

Proyecto Internet de las cosas

Allen Jimenez Gonzalez *, Daniela Alvarado Otoya † y Gustavo Mendez Poveda ‡

Escuela de Computación San Carlos, Instituto Tecnológico de Costa Rica

Email: *allenjimenezgonzalez@outlook.com, †alvarado22daniela@gmail.com, ‡mendez.p.gus@gmail.com

I. SOLUCIÓN PLANTEADA

En este proyecto se quiere conseguir la publicación y almacenamiento de datos de sensores de humedad por medio de Twitter y bases de datos, además de la conexión de los dispositivos necesarios para hacer que todo esto suceda.

Para esto se necesitará de Node-RED, conocimientos de JavaScript para usar Node-RED, 2 SP32, un sensor de humedad del aire, un sensor de humedad de la tierra y conocimientos básicos de C para programar sobre el SP32 y sus funcionalidades en los sensores.

También se necesita del uso de un servidor en la nube, para este proyecto se planea utilizar Digital Ocean ya que este tiene un acceso para estudiantes el cual no se había utilizado todavía, por lo tanto se tomara esto como opción para desarrollar la aplicación en esta plataforma, el único problema con la misma es que para el uso de esta se necesitara configurar manualmente y no como en la otras opciones (AWS y Azure).

Se necesitará conectar los SP32 a la red para que estos puedan conectarse directamente a la maquina virtual la cual estará hosteada en allen.wiki, en esta página se podrá abrir el node-red en donde esta la programación para la toma de los datos de los sensores y su publicación dentro de la base de datos de la maquina virtual y el posteo de los mismos en Twitter.

También se planea que los datos se puedan ver por medio de la UI que ofrece node-red y desde ahí poder

postear los datos de los dispositivos.

Para la realización del código de los sensores se planea tomar de ejemplo lo visto en clases de Redes GR50 2022 en donde se explicó la mayoría del funcionamiento del mismo proyecto a realizar.

II. RESULTADOS

Los resultados mostrados en este proyecto fueron exitosos, se pudieron tomar los datos de los diferentes dispositivos los cuales se encontraban conectados entre ellos y postear esa información en los diferentes lugares, además de enseñarla por medio de un chart en Node-RED.

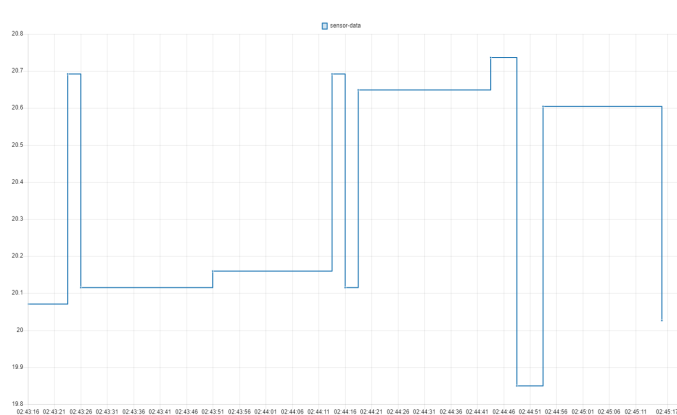
Y como se mencionó anteriormente, se tuvo que configurar la maquina virtual de Digital Ocean y se tuvieron inconvenientes con las librerías implementadas en el proyecto ya que estas estaban desactualizadas y entonces el programa no recogía algunas de las funciones y por lo tanto se tuvieron que actualizar para el desarrollo del proyecto.

Aunque todo lo realizado fue satisfactorio, también hay partes del proyecto las cuales no fueron implementadas tal y como la parte en donde se debía poder modificar la configuración de los sensores para tomar los datos. El código realizado se puede consultar en el repositorio [Redes-IoT](#).

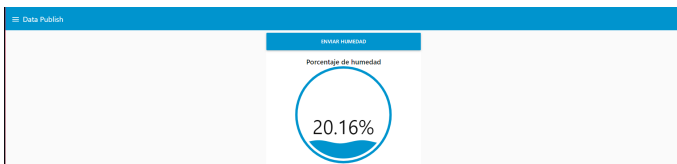
III. ANÁLISIS DE SOLUCIÓN/RESULTADOS

Analizando los resultados obtenidos de los sensores de humedad, se llegó a la conclusión de que los datos

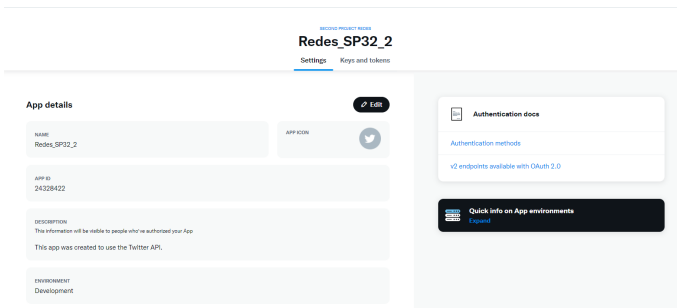
eran correctos ya que estos si cambiaban dependiendo de la humedad del ambiente en donde se encontraban los dispositivos, tal y como se puede ver en la siguiente imagen.



