29-1-2022

**Proyecto IoT**

**GRUPO 1 Informática Industrial**

Cristina Jurado Herrero

Julián Martín Caro

# ÍNDICE

[ÍNDICE 1](#_Toc93338606)

[INTRODUCCIÓN Y OBJETIVOS 2](#_Toc93338607)

[DISEÑO HW Y ESQUEMAS DE CONEXIONADO 3](#_Toc93338608)

[Pines usados para cada sensor/dispositivo 3](#_Toc93338609)

[DISEÑO SW 4](#_Toc93338610)

[Diagrama de bloques/flujo del programa, Descripción de funcionamiento 4](#_Toc93338611)

[Técnicas que se han utilizado para hacer robusto el sistema. Por ejemplo, cómo se maneja el 6](#_Toc93338612)

[Estudio del tiempo de ejecución de distintas fases de funcionamiento y el respectivo 7](#_Toc93338613)

[consumo de energía. Posibles mejoras u optimizaciones introducidas y análisis de su 7](#_Toc93338614)

[impacto. 7](#_Toc93338615)

[Librerías utilizadas 8](#_Toc93338616)

[RESULTADOS Y CONCLUSIONES 9](#_Toc93338617)

[MANUAL DE USUARIO 10](#_Toc93338618)

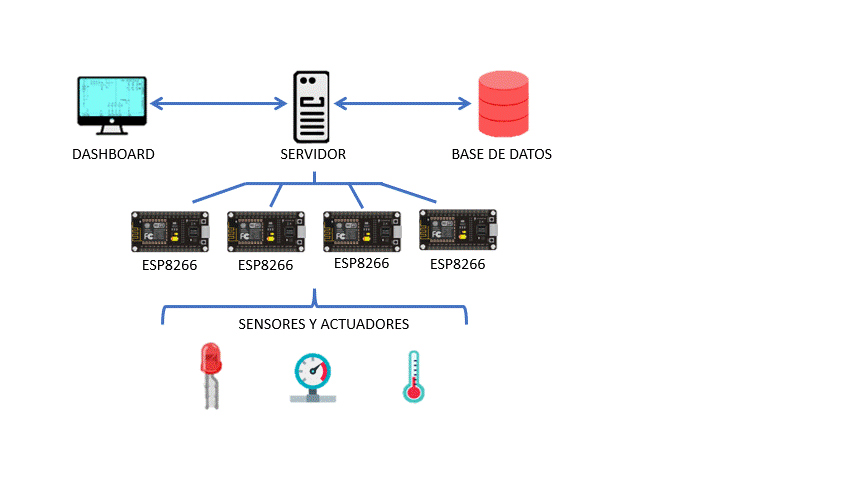
[LISTA Y DESCRIPCIÓN DE LOS FICHEROS ENTREGADOS 11](#_Toc93338619)

# INTRODUCCIÓN Y OBJETIVOS

Para la realización de este proyecto se ha partido de los requisitos mínimos proporcionados por el guion, y se ha añadido la funcionalidad de control por voz mediante una aplicación de móvil y Alexa.

El objetivo es crear un sistema SCADA accesible y comprensible para monitorizar y actualizar la información obtenida de sensores y actuadores, tales como leds y sensores de temperatura y humedad. Se ha implementado también que sea posible modificar a través del dashboard la frecuencia de muestreo de los datos, y que se almacenen en una base de datos de la que se puedan extraer y realizar operaciones.

Por último, se ha incluido que Telegram y Alexa, mediante comandos de voz, puedan ser preguntada por el estado de los sensores y que actúe sobre los actuadores.



# DISEÑO HW Y ESQUEMAS DE CONEXIONADO

### Pines usados para cada sensor/dispositivo

# DISEÑO SW

### Diagrama de bloques/flujo del programa, Descripción de funcionamiento

<https://www.diagrameditor.com/>

En Node-RED, hemos dividido el diagrama en 3 partes:

**TOPICS**

Se trata del flujo que implementa las recepciones y envíos de topics mediante MQTT en formato JSON desde el programa de Arduino. En este grupo se encuentran tanto señales de control, de actuación y de sensado.

**MONGODB**

En este flujo se implementa el almacenamiento en la base de datos de nuestro grupo, las señales medidas. Por otra parte admite peticiones de muestra de los datos obtenidos en un periodo de tiempo.

**TELEGRAM**

Mediante este flujo, se puede pedir a un Bot de Telegram las medidas actuales de los sensores, que notifique en el caso de que superen un máximo o mínimo, y también

Por otra parte, el programa de los ESP8266 realizado con Arduino, también se ha subdividido en partes para facilitar su comprensión:

**VARIABLES GLOBALES**

Se incluyen al principio del programa ya que se usan por las siguientes funciones de forma continuada.

**intenta\_OTA**

Comprueba si hay actualizaciones disponibles para el dispositivo, y actualiza automáticamente si se dá el caso.

**conecta\_wifi**

Conecta la placa a la red wifi para el envío-recepción de datos y consignas.

**conecta\_mqtt**

Comprueba si es posible la conexión mediante mqtt al programa de Node-RED.

**procesa\_mensaje**

Procesa todos los mensajes de entrada según el topic que presenten. Deserializa el JSON y asigna a las variables los valores que se correspondan según el mensaje.

**SerializeComplex**

Serializa los datos correspondientes con su topic a un formato JSON. Además, los publica.

Técnicas que se han utilizado para hacer robusto el sistema. Por ejemplo, cómo se maneja el fallo de un sensor, cómo se asegura el paso de mensajes, etc.

En la programación de los ESP8266 mediante el código en Arduino, se ha evitado los comandos tipo wait para hacer más robusta la toma de datos, ya que de esta forma el sistema es podrá recibir un comando de entrada en cualquier momento.

Se ha implementado un cortafuegos para los topics, mediante un diagrama if-else de forma que solo se deserialicen y lean los topics que encajan exactamente con el requerido, de lo contrario se podrá leer el mensaje “Error: topic desconocido”.

### Estudio del tiempo de ejecución de distintas fases de funcionamiento y el respectivo

### consumo de energía. Posibles mejoras u optimizaciones introducidas y análisis de su

### impacto.

Se ha conectado la placa ESP8266 a la estación de medida formada por un sensor de corriente, y se han tomado los datos desde el ini

### Librerías utilizadas

NODE-RED:

ARDUINO:

#include <ESP8266WiFi.h>

#include <ESP8266httpUpdate.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <ArduinoJson.hpp>

#include "DHTesp.h"

# RESULTADOS Y CONCLUSIONES

# MANUAL DE USUARIO

# LISTA Y DESCRIPCIÓN DE LOS FICHEROS ENTREGADOS

o Proyectos Arduino (\*.ino)

o Flujos Node-RED (\*.json)

ETC