

GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y  
SISTEMAS DE INFORMACIÓN

## **TRABAJO FIN DE GRADO**

***CONTROL DEL LLENADO DE UN TANQUE DE  
AGUA CON UN ARDUINO Y UN PLC  
CONECTADOS MEDIANTE UN GATEWAY  
IOT2020***

***DOCUMENTO 1- MEMORIA***

**Alumno:** López Barrios, Ibon  
**Director (1):** Oyarzabal Casquero, Oskar  
**Director (2):** Eguiraun Martínez, Harkaitz

**Curso:** 2017-2018

**Fecha:** Bilbao, 29/06/2018

## Resumen

Siemens, uno de los grandes fabricantes de PLCs industriales, ha sacado al mercado una familia de módulos, llamada IoT2000 que, integrándose dentro del ecosistema creado por la marca, posibilita el intercambio de datos con otros sistemas electrónicos de control de bajo coste y/o código abierto, facilitando así, el aporte de soluciones por parte de terceros.

En el caso que nos ocupa, debemos proporcionar una solución discreta a un proceso de llenado de tanques de líquido, que ya dispone de un sistema de control basado en PLCs de la marca Siemens, con la mínima intervención posible sobre el sistema ya en funcionamiento.

Como planteamiento inicial se le ha propuesto al cliente la adquisición de un módulo IoT2020 y un módulo Arduino como sistema externo de registro de datos y control, obteniéndose así un sistema que lea, procese y envíe automáticamente la información de los datos de un proceso a un servidor remoto a través del uso de protocolos de comunicación y la conexión a Internet, para facilitar la actualización de cualquier planta a la Industria 4.0 mediante el Internet de las Cosas.

## Palabras clave:

Internet de las Cosas, IoT, IoT2020, Sensores, Arduino, PLC, MQTT, Sistema de Control, Industria 4.0, Siemens.

## Abstract

Siemens, one of the largest manufacturers of industrial PLCs, has brought to the market a family of modules, called IoT2000 that, integrating within the ecosystem created by the brand, enables the exchange of data with other electronic control systems of low cost and/or open source, thus facilitating the contribution of solutions by third parties.

In our case, we must provide a discreet solution to a liquid tank filling process, which already has a control system based on PLCs from the Siemens brand, with the minimum possible intervention on the system already in operation.

As an initial approach, the client has been proposed to purchase an IoT2020 module and an Arduino module as an external data and control system, thus obtaining a system that automatically reads, processes and sends the data information of a process to a remote server through the use of communication protocols and Internet connection, to facilitate the upgrade of any plant to Industry 4.0 through the Internet of Things.

## Keywords:

Internet of Things, IoT, IoT2020, Sensors, Arduino, PLC, MQTT, Control System, Industry 4.0, Siemens.

## Laburpena

Gaur egungo PLC fabrikatzaile handienetarikoa bat Siemens da. Gainera, IoT2000 deitzen den modulu-familia bat merkaturatu du. Honek, kostu txikiko eta/edo iturburu irekia duten beste sistema elektronikoen batzuekin datuak elkar trukatzeko aukera ematen du.

Kasu honetan, konponbide diskretu bat eskaini behar diogu likido depositu bat betetzeko prozesuari. Depositu horrek jada badu PLCn oinarritutako Siemens markako kontrol sistema bat. Konponbide honek ahalik eta gutxien eragin behar du martxan dagoen sistemarengan.

Erabiltzaileari IoT2020 bat erabiltzea proposatu zaio eta Arduino modulo bat kanpo sistema bezala erabiltzeko. Horrela, informazioa irakurtzen eta bidaltzen duen sistema bat lortzen da. Informazioa kanpoko zerbitzaile bati bidaltzen zaio komunikazio protokolo eta Internet konexio batetik.

## Gako-hitzak:

Gauzen Internet-a, IoT, IoT2020, Sentsoreak, Arduino, PLC, MQTT, Kontrol Sistema, Industria 4.0, Siemens.

## Índice

<b>1.</b>	<b>INTRODUCCIÓN</b> .....	<b>10</b>
<b>2.</b>	<b>OBJETIVOS Y ALCANCE DEL TRABAJO</b> .....	<b>11</b>
<b>3.</b>	<b>BENEFICIOS QUE APORTA EL TRABAJO</b> .....	<b>12</b>
<b>4.</b>	<b>DESCRIPCIÓN DE REQUERIMIENTOS</b> .....	<b>13</b>
<b>5.</b>	<b>ANÁLISIS DEL ESTADO DEL ARTE</b> .....	<b>14</b>
<b>6.</b>	<b>ANÁLISIS DE RIESGOS</b> .....	<b>16</b>
6.1.	APAGÓN DE LUZ .....	16
6.2.	MALA PLANIFICACIÓN DEL TIEMPO INVERTIDO EN EL PROYECTO .....	16
6.3.	BAJA POR FUERZA MAYOR.....	17
6.4.	PÉRDIDA DE DATOS .....	17
6.5.	MAL FUNCIONAMIENTO DEL ORDENADOR.....	17
6.6.	ERROR DE CONEXIÓN A INTERNET.....	18
6.7.	MALA FINANCIACIÓN.....	18
6.8.	ROTURA DE ELEMENTOS DE LA MAQUETA .....	18
<b>7.</b>	<b>DISEÑO DE ALTO NIVEL</b> .....	<b>19</b>
<b>8.</b>	<b>DESCRIPCIÓN DE TAREAS</b> .....	<b>21</b>
<b>9.</b>	<b>DIAGRAMA DE GANTT</b> .....	<b>38</b>
<b>10.</b>	<b>DISEÑO DE BAJO NIVEL</b> .....	<b>40</b>
10.1.	GATEWAY IOT2020.....	40
10.1.1.	<i>Node-RED</i> .....	43
10.1.2.	<i>MQTT</i> .....	44
10.1.3.	<i>Broker MQTT (Mosca)</i> .....	45
10.1.4.	<i>S7</i> .....	46
10.1.5.	<i>Programa principal node-RED</i> .....	46
10.2.	GESTIÓN DE SENSORES.....	51
10.2.1.	<i>Arduino MEGA</i> .....	51
10.2.2.	<i>Sensor temperatura (LM35DZ)</i> .....	52
10.2.3.	<i>Sensor de distancia (OsiSense XX918A3C2M12)</i> .....	55
10.2.4.	<i>Sensor de flujo (YF-S201)</i> .....	60
10.2.5.	<i>Envío de datos</i> .....	62
10.3.	GESTIÓN ACTUADORES.....	67
10.3.1.	<i>SIMATIC S7-1200</i> .....	67
10.3.2.	<i>TIA Portal (STEP 7 Basic)</i> .....	68
10.3.3.	<i>Actuadores</i> .....	69
10.4.	PLANTA.....	70
<b>11.</b>	<b>DESCRIPCIÓN DE LOS RESULTADOS</b> .....	<b>74</b>
<b>12.</b>	<b>DESCRIPCIÓN DEL PRESUPUESTO</b> .....	<b>75</b>
<b>13.</b>	<b>CONCLUSIONES</b> .....	<b>77</b>
<b>14.</b>	<b>BIBLIOGRAFÍA</b> .....	<b>78</b>

## Lista de figuras

Figura 1: LM35.....	6
Figura 2: XX918A3C2M12.....	6
Figura 3: YF-S201 .....	7
Figura 4: Diagrama de bloques IoT Boiler Controller .....	14
Figura 5: SIMATIC IoT2020 .....	41
Figura 6: Arquitectura Yocto.....	42
Figura 7: Panel de control.....	47
Figura 8: Flujo visualización de los valores de los sensores .....	48
Figura 9: Flujo comprobación del proceso.....	48
Figura 10: Flujo principal .....	49
Figura 11: Flujo parada emergencia .....	49
Figura 12: Flujo publicación datos ThingSpeak.....	49
Figura 13: Gráficos ThingSpeak.....	50
Figura 14: Arduino MEGA.....	51
Figura 15: LM35DZ .....	52
Figura 16: OsiSense XX918A3C2M12 .....	55
Figura 17: YF-S201 .....	60
Figura 18: Arduino Shield .....	63
Figura 19: Conexiones Arduino Shield .....	64
Figura 20: SIMATIC S7-1200.....	67
Figura 21: Panel de control.....	58
Figura 22: Gráficos ThingSpeak.....	59

## Lista de tablas

Tabla 1: Formación Arduino .....	23
Tabla 2: Elección de sensores.....	23
Tabla 3: Lectura sensores .....	23
Tabla 4: Integración sensores .....	24
Tabla 5: WiFi.....	24
Tabla 6: Envío de datos MQTT .....	24
Tabla 7: Testeo bloque Arduino .....	25
Tabla 8: Reuniones con tutor bloque Arduino.....	25
Tabla 9: Formación IoT2020 .....	25
Tabla 10: Instalación sistema operativo Yocto.....	26
Tabla 11: Configuración IoT2020 .....	26
Tabla 12: Configuración WiFi.....	26
Tabla 13: Probar placa microcontrolador .....	27
Tabla 14: Eclipse IDE.....	27
Tabla 15: Formación Node-RED.....	27
Tabla 16: Módulo S7-Node-RED .....	28
Tabla 17: Módulo MQTT-Node-RED.....	28
Tabla 18: Módulo MQTT-Broker-Node-RED .....	29
Tabla 19: Programa de control .....	29
Tabla 20: Elección Broker-MQTT-node-RED .....	30
Tabla 21: Panel de Control en Node-RED .....	30
Tabla 22: Testeo bloque IoT2020.....	30
Tabla 23: Reuniones con tutor bloque IoT2020 .....	31
Tabla 24: Formación PLC .....	31
Tabla 25: Preparar entorno de desarrollo (Configuración) de TIA Portal.....	31
Tabla 26: Conectar actuadores .....	32
Tabla 27: Testeo bloque PLC .....	32
Tabla 28: Reuniones con tutor bloque PLC.....	32
Tabla 29: Formación planta .....	33
Tabla 30: Diseño de piezas .....	33
Tabla 31: Impresión piezas.....	33
Tabla 32: Compra de material .....	34
Tabla 33: Montar maqueta.....	34

