Informa Final de Actividades de Servicio Social

I. Datos generales y matrícula del prestador

Alumno:
Cruz Díaz Luis Ángel

Matrícula:
2183038433

II. Lugar y periodo de realización

Periodo de realización:
26 de Junio al 26 de Diciembre

Lugar:
Universidad Autónoma Metropolitana

III. Unidad, División y Licenciatura

Unidad:
Azcapotzalco

División:
Ciencias Básicas e Ingeniería

Licenciatura:
Ingeniería en Computación

IV. Nombre del plan, programa o proyecto en el que se participó

Nombre del Proyecto de Servicio Social:

Apoyo en el Laboratorio de Procesamiento de Lenguaje Natural e Internet de las Coosas

V. Nombre del asesor

Asesor:

Dr. Sánchez Martínez Leonardo Daniel

VI. Introducción

El presente documento reúne las actividades realizadas en el desarrollo de diversos proyectos relacionados con el Internet de las Cosas (IoT) y tecnologías asociadas, en el marco de un proyecto educativo desarrollado en la Universidad Autónoma Metropolitana Unidad Azcapotzalco.

El enfoque principal se centra en la integración de hardware y software para la monitorización, gestión y visualización de datos, aprovechando herramientas y tecnologías como microcontroladores (ESP8266 y ESP32), sensores, plataformas de desarrollo como Node-RED, bases de datos como InfluxDB y MySQL, y sistemas de visualización como Grafana.

Los resultados documentados en este trabajo representan una aproximación práctica al desarrollo de soluciones IoT, cubriendo desde la instalación y configuración de los entornos de desarrollo, hasta la implementación de sistemas completos de monitoreo en tiempo real.

VII. Objetivos generales y específicos

Objetivos generales

- Diseñar e implementar un sistema de monitoreo basado en IOT que permite recopilar, procesar y visualizar datos en tiempo real.
- Facilitar la replicación y escalabilidad del proyecto mediante una documentación técnica detallada y el uso de Docker como herramienta de contenedorización.

Objetivos específicos

- Configurar el entorno de desarrollo para trabajar con microcontroladores ESP32 y ESP8266.
- Implementar proyectos individuales con sensores conectados a un microcontrolador para preparar la integración con el sistema de monitoreo.
- Desplegar los servicios de InfluxDB, MySQL, Node-RED y Grafana en contenedores Docker sobre una Raspberry Pi.
- Diseñar flujos en Node-RED para integrar los datos recopilados y enviar alertas personalizadas por Telegram en caso de detectar anomalías.
- Crear dashboards en Grafana para visualizar datos históricos y en tiempo real provenientes de los sensores IoT.
- Documentar cada paso del proceso de desarrollo, desde la configuración inicial del entorno hasta la implementación y visualización final, con el fin de garantizar la reproducibilidad del proyecto.

VIII. Metología utilizada

La metodología empleada para el desarrollo de este proyecto se diseñó con base en un enfoque modular, asegurando que cada componente, desde la configuración de software hasta la implementación de hardware, fuera integrado de manera eficiente y documentado en detalle. A continuación, se describen los pasos principales:

- **Preparación del entorno de desarrollo:** Configuración de Linux Mint como sistema operativo base y la instalación de herramientas como Visual Studio Code, Arduino IDE y MQTTX.
- **Configuración de la Raspberry Pi como servidor IoT:** Instalación de Raspberry Pi OS, Docker y despliegue de servicios como EMQX, InfluxDB, MySQL, Node-RED y Grafana.
- **Desarrollo de proyectos individuales:** Implementación de proyectos con microcontroladores ESP32 y ESP8266, integrando sensores de temperatura, humedad, luz y gas.
- **Comunicación y almacenamiento de datos:** Establecimiento de comunicación entre los sensores y el servidor mediante MQTT, creación de flujos en Node-RED para procesar y enviar datos a las bases de datos, y generación de alertas por Telegram.
- **Visualización de datos en Grafana:** Creación de dashboards personalizados para visualizar datos en tiempo real y de forma histórica.
- **Documentación del proyecto:** Creación de una página web con tutoriales, guías y ejemplos de código para facilitar la comprensión y replicabilidad del proyecto.

IX. Actividades realizadas

A continuación, se detallan las actividades realizadas durante el desarrollo del proyecto, abarcando desde la preparación del entorno de desarrollo hasta la implementación y visualización de los datos recopilados.

