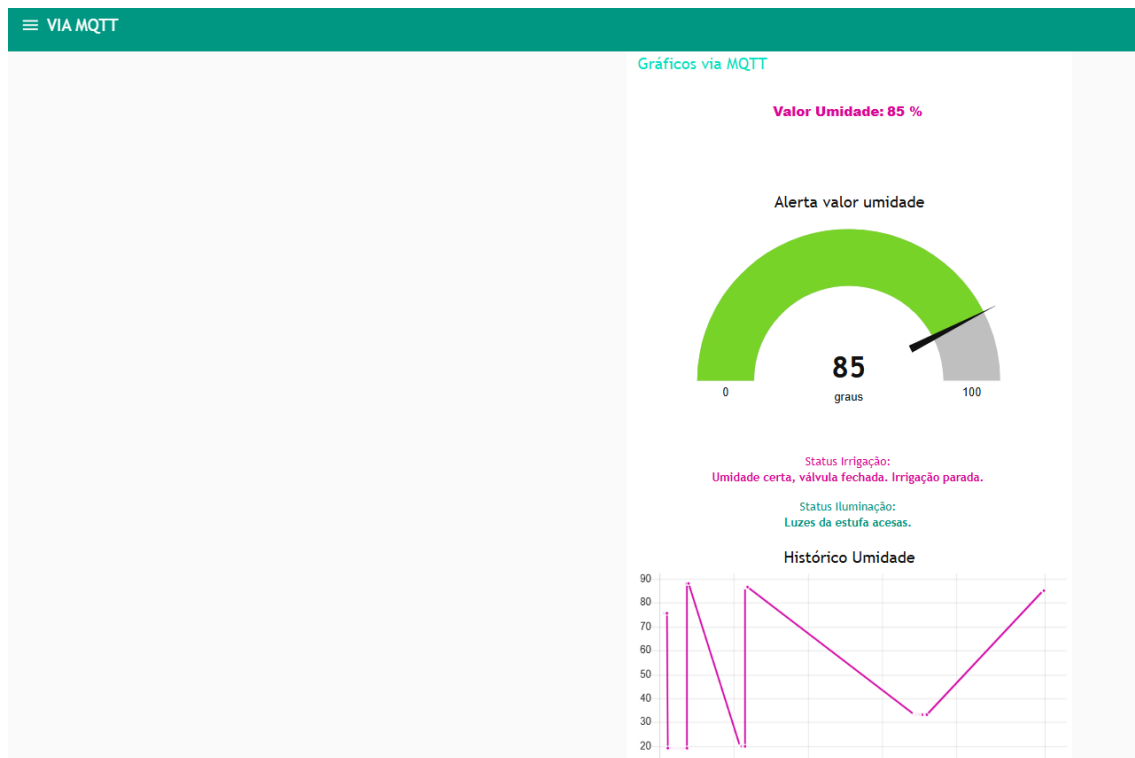
4.1 - Forma oficial, via circuito para o MQTT:

Caso o código oficial, com integração com MQTT, esteja rodando (sem falhas do servidor Wokwi), o flow utilizado será o “Fluxo MQTT”.



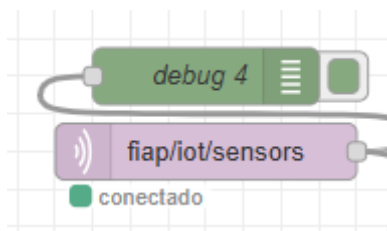
Mandamos do circuito no Wokwi para o MQTT, o json com as informações dos sensores e seus valores. Com esse json vindo do MQTT, fizemos uma função que separa esses valores vindos e mostra os status de irrigação e iluminação. Para o status de irrigação, fizemos uma condição se o valor da irrigação for menor que 50, a mensagem de “Irrigação baixa, válvula aberta” é exibida. Se o valor for maior que 50, a mensagem de “Irrigação correta, válvula fechada” é exibida. Para o status da iluminação, pegamos o valor vindo do mqtt. Se o valor for 0, a mensagem “Luzes acesas” é exibida. Se o valor for 4095 a mensagem “Luzes apagadas” é mostrada. Para essa exibição colocamos dois textos conectados ao function node. Também colocamos um gauge para mostrar o

valor da umidade e servir de alerta para o usuário. Como funcionalidade final colocamos um chart que mostra o histórico de umidade.



Para que as informações do circuito sejam enviadas para o MQTT e os dados apareçam no NODE-RED, você deve:

- 1-Baixar o arquivo flows do repositório do GitHub
- 2-Abrir esse flow no Node-Red e clicar em "Implementar"
- 3-Você deve estar o status "conectado" aparecer no bloco "mqtt in":