Tabla 34: Reuniones con tutor y técnico laboratorio bloque planta .....	34
Tabla 35: Análisis de objetivos .....	35
Tabla 36: Reuniones con el tutor .....	35
Tabla 37: Documentación.....	35
Tabla 38: Revisión de la memoria .....	36
Tabla 39: Especificaciones técnicas SIMATIC IoT2020 .....	42
Tabla 40: Características Arduino MEGA .....	52
Tabla 41: Descripción pines LM35DZ.....	54
Tabla 42: Conexiones Arduino LM35DZ .....	55
Tabla 43: Estado del LED OsiSense XX918A3C2M12.....	56
Tabla 44: Descripción pines OsiSense XX918A3C2M12 .....	57
Tabla 45: Conexión Arduino OsiSense XX918A3C2M12 .....	59
Tabla 46: Descripción pines YF-S201.....	61
Tabla 47: Conexiones Arduino YF-S201 .....	62
Tabla 48: Características SIMATIC S7-1200 .....	68



## Lista de Abreviaturas

<i>IoT</i>	<i>Internet of Things</i>
<i>PLC</i>	<i>Controlador Lógico Programable</i>
<i>M2M</i>	<i>Machine to Machine</i>
<i>I2C</i>	<i>Inter-Integrated Circuit</i>
<i>HTML</i>	<i>Hypertext Markup Language</i>
<i>JS</i>	<i>JavaScript</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>MQTT</i>	<i>Message Queuing Telemetry Transport</i>
<i>TFG</i>	<i>Trabajo Fin de Grado</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>MVC</i>	<i>Model-View-Controller</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>TTL</i>	<i>Transistor-Transistor Logic</i>
<i>IO</i>	<i>Input/Output</i>
<i>COM</i>	<i>Puerto serie de una computadora</i>
<i>IEEE</i>	<i>Institute of Electrical and Electronics Engineers</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>SSH</i>	<i>Secure Shell</i>
<i>UPV/EHU</i>	<i>Universidad del País Vasco/Euskal Herriko Unibertsitatea</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>NPM</i>	<i>Node Package Manager</i>
<i>OPKG</i>	<i>Open PacKaGe Management</i>

## 1. Introducción

La llamada Industria 4.0 ha provocado una evolución en la racionalización de los procesos de producción y una descentralización de los centros productivos. Paralelamente a esta revolución, el desarrollo de la llamada Internet de las cosas (IoT en adelante) ha propiciado la acumulación de todo tipo de datos que pueden emplearse, entre otras cosas, en la optimización de los procesos productivos. Todos estos avances no pueden tratarse de un modo aislado, estando todos ellos íntimamente relacionados (Industria 4.0, IoT, BigData, etc.) y dando lugar, también al desarrollo de nuevas tecnologías de bajo coste que proporcionan soluciones, hasta hace poco impensables, a problemas habituales dentro del ámbito industrial.

Dentro del proyecto que nos ocupa, tenemos, por una lado, los sistemas productivos basados en PLCs industriales que, siendo equipos robustos y de eficacia probada, tienen un coste elevado que los hace poco interesantes a la hora de aportar soluciones en los casos más simples, y, por otro, una serie de controladores discretos de bajo coste, basados en electrónica programable de código abierto que, si bien son ideales para resolver necesidades sencillas, no son muy apropiados para ser incorporados dentro del ámbito industrial.

Como fruto de esta evolución tecnológica, recientemente uno de los grandes fabricantes de PLCs industriales, Siemens, ha sacado al mercado una familia de módulos, llamada IoT2000 que, integrándose dentro del ecosistema creado por la marca; posibilita el intercambio de datos con otros sistemas electrónicos de control de bajo coste y/o código abierto, facilitando así el aporte de soluciones por parte de terceros.

En el caso que nos ocupa, debemos proporcionar una solución discreta a un proceso de llenado de tanques de líquido que ya dispone de un sistema de control basado en PLCs de la marca Siemens, con la mínima intervención posible sobre el sistema ya en funcionamiento.

Como planteamiento inicial al cliente, se ha propuesto la adquisición de un modelo IoT y un módulo Arduino como sistema externo de registro de datos y control.

## 2. Objetivos y alcance del trabajo

El objetivo principal de este proyecto es el control de llenado de un tanque de agua en un contexto donde los nodos encargados de leer los sensores, de escribir los actuadores y controlar y monitorizar el proceso están distribuidos. Para lograr este objetivo, es preciso plantear los siguientes objetivos secundarios:

- Programar los distintos sensores (temperatura, distancia y caudal) para un correcto funcionamiento.
- Configurar el Gateway (IoT2020) para la gestión del flujo del programa, además también se ha configurado un servidor en el propio Gateway para la subida de datos de los sensores.
- Además del servidor comentado previamente en el punto anterior, se utiliza otro repositorio para almacenar los datos obtenidos con los sensores en otro servidor proporcionado por ThingSpeak.
- Crear un panel de control para controlar el sistema en todo momento, además, éste también incorpora pequeños gráficos para controlar los valores de los distintos sensores.
- Configurar y conectar los distintos actuadores (bomba 1, bomba 2, resistencia y agitador) en el PLC.
- Diseñar soportes para los sensores y elementos de la maqueta para una posterior impresión en una máquina 3D.
- Montaje de la maqueta para lograr que el proyecto sea realista y así, demostrar que todo lo propuesto en el proyecto se ha llevado a cabo.

### 3. Beneficios que aporta el trabajo

Desde que somos capaces de conectar cosas por cables nos enfrentamos a dos problemas: la fortaleza de la conexión y la facilidad de mantenimiento. Tras el paso de los años, las comunicaciones serie para las industrias basadas en lazos de corriente tienen la ventaja de la simplicidad de montaje (dos hilos) y la facilidad de mantenimiento. Frente a eso tenemos redes multipunto con infinitud de protocolos y tramas complicadas por todos lados que un electricista de planta no va a tocar y, por tanto, habrá que llamar a un ingeniero. Naturalmente, las nuevas redes posibilitan enviar gran cantidad de información de un punto a otro; pero, ¿en que casos se quieren enviar gigabytes de datos en una planta de montaje? Normalmente se quieren saber temperaturas, presiones y velocidades, y, para eso, con una vieja red basada en lazo de corriente, es suficiente.

En este contexto, como los fabricantes de PLCs lo que quieren es ganar dinero y sus clientes no quieren gastarlo, los PLCs siguen teniendo disponibles interfaces de comunicación basados en protocolos sencillos (por ejemplo, RS485) para poder ser integrados en las instalaciones viejas. A su vez, los PLCs actuales vienen con redes nuevas, ya que, naturalmente, hay que seguir evolucionando. El resultado es que todos (los ingenieros viejos y los nuevos) podemos seguir aportando soluciones a la industria, cada uno con lo que sabe.

No obstante, hoy en día, tenemos un nuevo tipo de personas, que podríamos denominar “solucionadores”. Los "solucionadores" son gente que, sabiendo mucho más que el electricista de planta, pero sin los conocimientos específicos sobre las tecnologías de una determinada marca, son capaces de aportar soluciones económicas a un problema determinado. Estos “solucionadores” pueden sacar a clientes modestos de la cuenta de resultados de los grandes fabricantes como Siemens, con el consiguiente perjuicio económico estos últimos.

Por ello, dispositivos como el IoT2020 presentan un gran beneficio técnico y económico porque permiten conectar los "inventos", basados en plataformas de bajo coste como Arduino, creados por los "solucionadores" a los productos de los grandes fabricantes de PLCs.

## 4. Descripción de requerimientos

La planta de este proyecto ha sido modelizada a través de una maqueta que consta de una pecera dividida en dos por un cristal transparente. En cada lateral hay una pieza que sirve de soporte para la sujeción de diferentes tubos mediante los cuales se realizara el paso del agua, así como diferentes sensores: el sensor de distancia y el sensor de flujo.

Además, en ambas partes del tanque, concretamente en el fondo, se puede encontrar un motor para la circulación del agua, que van sujetos a los tubos anteriormente citados.

En el lado en el que se encuentra el sensor de flujo hay una resistencia que actúa a modo de calentador del agua, así como un agitador para remover el agua por todo el tanque. A su vez, para la medición de temperatura, se requiere de un sensor de temperatura.

Además, se han incluido dos nodos de control distribuido para la activación, mediante un PLC de Siemens, y la sonorización a través de un Arduino MEGA. Para el envío de datos a través del Arduino se ha colocado un Ethernet Shield para que éste tenga salida a internet.

Por ultimo, para integrar los dos nodos anteriores dentro de un sistema de control distribuido, se ha utilizado un IoT2020.

## 5. Análisis del estado del arte

Mientras se hacía este trabajo, se han encontrado dos proyectos que también utilizan un IoT2000 y analizan las características de éste y algunas funcionalidades del mismo. Uno trata sobre controlar una caldera utilizando un IoT2020 y el otro la puesta en marcha del IoT2040. El primero es un proyecto más práctico y el segundo más teórico.

- “IoT Boiler Controller using Siemens IoT2020”

En este trabajo se puede ver como se automatiza el control de una caldera industrial.