Configuración del entorno de desarrollo

- **Instalación de Linux Mint:** Se configuró como sistema operativo base para desarrollar el proyecto, asegurando compatibilidad con las herramientas requeridas.
- **Instalación de herramientaa:**
  - **Visual Studio Code:** con la extensión PlatformIO para programar los microcontroladores ESP32 y ESP8266.
  - **Arduino IDE:** como alternativa para utilizar el serial protter y graficar datos de los sensores.
  - **MQTTX:** para probar la comunicación MQTT entre los dispositivos y el broker.

Configuración de la Raspberry Pi como servidor IoT

- **Instalación de Raspberry Pi OS:** Configuración del sistema operativo en la Raspberry Pi y la habilitación de SSH para acceder de forma remota.
- **Instalación de Docker:** Instalación y configuración para desplegar los servicios en contenedores de manera automática al iniciar la Raspberry Pi.
- **Despliegue y configuración de servicios:**
  - **EMQX:** como broker MQTT para la comunicación entre los sensores y el servidor.
  - **InfluxDB:** para almacenar datos de series temporales provenientes de los sensores.
  - **MySQL:** para el almacenamiento de los usuarios a los que se enviarán las alertas.
  - **Node-RED:** para la creación de flujos de datos y la integración con los servicios anteriores.
  - **Grafana:** para la visualización de los datos en dashboards personalizados.

Desarrollo de proyectos individuales

- **Microcontroladores ESP8266 y ESP32:** Configuración de pines, conexión a redes Wi-Fi y ejecución de pruebas básicas.
- **Integración de sensores:**
  - **DHT:** para medir temperatura y humedad.
  - **BH1750:** para medir la intensidad de la luz.
  - **MQ:** para detectar la presencia de gas.
- **Implementación de una pantalla LCD 16x2:** Configuración con interfaz I2C para mostrar lecturas en tiempo real de los sensores.

Comunicación y almacenamiento de datos

- **Flow para enviar alertas:** Creación de un flujo en Node-RED para enviar alertas por Telegram de forma automática y personalizada en caso de detectar anomalías.
- **Flow para guardar datos en InfluxDB:** Configurar un flujo para almacenar en influxDB los datos de los sensores enviados a través de MQTT.
- **Flow para almacenar usuarios en MySQL:** Creación de un flujo para almacenar los usuarios a los que se enviarán las alertas en una base de datos MySQL desde Telegram.
- **Creación de dashboards en Grafana:** Diseño de dashboards personalizados para visualizar los datos de los sensores en tiempo real y de forma histórica.

Documentación del proyecto

Se documentó cada paso del proceso de desarrollo en una página web, incluyendo tutoriales de instalación, guías de configuración y ejemplos de código. Esto asegura la replicabilidad del proyecto y facilita la comprensión de los conceptos y tecnologías utilizadas.