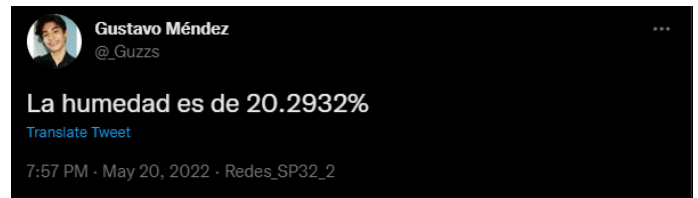
Por la parte visual para postear los datos en la base de datos y en Twitter, se utilizo el UI de Node-RED tal y como se puede ver a continuación.



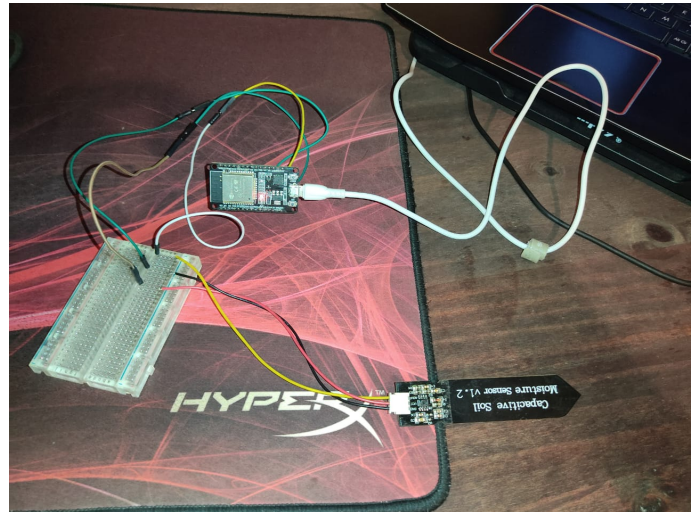
Para poder postear en Twitter se necesitó crear una cuenta de desarrollador y pedir suficiente acceso para poder postear Twitts (Elevated).



También podemos observar que luego de accionar el botón creado en Node-RED, este posteaba los datos que se recibían en ese momento en Twitter.



Se pueden observar las conexiones físicas entre el esp32 y el sensor de humedad en suelo



Todos los recursos utilizados fueron las guías y documentación que proveen cada página de cada tecnología utilizada:

- ESP-IDF Programming Guide [1]
- NodeRed:Documentation [2]
- Mosquitto:Documentation [3]
- Certbot:Documentation [4]
- Docker:Documentation [5]
- Mysql:Documentation [6]
- DigitalOcean:Documentation [7]

El código base posteriormente modificado para ajustarlo a las necesidades del proyecto fue tomado de los ejemplos de código abierto que provee la biblioteca de ESP-IDF Programming Guide [1]

IV. CONCLUSIONES

Se pudo lograr satisfactoriamente la comunicación entre el esp32 y el servidor de

almacenamiento/procesamiento de datos a través del protocolo MQTT con seguridad HTTPS, estos datos son recopilados por un nodo padre que posteriormente se encarga de publicarlos en el servidor, en el cual los recibe el NodeRed para distribuirlos a través de Twitter, medios gráficos o guardarlos en la base de datos.

No se pudo conseguir la comunicación desde el servidor hacia el dispositivo esp32 para su configuración

La red MESH no se pudo probar debido a la carencia de un segundo esp32, sin embargo, el código esta implementado y solo restaría probarlo.

V. RECOMENDACIONES

Se recomienda probar el resto de código realizado en un entorno con 2 esp32 conectados mediante red mesh para comprobar la correcta funcionalidad.

También se tendría un panorama más completo del proyecto con la implemetación de la modificación de la configuración de sensores para la toma de datos.

Se recomienda finalizar la parte de envío de datos desde la plataforma web hacia la red de sensores, donde falta implementar la recepción mqtt desde la web.

REFERENCES

- [1] Expressif, *ESP-IDF Programming Guide*, ver. 4.4.1. [Online]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html>
- [2] NodeRed, *NodeRed:Documentation*, ver. latest. [Online]. Available: <https://nodered.org/docs/>
- [3] Eclipse, *Mosquitto:Documentation*, ver. 2.0. [Online]. Available: <https://mosquitto.org/documentation/>
- [4] Certbot, *Certbot:Documentation*, ver. stable. [Online]. Available: <https://eff-certbot.readthedocs.io/en/stable/using.html>
- [5] Docker, *Docker:Documentation*, ver. 20.10.16. [Online]. Available: <https://eff-certbot.readthedocs.io/en/stable/using.html>
- [6] Mysql, *Mysql:Documentation*, ver. 5.37.38. [Online]. Available: <https://dev.mysql.com/doc/>
- [7] DigitalOcean, *DigitalOcean:Documentation*, ver. stable. [Online]. Available: <https://docs.digitalocean.com>