El siguiente diagrama de bloques muestra los principales componentes del proyecto. En el centro del diseño se encuentra el IoT2020 que es responsable de comunicar y controlar todos los dispositivos periféricos. El sistema contiene una bomba controlada por relé para pasar el combustible, un sensor de flujo para controlar la cantidad de combustible que hay que pasar y un sensor de temperatura adicional que permite ampliar la funcionalidad para demostraciones de seguridad y corte, por ejemplo, que cuando llegue a una temperatura se pare el paso de combustible.

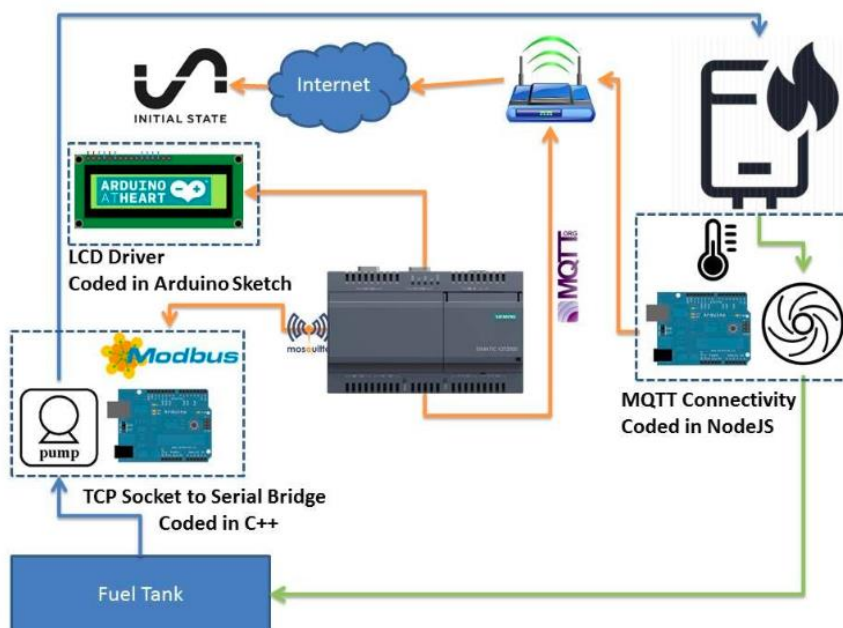


Figura 1: Diagrama de bloques IoT Boiler Controller

La comunicación entre el Gateway y los diversos componentes está habilitada por protocolos como ModBUS/TLV, MQTT, REST y TCP/IP.

- Puesta en marcha de un sistema de automatización Siemens SIMATIC IoT2040

Describe la puesta en marcha del sistema, puesta en marcha que comprende los siguientes aspectos:

- Instalación del SO en la tarjeta SD
- Configuración inicial del sistema (por ejemplo: configuración de red)
- Configuración del entorno de desarrollo mediante Eclipse
- Desarrollo de 4 programas para probar las capacidades del IoT2020

## 6. Análisis de riesgos

Todo proyecto conlleva una serie de riesgos que deben ser identificados de manera previa para así poder evitarlos, ya que si esto no se llevara a cabo habría que actuar de manera eficaz para intentar solucionarlos.

En esta sección se detallarán los posibles riesgos detectados, la probabilidad de que ocurran, el impacto que supondría en el proyecto, así como las consecuencias que causarían. Se detallarán también las medidas de prevención tomadas, así como el plan de contingencia que se aplicará para arreglar el problema.

### 6.1. Apagón de luz

- **Descripción:** Es posible que en algún momento dado haya un apagón de luz.
- **Probabilidad:** Baja.
- **Impacto:** Muy alto.
- **Consecuencia:** Interrupción total del proceso.
- **Prevención:** Tener baterías de emergencias.
- **Plan de contingencia:** programar el proceso para que se pare todo al instante y no haya pérdida de datos.

### 6.2. Mala planificación del tiempo invertido en el proyecto

- **Descripción:** Planificar de manera incorrecta los tiempos para la realización del proyecto.
- **Probabilidad:** Alta.
- **Impacto:** Alto.
- **Consecuencia:** Una mala planificación del proyecto retrasaría todos los procesos y con ello la fecha de entrega se vería afectada.
- **Prevención:** Establecer plazos de entrega que se adecuen al usuario.
- **Plan de contingencia:** Aumentar horas de empleo en el proyecto.



### 6.3. Baja por fuerza mayor

- **Descripción:** No poder trabajar debido a una serie de causas por fuerza mayor.
- **Probabilidad:** Baja.
- **Impacto:** Media.
- **Consecuencia:** Pérdida de días lectivos.
- **Prevención:** No existe una prevención contra este tipo de problemas.
- **Plan de contingencia:** Se aumentarán las horas de trabajo de otros días a fin de compensar las horas perdidas.

### 6.4. Pérdida de datos

- **Descripción:** Es posible que los datos sufran algún tipo de corrupción o pérdida que imposibilite su lectura y trabajar con ellos.
- **Probabilidad:** Baja.
- **Impacto:** Baja.
- **Consecuencia:** Pérdida parcial del trabajo.
- **Prevención:** Tener copias de seguridad.
- **Plan de contingencia:** En caso de que produjera este accidente, se procedería a restaurar una copia de seguridad en el lugar afectado para seguir trabajando.

### 6.5. Mal funcionamiento del ordenador

- **Descripción:** Es posible que el entorno de trabajo sufra daños y deje de funcionar.
- **Probabilidad:** Baja.
- **Impacto:** Medio.
- **Consecuencia:** Pérdida del entorno de trabajo.
- **Prevención:** Tener más dispositivos para poder seguir trabajando en caso de que un equipo falle.
- **Plan de contingencia:** Usar otro equipo para seguir con el trabajo hasta que el principal esté en condiciones de volver a ser usado.

#### 6.6. Error de conexión a internet

- **Descripción:** Pérdidas de conexión a internet.
- **Probabilidad:** Alta.
- **Impacto:** Medio.
- **Consecuencia:** Aumenta el tiempo de espera.
- **Prevención:** No existe una prevención contra este tipo de problemas pues son aleatorios.
- **Plan de contingencia:** Utilizar protocolos de comunicación a internet de bajo consumo.

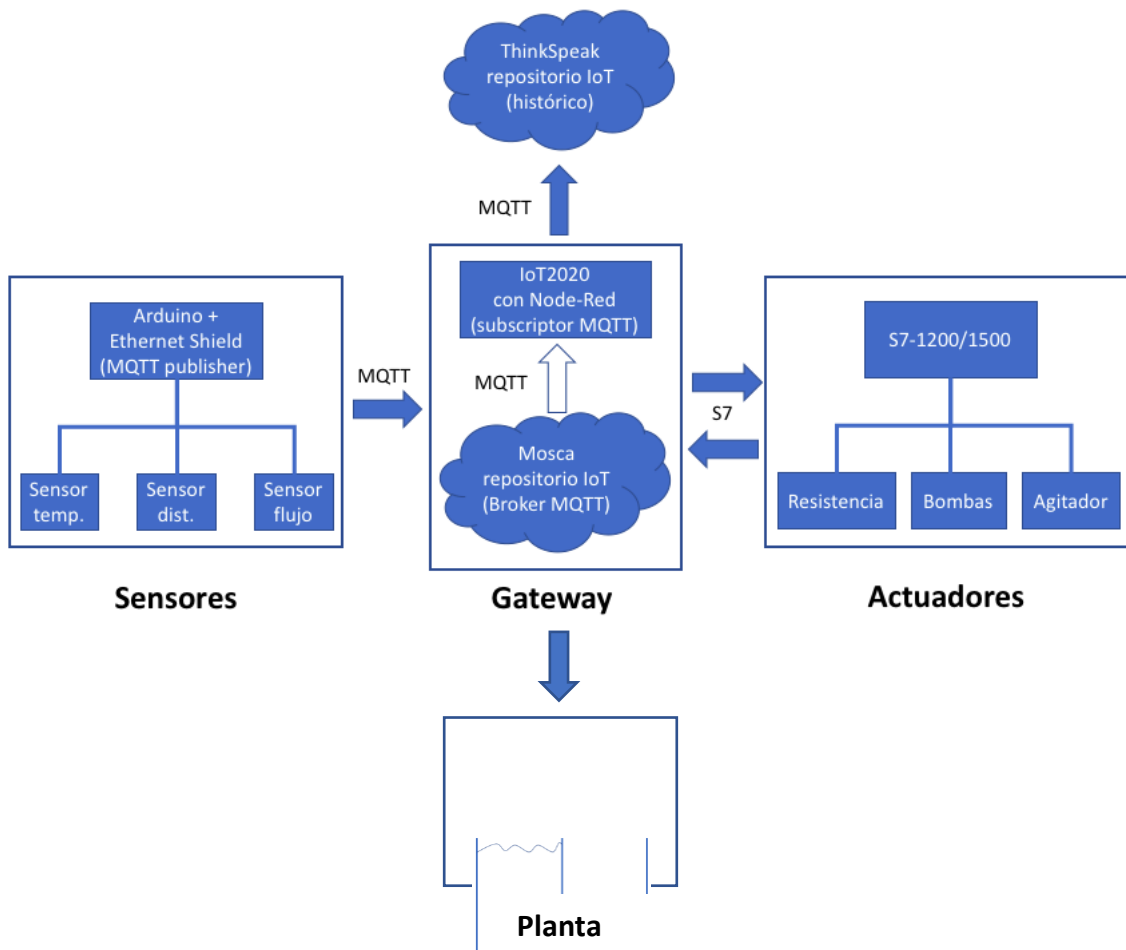
#### 6.7. Mala financiación

- **Descripción:** Un presupuesto bajo conlleva comprar elementos de baja calidad y que los errores más comunes aumenten.
- **Probabilidad:** Media.
- **Impacto:** Bajo.
- **Consecuencia:** Errores comunes.
- **Prevención:** Escoger componentes con buenas referencias.
- **Plan de contingencia:** No se puede solventar ninguna contingencia ya que es aleatorio.