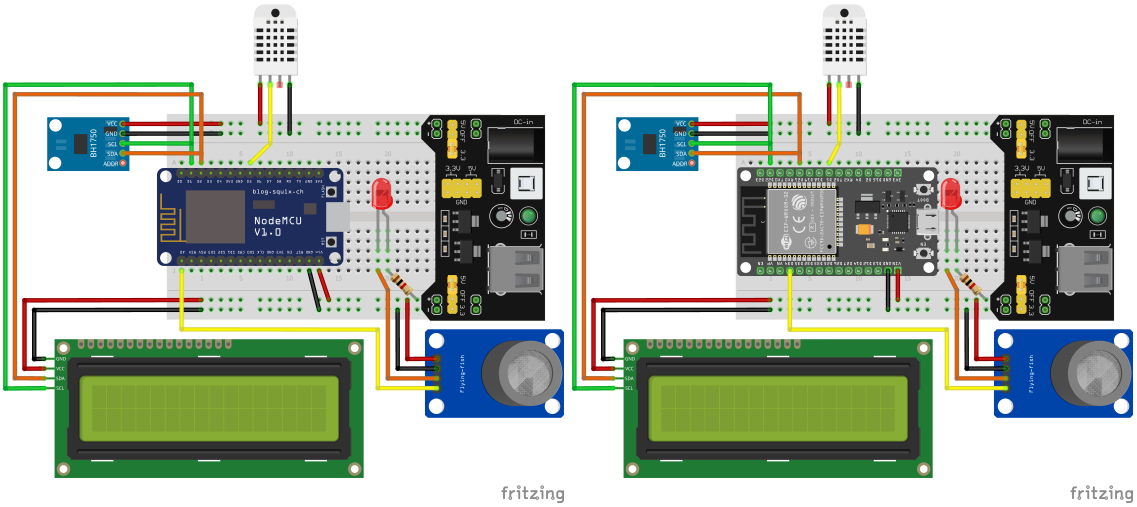
X. Objetivos y metas alcanzadas

- Configuración del entorno de desarrollo para trabajar con los microcontroladores ESP32 y ESP8266.
- Implementación de proyectos individuales con sensores como el de temperatura, humedad, luz y gas a un microcontrolador para la integración con el sistema de monitoreo.
- Despliegue de servicios de InfluxDB, MySQL, Node-RED y Grafana en contenedores Docker sobre una Raspberry Pi para la gestión y visualización de datos.
- Establecimiento de comunicación entre los sensores y el servidor mediante el protocolo MQTT.
- Creación de flujos en Node-RED para procesar y enviar datos a las bases de datos y generar alertas mediante Telegram.
- Creación de dashboards en Grafana para visualizar datos históricos y en tiempo real provenientes de los sensores IoT.
- Documentación detallada que abarca desde la configuración de herramientas hasta la implementación y el despliegue del proyecto en una página web, diseñada para ser fácil de usar y comprender.

XI. Resultados y conclusiones

Circuito del proyecto

El circuito completo después de juntar todos los proyectos individuales se muestra en la figura 1. En la figura 1a se muestra el circuito con los sensores conectados a un ESP8266 y en la figura 1b se muestra el circuito con los sensores conectados a un ESP32.



(a) Sensores conectados a un ESP8266 (b) Sensores conectados a un ESP32

Figura 1: Circuitos del proyecto

Para enviar los datos de los sensores al servidor, primero es necesario conectarse a Internet. Para simplificar este proceso, al encender el microcontrolador, este creará un hotspot al que podremos conectarnos para configurar las credenciales de la red a la que se conectará. En la figura 2 se muestra la página web que se despliega al conectarse al hotspot del microcontrolador. Una vez ingresadas las credenciales de la red, el microcontrolador se conectará a Internet y enviará los datos por MQTT al servidor.

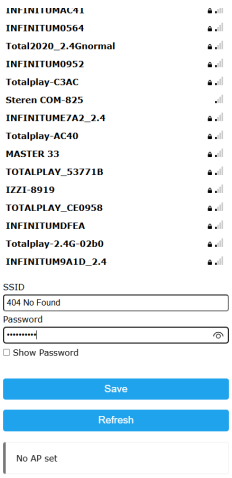


Figura 2: Página web para configurar las credenciales de la red

Guardar información de los sensores y enviar alertas personalizadas

Los datos enviados por los sensores se almacenan en una base de datos InfluxDB. En la figura 3 se puede ver el flujo de Node-RED que se encarga de guardar los datos en la base de datos después de recibirlos por MQTT.

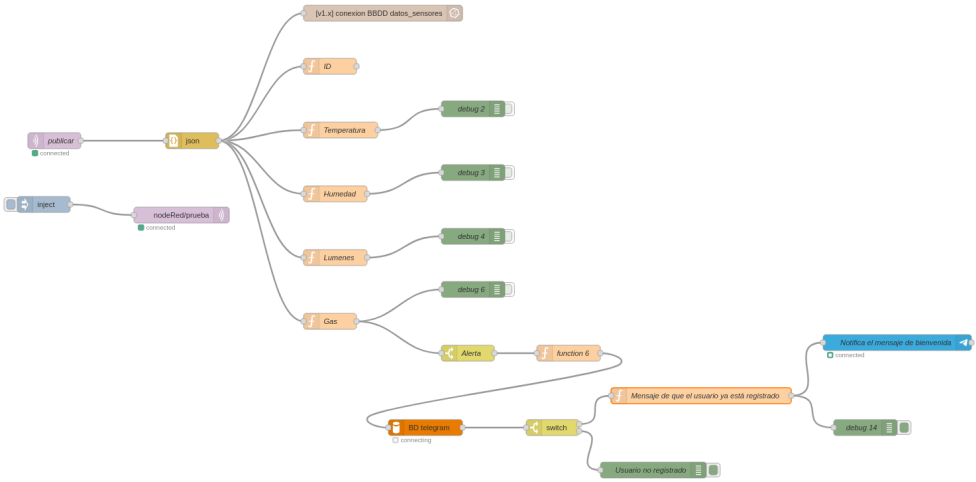


Figura 3: Flujo de Node-RED para guardar datos en InfluxDB

El flujo anterior también se encarga de enviar alertas personalizadas a través de Telegram cuando se detectan anomalías en los datos recibidos por MQTT. En la figura 4 se muestra un mensaje de alerta personalizado enviado por Telegram, el cual se envía únicamente a los usuarios registrados.

