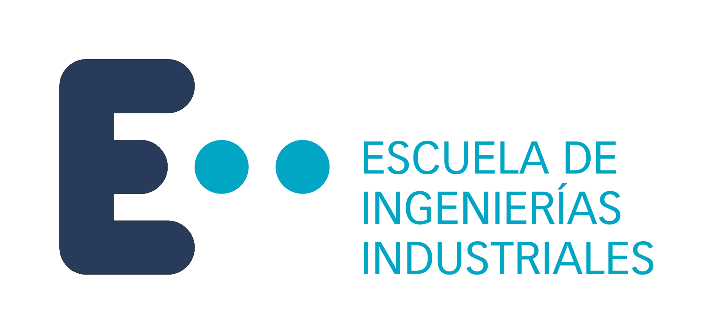
Grupo 1

GIERM

Fecha:



Memoria trabajo

Informática Industrial

Contenido

[Introducción 1](#_Toc61276931)

[Ampliaciones 1](#_Toc61276932)

[Desarrollo del proyecto 2](#_Toc61276933)

[Método de trabajo 2](#_Toc61276934)

[Separación IDE 2](#_Toc61276935)

[Repositorios 2](#_Toc61276936)

[Código IDE 2](#_Toc61276937)

[Global\_G1 2](#_Toc61276938)

[Global\_G1 3](#_Toc61276939)

[Config 4](#_Toc61276940)

[Datos 4](#_Toc61276941)

[Debug 4](#_Toc61276942)

# Introducción

Este proyecto consiste en la realización de un software capaz de realizar la comunicación entre un ESP8266 y un servidor de NodeRed mediante mqtt. Así mismo se pretende controlar diferentes actuadores asociados al ESP y recolectar información de los sensores que se conecten a este. Se gestionarán diferentes placas ESP, que deben identificarse a la hora de enviar y recibir información. Los datos recibidos deben visualizarse en el Dashboard de Node Red y las señales de control se enviarán desde él. Se almacenan y consultan datos de una base de datos de mongoDB. Además, se implementa el control de algunas características mediante telegram. El código del ESP se podrá actualizar mediante FOTA

## Ampliaciones

Como ampliación se ha decidido controlar un robot de tipo PIERO. Este robot se controla mediante un Arduino, que en este proyecto se conecta con la placa ESP mediante comunicación Serie. El flujo de la información se plantea en los dos sentidos. El robot envía al ESP información de los sensores que tiene incorporados. Por su parte, el ESP envía al robot diferentes señales de control para modificar su comportamiento o para controlarlo directamente. Esta ampliación se ha integrado también con Telegram.

Para programar el comportamiento del robot y su comunicación con la ESP se ha hecho uso de Matlab, concretamente de Simulink.

# Desarrollo del proyecto

## Método de trabajo

El trabajo se ha repartido entre los miembros del grupo, generalmente se ha asignado tareas concretas a cada uno, aunque puntualmente se han realizado tareas entre varios miembros. La organización se ha llevado a cabo en las sesiones de clase, utilizadas como momento de reunión para comunicar avances, dudas, propuestas y asignación de tareas. Se ha dejado constancia de la asignación de tareas en una nota de texto contenía tanto el historial de tareas realizadas como la lista de tareas pendientes.

## Separación IDE

El programa que utiliza la placa ESP se ha separado en diferentes programas que se utilizan a modo de librerías. Se permite así una mejor organización y claridad en la estructura del programa.

## Repositorios

Se ha utilizado la plataforma GitHub como repositorio de código, donde se ha depositado el código de la placa, el de node red y el de Arduino.

# Código IDE

El código que hace funcionar el ESP se ha dividido en diferentes programas que hacen de librerías para el programa principal, que hemos llamado Global\_G1.

## Global\_G1

Este programa se encarga de administrar las diferentes partes del proyecto y servir de nexo entre ellas, logrando así una mejor organización.

En primer lugar, se encarga de establecer la configuración inicial del ESP, del WiFi y del servidor Mqtt. A continuación, pasa a implementar las siguientes funcionalidades programadas.

Controla tanto el led como el switch, enviando el valor de estos mediante Mqtt cada vez que es necesario y gestionando el cambio de la intensidad del led para alcanzar el valor requerido.

Comprueba si se debe realizar una actualización FOTA, que se realiza periódicamente y llama a la función necesaria para ello.

Envía la estructura de datos programada en en Datos mediante mqtt, realizando un envío cada 5 min. El tiempo entre envíos es variable y puede controlarse mediante el dashboard.

Se encarga de enviar los datos de los sensores del robot cada segundo, también mediante mqtt.

## Global\_G1

Parte declarativa:

En primer lugar, se encuentran las librerías necesarias para la ejecución de este programa. A continuación, se declaran las variables globales necesarias y finalmente, se añade el resto de los programas que van a usarse, esto se hace tratándolos como librerías (funciones “.h”).

Setup:

Inicializa el ESP, declarando el modo de funcionamiento de determinados pines, la interrupción y realizando la conexión wifi.

Comprueba si hay actualizaciones FOTA

Inicializa la conexión con el servidor mqtt y se inicializan los topics de acuerdo a la configuración de config.

Inicializa el sensor.

Avisa que se ha activado el modo debug, para ello se ponde debug a true en config.

Loop:

Datos

Primero se generan los datos que se desean enviar por mqtt. Estos se organizan en una estructura creada en la función tomaDatos localizada en el programa Datos. Se serializan y se envían en el topic previsto.

Led

A continuación, se controla el funcionamiento del led. Para ello se comprueba si el led se ha modificado, lo que se indica mediante la variable ready\_led. Esta valdrá true solo cuando se haya detectado una pulsación del botón. Cuando esto ocurra, se registra el momento de la pulsación y se envían los datos haciendo uso de la función led\_mqtt(). Se procede de igual manera con el switch.

Además, también se controla el cambio de la intensidad del led, se consigue aplicando una condición que actualiza el valor de la intensidad cada cierto intervalo de tiempo. El intervalo se puede configurar desde nodeRed, modificando así la velocidad de actualización del led hasta que llegue al valor deseado.

Durante el envío de datos se trabaja con valores del led entre 0 y 100, mientras que para dar valores directamente a la placa se utilizan valores entre 0 y 1023.

## Config

Se encarga de declarar y gestionar todas las variables que tienen que ver con la configuración del programa, la conexión wifi, la distribución de pines en la placa, modo debug y configuración de topics.

Con respecto a los topics, se han preparado para ser flexibles. Se forma una cadena de caracteres a partir de diferentes variables de manera que cambiar un topic solo significa cambiar el valor de una variable. Se evita así tener que modificar todas las llamadas a dicho topic. En el Setup del programa global se llama a la función mqttTopics que se encarga de construir las cadenas de caracteres completas.

## Datos

Contiene las funciones para gestionar los datos que se manejan en el programa. Serializa\_datos\_JSON se encarga de crear la estructura de datos que contiene la información que se desea enviar por el topic “datos”. Esta información se asigna en la función tomaDatos.

Las funciones que se encargan de realizar el estado del switch y del led también se encuentran en este programa.

Finalmente está la función encargada de enviar la información de los sensores del robot.

## Debug

Esta función se emplea para mostrar mensajes por el puerto serie de la placa ESP con el objetivo de depurar las funcionalidades del programa. La ventaja de usar esta función en lugar de un simple Serial. Print() es que es posible decidir cuando mostrar mensajes y cuando no solo modificando una variable del paratado de config, “debug”.