#### 6.8. Rotura de elementos de la maqueta

- **Descripción:** Que algún elemento de la maqueta se rompa o deje de funcionar.
- **Probabilidad:** Media.
- **Impacto:** Medio.
- **Consecuencia:** Paralización del proceso.
- **Prevención:** tener más cantidad de piezas.
- **Plan de contingencia:** Cuidar la maqueta.

## 7. Diseño de alto nivel



Para desarrollar este proyecto de forma ordenada se ha dividido éste en 3 bloques. Éstos son los siguientes: gestión de sensores, el Gateway IoT2020 y gestión de actuadores.

Los datos que se obtienen de los sensores, es decir, la temperatura, la distancia y el caudal, son enviados mediante el protocolo MQTT al Gateway IoT2020. Éste es un protocolo usado para la comunicación M2M (Machine to machine) en el que el Broker envía mensajes en tiempo real a todos sus subscriptores al recibir una publicación nueva. En este proyecto se han utilizado dos servidores en los que se han instalado dos Broker, uno en un servidor remoto de thingspeak.com y otro en un servidor local instalado en el Gateway IoT2020. Para conseguir los datos de los sensores se ha utilizado un Arduino Mega y para la posterior publicación mediante MQTT se le ha añadido una placa Ethernet al Arduino.

Una vez los datos son enviados a los distintos servidores es necesario tener un suscriptor MQTT para poder gestionar los datos mandados, para ello se ha instalado una herramienta de programación basada en flujo llamada node-red en el Gateway IoT2020. Node-RED es el pilar central del proyecto en el que se gestiona todo el flujo, desde la recepción de datos de los sensores hasta la activación de los distintos actuadores, todo esto pasando por la propia programación (por bloques) de proyecto.

Para la recepción de datos a tiempo real se ha utilizado el servidor Mosca, que está instalado en el propio IoT2020. El Arduino se encarga de enviar los datos a dicho servidor. Una vez los datos llegan a node-red, éstos se envían al servidor remoto thingspeak.com para guardar un historial y poder ver esos datos de forma gráfica. Además, gracias a esta herramienta se ha creado el panel de control del proyecto.

Éste se divide en dos partes, una para la visualización de los valores de los sensores para verificar que los datos obtenidos son lógicos y por otra parte el propio panel de control para gestionar el proyecto. En esta parte se podrá activar un pre proceso para comprobar que todos los componentes del proyecto están preparados para llevar a cabo la tarea. Desde este panel también se dará comienzo a la automatización del sistema pudiendo personalizar los parámetros que se deben cumplir.

Para la gestión de actuadores se ha utilizado el programa Tia Portal en el que se ha creado una tabla de observación para ver que actuadores (bomba 1, bomba 2, resistencia y calentador) están activos y cuales no. Previamente se han conectado de manera ordenada todos los actuadores al PLC.

Es importante subrayar que es necesario que todos los equipos utilizados en este proyecto deben estar en la misma red local para poder intercambiar datos entre ellos.

Por último, en la planta se han colocado los sensores y actuadores mencionados anteriormente y con la ayuda de node-RED se ha conseguido que dos nodos que no están unidos por ningún lado, es decir, que está aislado el uno del otro, interactúen entre ellos. Además, para que la planta sea una maqueta didáctica se han diseñado e impreso dos piezas 3D para la sujeción de algunos sensores y de los tubos por donde pasa el agua.

## 8. Descripción de tareas

A lo largo del proyecto se han ido haciendo pequeñas tareas que en su conjunto han creado el trabajo entero. Estas tareas se han dividido en 5 bloques para llevar un orden y así conseguir un trabajo de calidad. A continuación, se van a enumerar todos los bloques con sus respectivas tareas.

El primer bloque es todo lo que tenga que ver con Arduino. Las tareas que se han llevado a cabo en este ámbito han sido las siguientes:

- Formación
- Elección de sensores
- Lectura sensores
- Integración sensores
- WiFi
- Envío de datos MQTT
- Testeo bloque Arduino
- Reuniones con tutor bloque Arduino

El segundo bloque tiene que ver con el IoT2020 y se han realizado las siguientes tareas:

- Formación
- Instalación sistema operativo Yocto
- Configuración IoT2020
- Configuración WiFi
- Probar placa microcontrolador
- Eclipse IDE
- Formación Node-RED
- Módulo S7-Node-RED
- Módulo MQTT-Node-RED
- Módulo MQTT-Broker-Node-RED
- Programa de control
- Elección Broker-MQTT-Node-RED
- Panel de Control en Node-RED
- Testeo bloque IoT2020

- Reuniones con tutor bloque IoT2020

El tercer bloque es el PLC con las siguientes tareas:

- Formación
- Preparar entorno de desarrollo (Configuración) de TIA Portal
- Conectar actuadores
- Testeo bloque PLC
- Reuniones con tutor bloque PLC

En cuarto lugar, está todo lo que tenga que ver con la planta (maqueta):

- Formación
- Diseño de piezas
- Impresión piezas
- Compra de material
- Montar maqueta
- Reuniones con tutor y técnico laboratorio bloque planta

Y por último, cómo se ha gestionado el proyecto:

- Análisis de objetivos
- Reuniones con el tutor
- Documentación
- Revisión de la memoria

Una vez comentadas todas las tareas que se han llevado a cabo, se va a entrar más en profundidad y se va a explicar cada una de forma más significativa.

## Tareas Arduino:

- Formación

Duración	50 horas
Encargado	Ibon López
Descripción	Formación para desarrollar programas Arduino y para leer los sensores
Entradas	Documentación e internet
Recursos necesarios	Internet
Salidas/Entregables	Conocimiento

Tabla 1: Formación Arduino

- Elección de sensores

Duración	5 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Elegir los sensores idóneos para el proyecto
Entradas	Señales analógicas y digitales
Recursos necesarios	Arduino, sensores y ordenador
Salidas/Entregables	Temperatura, distancia y flujo

Tabla 2: Elección de sensores

- Lectura sensores

Duración	5 horas
Encargado	Ibon López
Descripción	Leer los sensores utilizados en el proyecto
Entradas	Señales analógicas y digitales
Recursos necesarios	Arduino, sensores y ordenador
Salidas/Entregables	Temperatura, distancia y flujo

Tabla 3: Lectura sensores

- Integración sensores

Duración	5 horas
Encargado	Ibon López
Descripción	Integrar los tres sensores en un mismo programa de Arduino para poder enviar los datos en tiempo real
Entradas	Temperatura, distancia y flujo
Recursos necesarios	Arduino, sensores y ordenador
Salidas/Entregables	-

Tabla 4: Integración sensores

- WiFi

Duración	3 horas
Encargado	Ibon López
Descripción	Decidir cómo dar conexión a la red al Arduino
Entradas	-
Recursos necesarios	Placa Wifi, placa Ethernet, Arduino y ordenador
Salidas/Entregables	Conexión a la red

Tabla 5: WiFi

- Envío de datos MQTT

Duración	5 horas
Encargado	Ibon López
Descripción	Enviar los datos obtenidos con los sensores mediante protocolo MQTT
Entradas	Temperatura, distancia y flujo
Recursos necesarios	Arduino e internet
Salidas/Entregables	JSON{Temperatura, distancia y flujo}

Tabla 6: Envío de datos MQT



- Testeo bloque Arduino

Duración	5 horas
Encargado	Ibon López
Descripción	Comprobar que el bloque Arduino funciona perfectamente
Entradas	-
Recursos necesarios	Arduino, internet, sensores y ordenador
Salidas/Entregables	Calidad

Tabla 7: Testeo bloque Arduino

- Reuniones con tutor bloque Arduino

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Reuniones con el tutor para arreglar y mejorar el bloque Arduino
Entradas	Errores
Recursos necesarios	-
Salidas/Entregables	Calidad

Tabla 8: Reuniones con tutor bloque Arduino

Tareas IoT2020:

- Formación

Duración	60 horas
Encargado	Ibon López
Descripción	Formación en el bloque IoT2020
Entradas	-
Recursos necesarios	Documentación e internet
Salidas/Entregables	Conocimiento

Tabla 9: Formación IoT2020

- Instalación sistema operativo Yocto

Duración	3 horas
Encargado	Ibon López
Descripción	Instalar sistema operativo al IoT2020
Entradas	-
Recursos necesarios	IoT2020, imagen Yocto e instalador de sistemas operativos
Salidas/Entregables	Acceso al IoT2020

Tabla 10: Instalación sistema operativo Yocto

- Configuración IoT2020

Duración	5 horas
Encargado	Ibon López
Descripción	Configurar el IoT2020
Entradas	-
Recursos necesarios	IoT2020
Salidas/Entregables	-

Tabla 11: Configuración IoT2020

- Configuración WiFi

Duración	3 horas
Encargado	Ibon López
Descripción	Configurar el IoT2020 para que tenga salida a red eduroam
Entradas	-
Recursos necesarios	IoT2020
Salidas/Entregables	Conexión a internet

Tabla 12: Configuración WiFi

- Probar placa microcontrolador

Duración	5 horas
Encargado	Ibon López
Descripción	Compilar programas en el IoT2020 y que el microcontrolador que trae incorporado se encargue de ejecutarlo
Entradas	Programa Arduino
Recursos necesarios	IoT2020
Salidas/Entregables	-