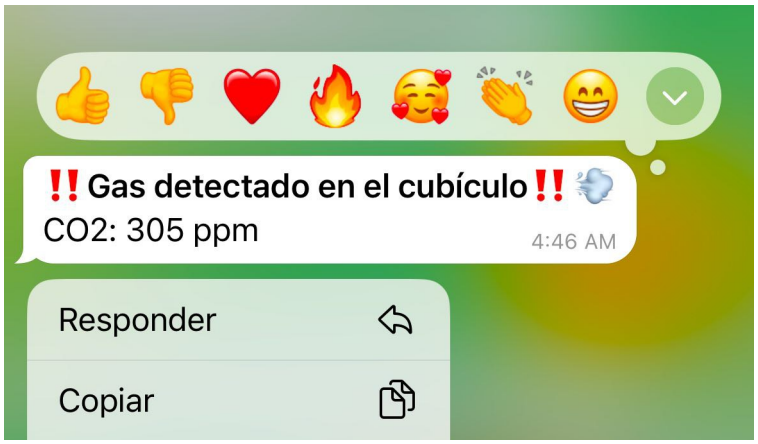


Figura 4: Alerta enviada por telegram

Registrar un usuario para recibir alertas

Para registrar un usuario y recibir alertas personalizadas por Telegram, es necesario enviar un mensaje al bot de Telegram con el comando `/start`. En la figura 5 se muestra el flujo de Node-RED que se encarga de registrar a los usuarios en una base de datos MySQL.

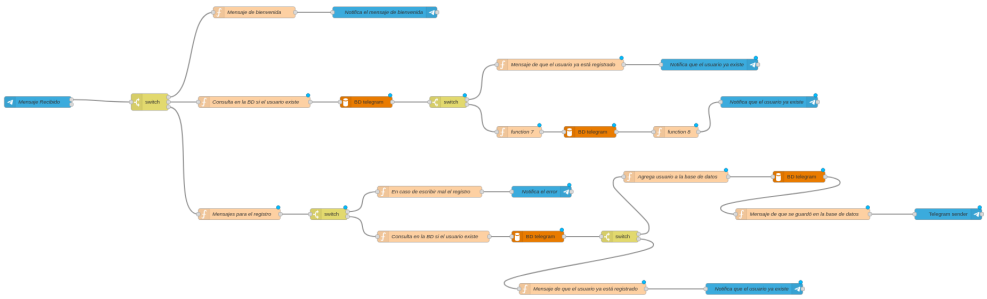


Figura 5: Flujo de Node-RED para registrar usuarios en MySQL

El flujo anterior se activa cuando un usuario envía el comando `/start` al bot de Telegram. En la figura 6 se muestra la interacción con el bot de Telegram para registrar un usuario.

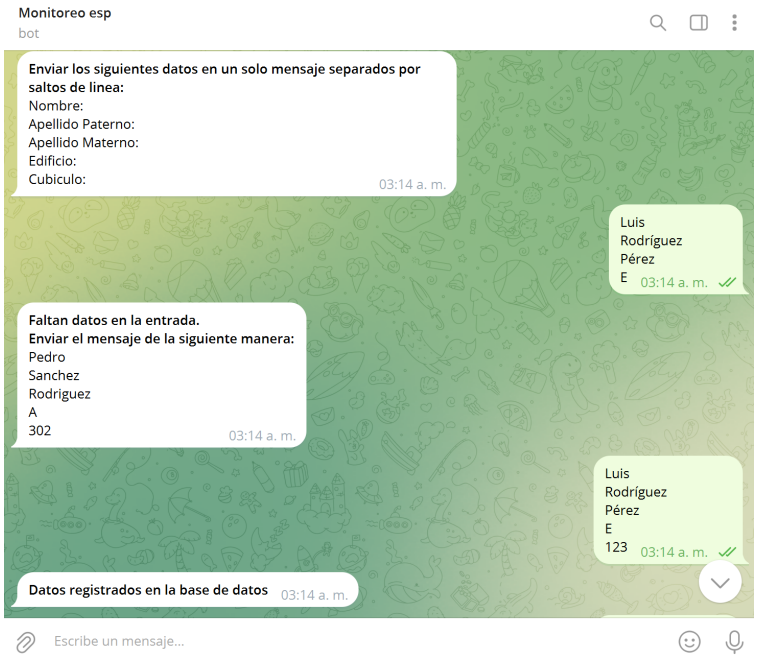


Figura 6: Registro de un usuario en Telegram

Visualización de datos en Grafana

Los datos almacenados en InfluxDB se visualizan en dashboards personalizados en Grafana. Estos datos son tomados desde la base de datos InfluxDB y se actualizan en tiempo real. En la figura 7 se muestra un dashboard con los datos de temperatura, humedad, luz y gas en tiempo real.