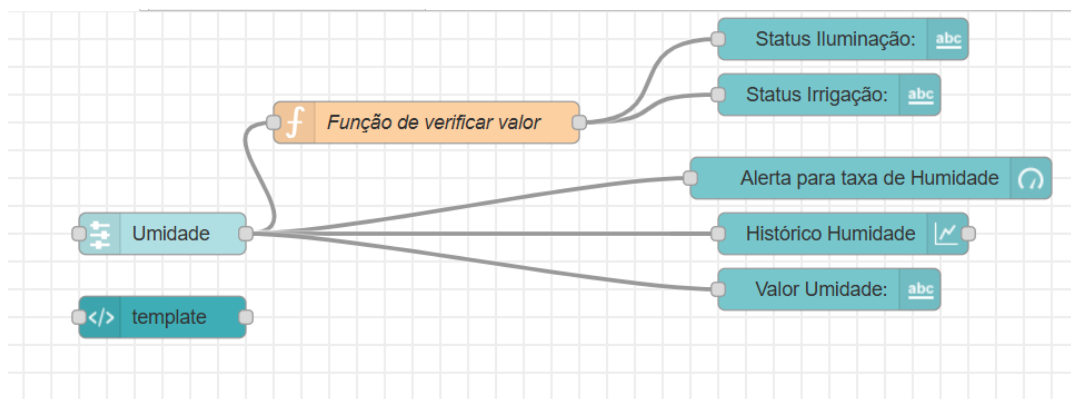
4-Depois que o status for para conectado, você deverá abrir o código no simulador do Wokwi e rodar o projeto.

5-Aguarde o código fazer a conexão com o MQTT e quando o json com as informações dos sensores aparecerem no monitor serial, você poderá abrir o dashboard no Node-RED, e verá que as informações enviadas no circuito, já estão nos gráficos do NODE.

4.2 – Caso o código com MQTT não rode:

Caso o código com MQTT não rode, você terá 2 alternativas para testar a simulação dos dados no NODE-RED.

1- Utilizando o flow “Fluxo Json”, que também será enviado no arquivo flows do repositório do GitHub.



Simulamos nossa entrada de dados (teoricamente vindos do circuito e dos sensores) no node-red via SLIDDER. Com essas informações vindas do slider, que demonstravam o valor da umidade, fizemos uma função para mostrar os status de irrigação e iluminação. Para o status de irrigação, fizemos uma condição se o valor da irrigação for menor que 50, a mensagem de “Irrigação baixa, válvula aberta” é exibida. Se o valor for maior que 50, a mensagem de “Irrigação correta, válvula fechada” é exibida. Para o status da iluminação, pegamos o horário da entrada de dados e fizemos um condição onde se o horário for entre 05h e 17h30, a mensagem “Luzes apagadas” é exibida. Se o horário for entre 17h31 e 4h59, a mensagem “Luzes acesas” é mostrada. Para essa exibição colocamos dois textos conectados ao function node. Também colocamos um gauge para mostrar o valor da umidade e servir de alerta para o usuário. Como funcionalidade final colocamos um chart que mostra o histórico de umidade.

2- Utilizando o flow “Fluxo MQTT”, mas mandando o json manualmente via Hive MQ.

1-Abra o Node-red e importe o arquivo flow do github.

2-Clique no botão “Implementar” e abra os dashboards.

3-Vá na página “VIA MQTT”

4-Conecte o broker no site do Hive MQ. Espere aparecer a bolinha verde na parte superior

5-Copie o topic abaixo e cole no campo “Topic” na parte “Publish”

6-Cole o json igual abaixo no campo “Message”.

7-Clique em “Publish”

8-Ná página “VIA MQTT”, o valor enviado pelo hive, já irá aparecer nos gráficos.

The image shows a web interface for MQTT management, divided into two main sections: "Connection" and "Publish".

Connection Section:

- Host:** mqtt-dashboard.com
- Port:** 8884
- ClientID:** clientId-Tcd1IUNf59
- Connect:** A blue button to initiate the connection.
- Username:** An empty text field.
- Password:** An empty text field.
- Keep Alive:** 60
- SSL:** A checkbox with an 'x' icon, currently unchecked.
- Clean Session:** A checkbox with an 'x' icon, currently unchecked.
- Last-Will Topic:** An empty text field.
- Last-Will QoS:** 0
- Last-Will Retain:** A checkbox, currently unchecked.
- Last-Will Message:** A large empty text area.

Publish Section:

- Topic:** fiap/iot/sensors
- QoS:** 0
- Retain:** A checkbox, currently unchecked.
- Publish:** A blue button to send the message.
- Message:** {"temperatura":24,"umidade":33,"luminosidade":4095}

Subscriptions Section:

- This section is currently collapsed, indicated by a downward arrow icon.

5- Link GitHub:

O código fonte exportado do Wokwi, juntamente com o link do simulador Wokwi localizado no ReadMe, e o arquivo do Flows do Node-Red, se encontram no repositório do GitHub.

<https://github.com/AngPF/GS-SunWise-IOT.git>

6- Link do projeto no Wokwi:

Link do código e do circuito funcional no Wokwi, com integração com o MQTT:

<https://wokwi.com/projects/414636232313826305>

Caso haja instabilidade do servidor Wokwi durante a compilação do projeto, aqui está o código sem a integração com o MQTT, que printa o json com as informações no monitor serial:

<https://wokwi.com/projects/414393427034769409>

7- Link do vídeo explicação:

Vídeo da explicação do código para a Global Solution

https://youtu.be/1FAO4_O5Q_I