Tabla 13: Probar placa microcontrolador

- Eclipse IDE

Duración	5 horas
Encargado	Ibon López
Descripción	Configurar el Eclipse IDE para ejecutar programas desde el IoT2020
Entradas	-
Recursos necesarios	Eclipse IDE y IoT2020
Salidas/Entregables	-

Tabla 14: Eclipse IDE

- Formación Node-RED

Duración	10 horas
Encargado	Ibon López
Descripción	Aprender a utilizar node-red
Entradas	Documentación e internet
Recursos necesarios	Internet
Salidas/Entregables	Conocimiento

Tabla 15: Formación Node-RED

- Módulo S7-Node-RED

Duración	5 horas
Encargado	Ibon López
Descripción	Instalar el módulo S7 en node-red para comunicarse con el PLC. Además de aprender a utilizarlo.
Entradas	-
Recursos necesarios	Iot2020, node-red e internet
Salidas/Entregables	Conectar IoT2020 con PLC

Tabla 16: Módulo S7-Node-RED

- Módulo MQTT-Node-RED

Duración	5 horas
Encargado	Ibon López
Descripción	Aprender a publicar y suscribirse en un Broker MQTT
Entradas	-
Recursos necesarios	Iot2020, node-red e internet
Salidas/Entregables	Publicar y recibir datos

Tabla 17: Módulo MQTT-Node-RED

- Módulo MQTT-Broker-Node-RED

Duración	5 horas
Encargado	Ibon López
Descripción	Instalar el módulo MQTT-Broker en node-red para comunicarse con un servidor MQTT. Además de aprender a utilizarlo.
Entradas	-
Recursos necesarios	Iot2020, node-red e internet
Salidas/Entregables	Conectar IoT2020 con MQTT Broker

Tabla 18: Módulo MQTT-Broker-Node-RED

- Programa de control

Duración	10 horas
Encargado	Ibon López
Descripción	Automatizar el llenado del tanque de agua y comprobar que todo funciona perfectamente
Entradas	-
Recursos necesarios	IoT2020, node-red, PLC, Arduino, Sensores e internet
Salidas/Entregables	Automatización del proceso

Tabla 19: Programa de control

- Elección Broker-MQTT-Node-RED

Duración	3 horas
Encargado	Ibon López
Descripción	Elegir el Broker MQTT que se va a utilizar en el proyecto como repositorio de datos
Entradas	Servidores Broker MQTT
Recursos necesarios	-
Salidas/Entregables	Broker MQTT

Tabla 20: Elección Broker-MQTT-node-RED

- Panel de Control en Node-RED

Duración	5 horas
Encargado	Ibon López
Descripción	Crear y diseñar un panel de control para que todo el flujo se gestione desde ahí
Entradas	-
Recursos necesarios	Node-red, Iot2020 e internet
Salidas/Entregables	Panel de control

Tabla 21: Panel de Control en Node-RED

- Testeo bloque IoT2020

Duración	10 horas
Encargado	Ibon López
Descripción	Comprobar que la automatización del proceso funciona perfectamente y no hay ningún error
Entradas	-
Recursos necesarios	-
Salidas/Entregables	Calidad

Tabla 22: Testeo bloque IoT2020

- Reuniones con tutor bloque IoT2020

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Reuniones para corregir y mejorar el bloque IoT2020
Entradas	Errores
Recursos necesarios	-
Salidas/Entregables	Calidad

Tabla 23: Reuniones con tutor bloque IoT2020

Tareas PLC:

- Formación

Duración	50 horas
Encargado	Ibon López
Descripción	Formación en el bloque PLC
Entradas	-
Recursos necesarios	Documentación e internet
Salidas/Entregables	Conocimiento

Tabla 24: Formación PLC

- Preparar entorno de desarrollo (Configuración) de TIA Portal

Duración	3 horas
Encargado	Ibon López
Descripción	Configurar TIA Portal para que sistemas externos puedan activar y desactivar entradas y salidas
Entradas	-
Recursos necesarios	TIA Portal
Salidas/Entregables	-

Tabla 25: Preparar entorno de desarrollo (Configuración) de TIA Portal

- Conectar actuadores

Duración	3 horas
Encargado	Ibon López y técnico de laboratorio
Descripción	Conectar actuadores al PLC
Entradas	Actuadores
Recursos necesarios	PLC
Salidas/Entregables	Poder automatizar los actuadores

Tabla 26: Conectar actuadores

- Testeo bloque PLC

Duración	5 horas
Encargado	Ibon López
Descripción	Comprobar que el PLC funciona correctamente
Entradas	-
Recursos necesarios	PLC
Salidas/Entregables	-

Tabla 27: Testeo bloque PLC

- Reuniones con tutor bloque PLC

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Reuniones para corregir y mejorar el bloque PLC
Entradas	Errores
Recursos necesarios	-
Salidas/Entregables	Calidad

Tabla 28: Reuniones con tutor bloque PLC



## Tareas Planta:

- Formación

Duración	50 horas
Encargado	Ibon López
Descripción	Formación para poder diseñar e imprimir piezas 3D
Entradas	-
Recursos necesarios	Documentación e internet
Salidas/Entregables	Conocimiento

Tabla 29: Formación planta

- Diseño de piezas

Duración	10 horas
Encargado	Ibon López
Descripción	Diseñar las piezas para que encajen perfectamente en la maqueta
Entradas	-
Recursos necesarios	Fusion 360
Salidas/Entregables	Boceto pieza

Tabla 30: Diseño de piezas

- Impresión piezas

Duración	15 horas
Encargado	Ibon López
Descripción	Imprimir las piezas diseñadas con la impresora 3D
Entradas	Boceto pieza
Recursos necesarios	Cura
Salidas/Entregables	Pieza

Tabla 31: Impresión piezas

- Compra de material

Duración	3 hora
Encargado	Ibon López y encargado de laboratorio
Descripción	Comprar el material necesario para la construcción de la maqueta
Entradas	-
Recursos necesarios	-
Salidas/Entregables	Material

Tabla 32: Compra de material

- Montar maqueta

Duración	25 horas
Encargado	Ibon López y encargado de laboratorio
Descripción	Construcción de la maqueta
Entradas	Material y piezas 3D
Recursos necesarios	Pegamento, Sierra y lima
Salidas/Entregables	Maqueta

Tabla 33: Montar maqueta

- Reuniones con tutor y técnico laboratorio bloque planta

Duración	15 horas
Encargado	Ibon López, Oskar Casquero y encargado de mantenimiento
Descripción	Reuniones para arreglar y mejorar el bloque Planta
Entradas	Errores
Recursos necesarios	-
Salidas/Entregables	Calidad

Tabla 34: Reuniones con tutor y técnico laboratorio bloque planta

## Tareas Gestión de Proyecto:

- Análisis de objetivos

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Definir los objetivos del trabajo
Entradas	-
Recursos necesarios	Word
Salidas/Entregables	Objetivos y alcance del trabajo

Tabla 35: Análisis de objetivos

- Reuniones con el tutor

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Reuniones para dirigir el trabajo hacia un objetivo común
Entradas	-
Recursos necesarios	-
Salidas/Entregables	Claridad

Tabla 36: Reuniones con el tutor

- Documentación

Duración	40 horas
Encargado	Ibon López
Descripción	Documentar el proyecto realizado
Entradas	-
Recursos necesarios	Word y PowerPoint
Salidas/Entregables	Memoria TFG

Tabla 37: Documentación

- Revisión de la memoria

Duración	10 horas
Encargado	Ibon López y Oskar Casquero
Descripción	Revisar la documentación para mejorarla
Entradas	Memoria TFG
Recursos necesarios	Word
Salidas/Entregables	Memoria TFG mejorada

*Tabla 38: Revisión de la memoria*

Tareas	Estimación de horas
<b>Arduino</b>	<b>88</b>
Formación	50
Elección de sensores	5
Lectura sensores	5
Integración sensores	5
Wifi	3
Envío de datos MQTT	5
Testeo	5
Reuniones con tutor	10
<b>IoT2020</b>	<b>144</b>
Formación	60
Instalación sistema operativo Yocto	3
Configuración IoT2020	5
Configuración Wifi	3
Probar placa microcontrolador	5
Eclipse IDE	5
Formación Node-RED	10
Módulo S7-Node-RED	5
Módulo MQTT-Node-RED	5
Modulo MQTT-Broker-Node-RED	5
Programa de control	10
Elección Broker-MQTT-Node-RED	3
Panel de Control en Node-RED	5
Testeo	10
Reuniones con tutor	10
<b>PLC</b>	<b>71</b>
Formación	50
Configuración de TIA Portal	3
Conectar actuadores	3
Testeo	5
Reuniones con tutor	10
<b>Planta</b>	<b>119</b>
Formación	50
Diseño de piezas	10
Impresión piezas	16
Compra material	3
Montar maqueta	25
Reuniones con tutor y técnico laboratorio	15
<b>Gestión de Proyecto</b>	<b>70</b>
Análisis de objetivos	10
Reuniones con el tutor	10
Documentación	40
Revisión de la memoria	10
<b>TOTAL</b>	<b>492</b>