Figura 7: Dashboard en Grafana con los datos de los sensores

Documentación del proyecto

La documentación del proyecto fue publicada en github pages y se puede acceder a través del siguiente enlace: https://luisangel2801.github.io/Servicio-Social/. Este sitio web contiene tutoriales, guías y ejemplos de código para facilitar la comprensión y replicabilidad del proyecto. En la figura 8 se muestra la página principal de la documentación.

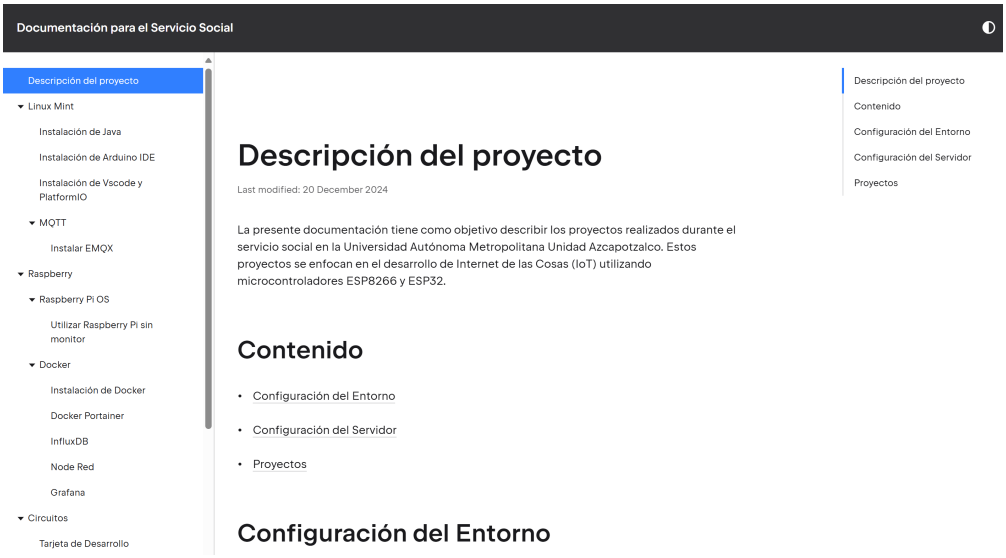


Figura 8: Pagina Web de la documentación del proyecto

La documentación en la página web ha sido diseñada para replicar el proyecto utilizando cualquier placa de desarrollo. Para mostrar el código adaptado a la placa seleccionada, debe elegirse la placa en la parte superior de la página. En la figura 9 se muestra cómo realizar esta selección.

Figura 9: Selección de la placa de desarrollo

XII. Recomendaciones

- Realizar pruebas de estrés en el servidor para evaluar su capacidad de respuesta y rendimiento ante un gran número de dispositivos conectados.
- Implementar un sistema de copias de seguridad para garantizar la integridad de los datos almacenados en las bases de datos.
- Escalar el proyecto para incluir más sensores y actuadores para realizar acciones en tiempo real en función de los datos leídos.

Referencias

[1] EMQ Technologies Inc. (n.d.). https://www.emqx.com/en.

[2] Docker. (2024, July 8). Docker: Accelerated Container Application Development. Docker. https://www.docker.com/

[3] InfluxData. (2021, December 10). InfluxDB: Open Source Time Series Database — InfluxData. https://www.influxdata.com/products/influxdb/

[4] Node-RED. (n.d.). Node-RED. https://nodered.org/

[5] Grafana Labs (n.d.). Grafana Labs — Leading observability tool for visualizations & dashboards. https://grafana.com/oss/grafana/

[6] Espressif Systems. (n.d.). ESP32 - Espressif Systems. https://www.espressif.com/en/products/socs/esp32

[7] Espressif Systems. (n.d.). ESP8266 - Espressif Systems. https://www.espressif.com/en/products/socs/esp8266