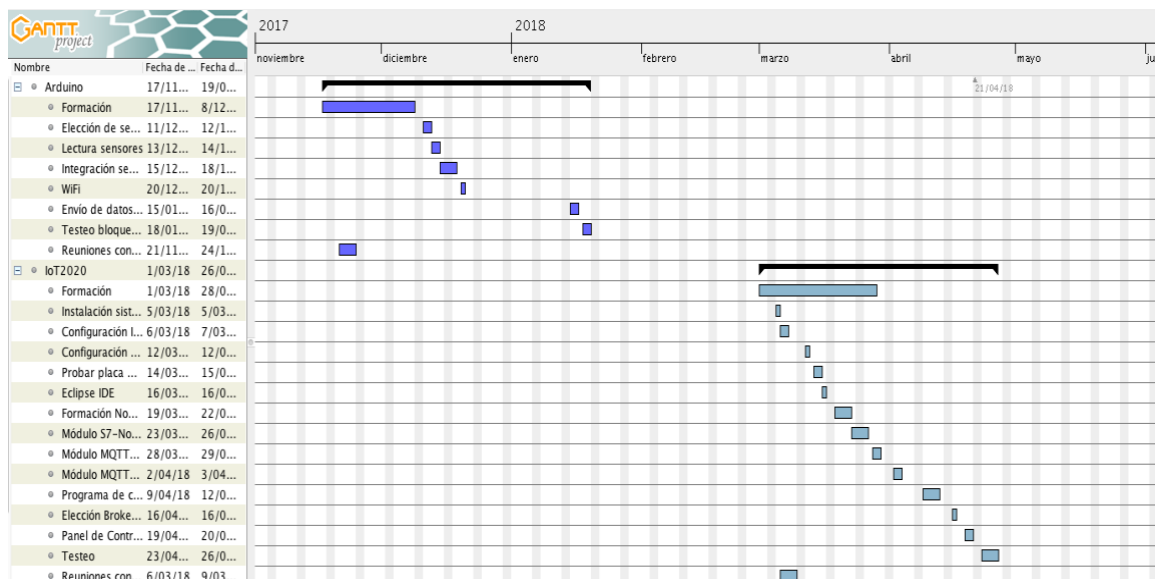
## 9. Diagrama de Gantt

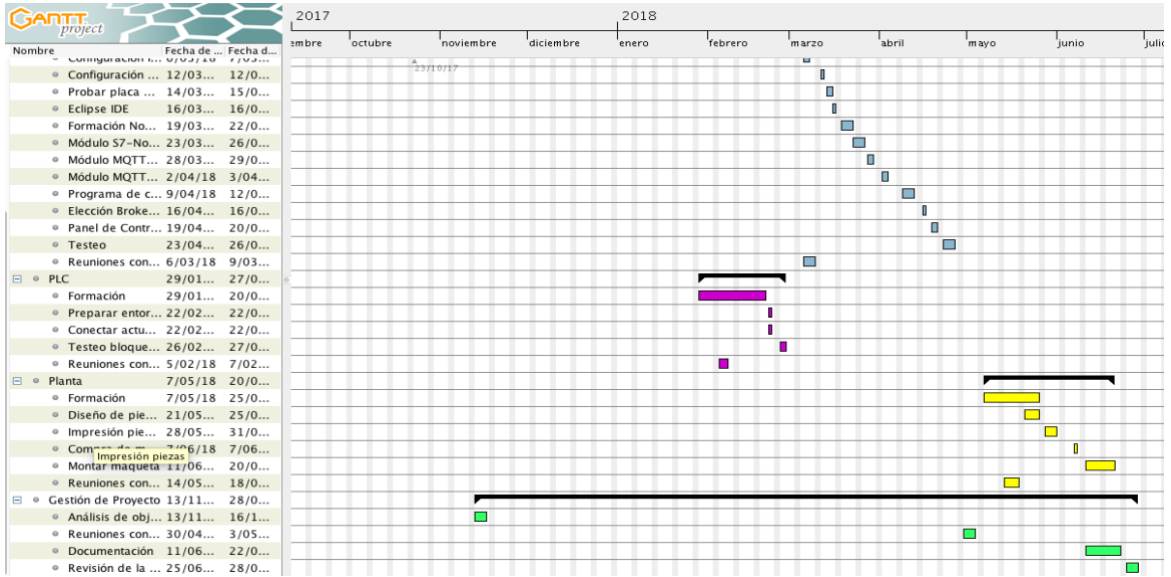
El total de horas invertidas en el trabajo ha sido de 492 horas. Ya que el trabajo comenzó a mediados de noviembre de 2017 y acaba a finales de junio de 2018.

Para mostrar algo de claridad se ofrece a continuación un diagrama de *GANTT* adaptado a semanas.

Guía de colores:

- Azul: Arduino
- Gris: IoT2020
- Morado: PLC
- Amarillo: Planta
- Verde: Gestión de proyecto

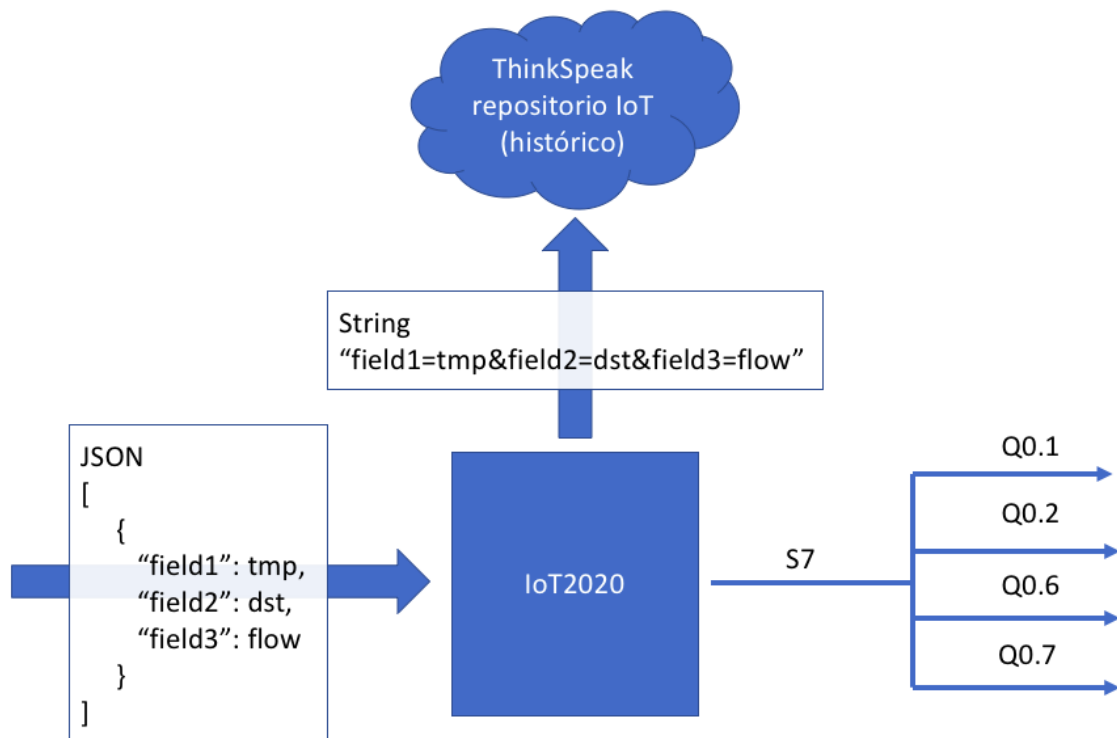




## 10. Diseño de bajo nivel

El proyecto esta dividido en cuatro bloques: el Gateway IoT2020, gestión de sensores y gestión de actuadores y la planta (maqueta didáctica).

### 10.1. Gateway IoT2020



En este bloque se gestiona la mayor parte del proyecto por lo que podría considerarse el bloque más importante. El IoT2020 se utiliza como pasarela entre el Arduino y el PLC. Por lo que recibe unos datos del Arduino, el IoT2020 los interpreta y envía ordenes al PLC. Para una explicación más precisa de este bloque a continuación se va a profundizar en el IoT2020 y su aplicación real en este proyecto.





Figura 2: SIMATIC IoT2020

El SIMATIC IoT2020 está diseñado por Siemens y es una plataforma abierta para desarrollar y realizar fácilmente ideas innovadoras en el campo del Internet de las Cosas (IoT – Internet of Things). Con su gran variabilidad y sus amplias posibilidades de expansión, las aplicaciones creativas pueden ser implementadas rápido y con un coste eficiente y todo esto con un hardware resistente diseñado para un rendimiento confiable 24/7 incluso en condiciones difíciles.

Para entrar fácilmente en el mundo de la automatización, Siemens proporciona un foro con una amplia comunidad y una gran variedad de ejemplos de aplicaciones.

Algunas de las características de las aplicaciones industriales IoT son las siguientes:

- Programación en lenguajes de alto nivel
- Amplio abanico de soluciones de comunicación con diversos protocolos, desde Modbus hasta protocolos en la nube como MQTT
- Beneficiarse de ejemplos de código abierto y de librerías
- Soporte de Yocto Linux

Especificaciones técnicas	SIMATIC IoT2020
Procesador	Intel Quark x1000
Almacenamiento masivo de datos	Espacio tarjeta microSD Card; SDHC hasta 32 GB;
RAM / Flash / SRAM	512 MB / 8 MB / 256 kB
Interfaz USB	1x controlador USB + dispositivo 1x
Interfaz Ethernet	1x 10/100 Ethernet RJ45
Firmware	Flash Firmware preinstalado
Sistema operativo	Yocto V2.1 (Krogoth) basado Linux
Tipo de voltaje	24 V DC

Tabla 39: Especificaciones técnicas SIMATIC IoT2020

Una vez se tiene el IoT lo siguiente que hay que hacer es instalar el sistema operativo en una tarjeta microSD. El OS que se va a utilizar es Yocto y se puede descargar la imagen desde la página de Siemens. Previamente se ha tenido que crear una cuenta para poder descargar archivos.

Yocto Project es un proyecto de colaboración de código abierto que proporciona plantillas, herramientas y métodos para ayudar a crear sistemas personalizados basados en Linux para productos IOT e integrados, independientemente de la arquitectura del hardware.

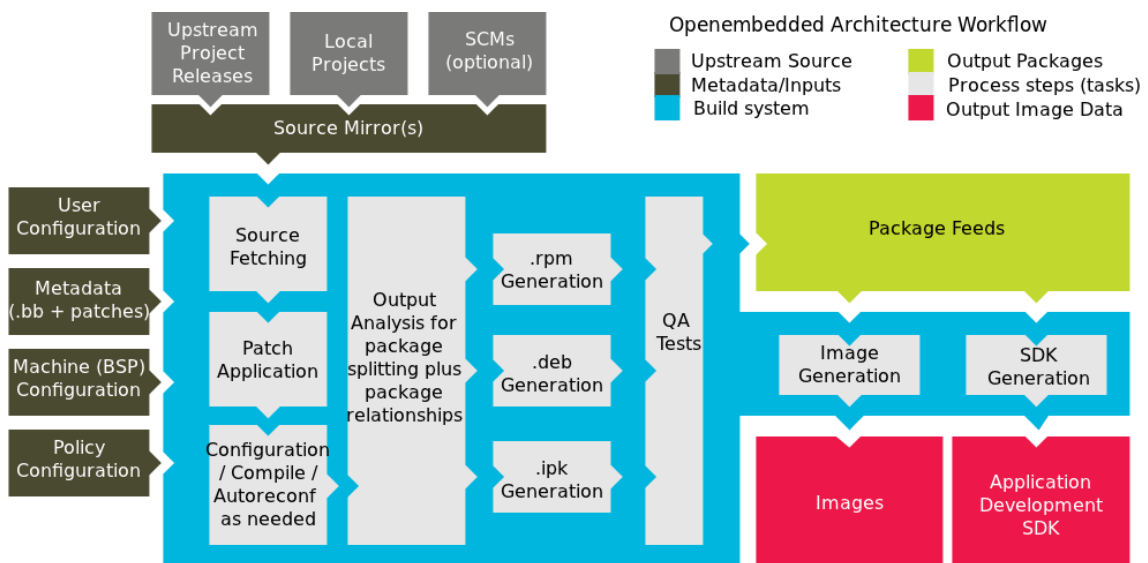


Figura 3: Arquitectura Yocto

El siguiente paso será copiar la imagen en la tarjeta microSD. En este proyecto se ha utilizado una tarjeta de 32GB. Una vez se copie la imagen ya estará el IoT2020 preparado para ser utilizado. Para conectarse remotamente al IoT2020 se ha utilizado el programa PuTTY para conectarse mediante ssh. Para una experiencia completa solo faltaría un último paso, configurar la red para que tenga acceso a la red local y así poder ser utilizado como Gateway. Para ello se han configurado dos archivos en el IoT2020:

- /etc/network/interfaces

En este fichero se ha configurado la parte del wifi. Para tener wifi se ha añadido un USB wifi al IoT2020.

- /etc/wpa\_supplicant.conf

En este fichero se ha configurado a la red que se quiere acceder y sus credenciales, tanto el usuario como la contraseña. En este proyecto se ha utilizado la red Eduroam.

#### 10.1.1. Node-RED

Uno de los principales motores que impulsan el Internet de las Cosas es IMB ya que está innovando mucho en esta tecnología y ofrece algunas creaciones con son muy útiles. Una de ellas es Node-RED, una nueva herramienta open-source que proporciona un editor de flujo, que se puede usar para crear funciones en JavaScript. Esta herramienta permite poder interconectar todos los elementos utilizados en el proyecto.

El tiempo de ejecución está creado en Node.js, por lo que si se quiere descargar node-RED, antes de nada, hay que instalar nodejs. Para ello hay que introducir el siguiente comando en la terminal del IoT2020: `opkg install nodejs`.

Una vez instalado nodejs se procede a instalar node-RED. Como se ha comentado antes, node-RED está basado en nodejs y para descargar paquetes javascript nodejs trae incorporado un administrador de paquetes JavaScript llamado npm. Para descargar hay que introducir el siguiente comando: `npm install node-red`. Esto puede tardar varios minutos y hay veces que el IoT2020 pierde la conexión con la red. Para solucionarlo hay

que dejar un proceso abierto haciendo ping a cualquier servidor. Así estará constantemente haciendo peticiones a la red y nunca perderá la conexión.

Para arrancar node-red hay que utilizar el comando `node /usr/lib/node_modules/node-red/red` y se abrirá en el puerto 1880 del localhost.

Para poder utilizar como Gateway el IoT2020 es obligatorio que se puedan enviar y recibir datos desde node-red. Para ello se han utilizado 3 módulos que soporta node-red, además de los módulos por defecto que trae node-red para crear un flujo estructurado y funcional. Estos tres módulos son: MQTT para publicar / suscribirse datos, Mosca para crear un Broker MQTT en el mismo IoT2020 y por último S7 para conectarse con el PLC.

A continuación, se va a explicar el protocolo MQTT, el Broker MQTT (Mosca), la comunicación S7 y el programa principal de node-RED:

#### 10.1.2. MQTT

MQTT, desarrollado por investigadores de IBM, es un protocolo de mensajería de publicación y suscripción basado en un Broker ligero, diseñado para ser abierto, simple, ligero y fácil de implementar. Los autores afirman que las características de MQTT lo hacen ideal para su uso en entornos restringidos, donde, por ejemplo, la red es costosa, tiene poco ancho de banda o no es confiable, o cuando se ejecuta en dispositivos integrados con capacidad limitada de procesamiento o memoria.

MQTT es un protocolo asincrónico. Algunos mensajes MQTT contienen un encabezado variable, presente después del encabezado fijo y antes de la carga útil, que contiene, por ejemplo, el nombre del protocolo, la versión del protocolo y los indicadores. Hay 14 tipos de mensajes diferentes definidos en MQTT, incluidos CONNECT, PUBLISH, SUBSCRIBE, UNSUBSCRIBE y DISCONNECT. Varios de estos mensajes tienen mensajes de reconocimiento dedicados. Por ejemplo, si un suscriptor de MQTT desea suscribirse a un tema, el suscriptor envía un mensaje SUSCRIBIR y espera la respuesta correspondiente de SUBACK del intermediario (servidor).

Aunque el uso de MQTT implica el uso de un protocolo de transporte orientado a la conexión confiable, como TCP, MQTT admite tres tipos de confiabilidad de aplicaciones para la entrega de mensajes de publicación:

- Nivel 2: entrega el mensaje exactamente una vez, lo que garantiza que los mensajes lleguen exactamente una vez;
- Nivel 1: entregue el mensaje al menos una vez, lo que garantiza que lleguen los mensajes, pero pueden producirse duplicados;
- Nivel 0: entregue el mensaje a lo sumo una vez, donde los mensajes llegan de acuerdo con los mejores esfuerzos de la red TCP / IP subyacente, lo que significa que puede ocurrir la pérdida o duplicación de mensajes, introducida por software u otras causas.

Estas opciones se seleccionan en el indicador QoS presente en el encabezado de cada mensaje MQTT.

#### 10.1.3. Broker MQTT (Mosca)

Además de enviar y recibir datos mediante MQTT en el IoT2020 también se va a instalar un servidor Broker llamado Mosca y así configurar de manera más precisa la manera de enviar y recibir datos.

Debido a que este Broker MQTT es implementado por Node.js, puede usar los nodos MQTT-in y MQTT-out sin un entorno MQTT como Mosquitto.

Para instalar basta con introducir este comando:

```
npm install node-red-contrib-mqtt-broker
```

#### 10.1.4. S7

La comunicación S7 es un nodo Node-RED para comunicarse con los PLC Siemens S7.

El nodo `node-red-contrib-s7comm` está diseñado para comunicarse con un PLC SIMATIC S7-300/1200/1500 de SIEMENS basado en el protocolo de comunicación RFC1006. El nodo `S7comm` puede establecer una conexión con el SIMATIC-S7. Además, puede leer/escribir direcciones del PLC con el tipo de datos S7 específico.

#### 10.1.5. Programa principal node-RED

A continuación, se explicará el programa principal del proyecto que se ha desarrollado en node-red. Para una explicación más sencilla y visual hay que fijarse en la imagen del panel de control de abajo. Este panel ha sido creado en node-red y consta de dos partes: el panel de control, en el que el usuario elegirá los valores óptimos y los valores de los sensores en tiempo real.

El programa consta de las siguientes 5 partes:

- Broker y valores de los sensores
- Comprobar que todos los elementos que van a actuar en la tarea estén a perfecta disposición
- Comportamiento normal de la tarea (flujo principal)
- Botón emergencia por si algo está saliendo mal
- Subir datos a ThingSpeak para guardar historial y visualización de datos

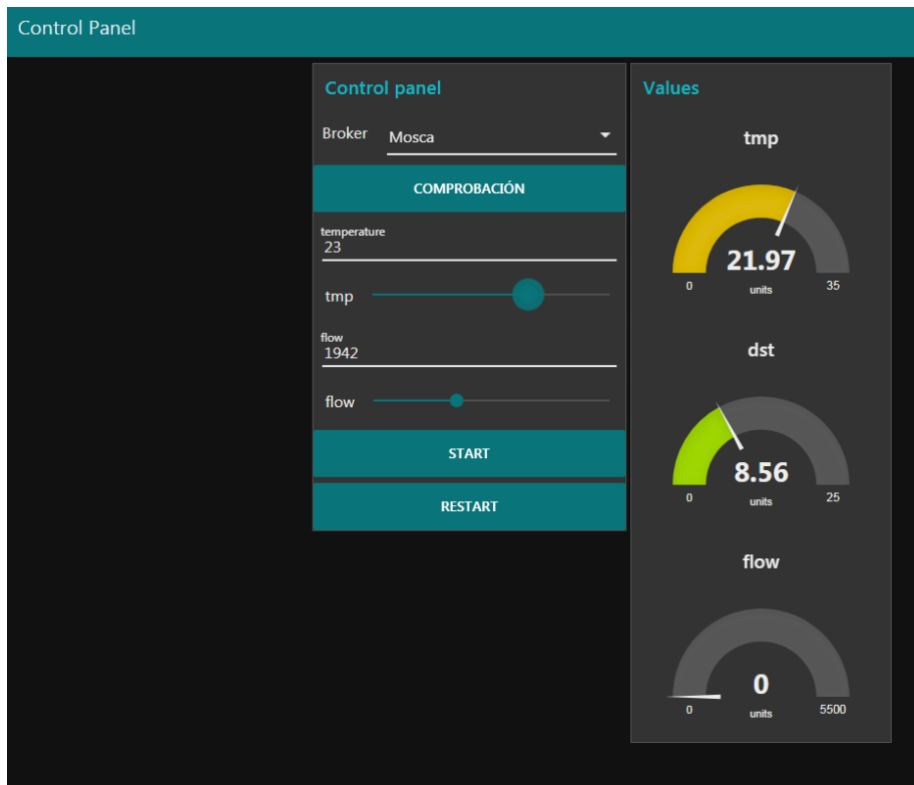


Figura 4: Panel de control

Una vez se han identificado todas las partes del programa se va a proceder a explicar el funcionamiento de todas ellas.

En primer lugar, como se ha mencionado anteriormente se ha instalado un servidor Broker MQTT llamado mosca. Para ello se ha configurado el nodo Broker para que escuche en el puerto 1884 del localhost. Una vez el servidor está conectado hay que crear un nodo MQTT de entrada para recibir los datos de los sensores en tiempo real. Éste recibirá todos los mensajes que se publiquen en el puerto 1884 del localhost en el topic mosca/sensors. Topic en el que el Arduino publicará todos los datos. Una vez se reciben los datos se convierte la cadena recibida en un objeto JSON y se cogen los tres valores de los sensores: temperatura, distancia y flujo de agua. Para finalizar con esta parte, se guardan los tres valores en tres variables globales distintas para luego gestionar esos valores de la forma adecuada. Por último, se muestran los tres valores en el panel de control. Se puede observar en la imagen de arriba como se muestran los tres valores.

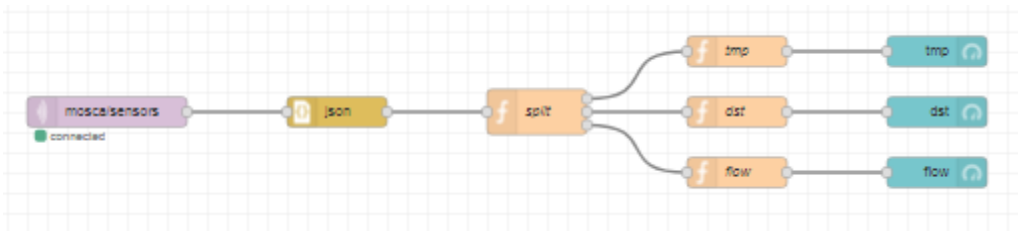


Figura 5: Flujo visualización de los valores de los sensores

Una vez se guardan los datos en las variables globales se procede a hacer la comprobación del sistema. Se añade un botón al panel de control para que cuando el usuario lo clique empiece a comprobar que el segundo tanque esta vacío. En caso de no estar vacío, es decir, que el agua esté a más de 4.5 cm de altura (valor mínimo para que la bomba funcione correctamente) se activará la bomba de ese tanque hasta que se vacíe. Una vez se vacíe se parará la bomba.

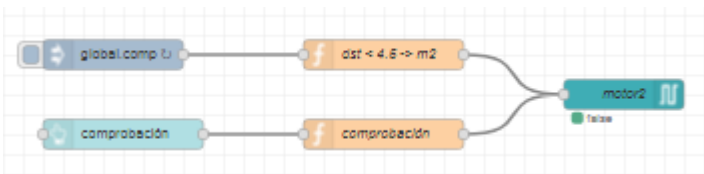


Figura 6: Flujo comprobación del proceso

Después de haber comprobado que todo esté en orden, el usuario deberá indicar una temperatura para calentar el agua y el flujo que hay que pasar al segundo tanque. Estos valores se introducirán en el panel de control. Seguidamente el usuario clicará el botón start para empezar el proceso. A continuación, se activará el calentador y el agitador y cuando la temperatura del agua llegue a la temperatura introducida por el usuario se pararán los dos actuadores y se pondrá en marcha la bomba. Cuando pasen los litros de agua que el usuario haya decidido se detendrá la bomba y aparecerá una notificación en el panel de control indicando que el proceso ha finalizado. Además, se enviará un correo al responsable del proceso para indicarle cómo ha ido el proceso que éste ha finalizado.



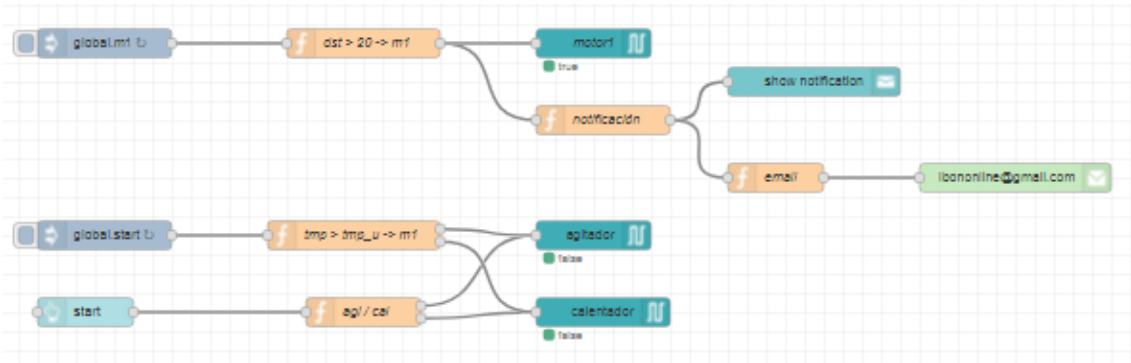


Figura 7: Flujo principal

También se ha puesto un botón de emergencia por si acaso durante el proceso algo estuviese saliendo mal y así poder parar el proceso y tener tiempo de reacción.

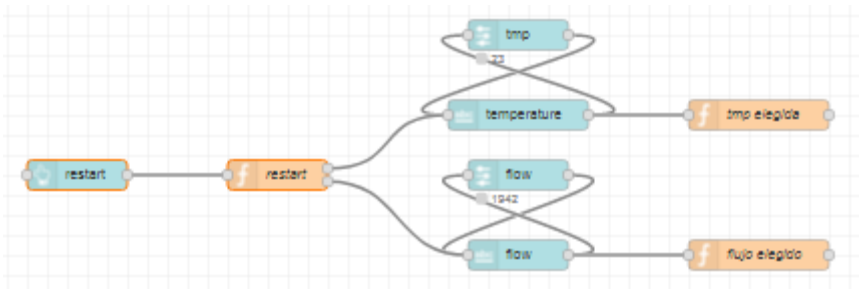


Figura 8: Flujo parada emergencia

Por último, se ha colocado un nodo MQTT de salida al servidor Broker de ThingSpeak para poder enviar los datos de los sensores y así poder guardar el historial del proceso y visualizar el proceso de forma gráfica.

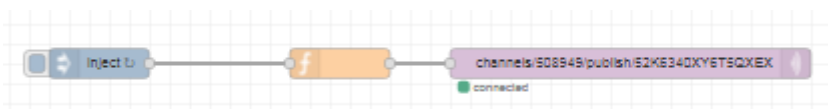


Figura 9: Flujo publicación datos ThingSpeak

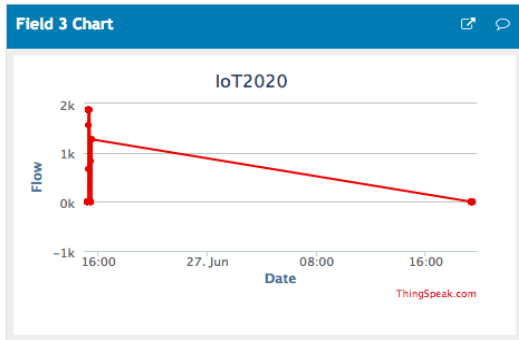
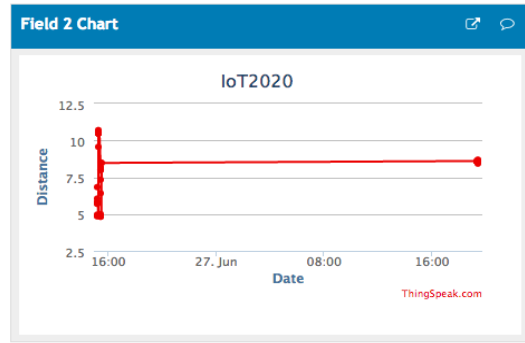
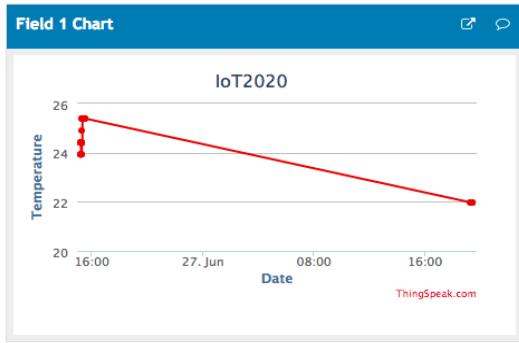
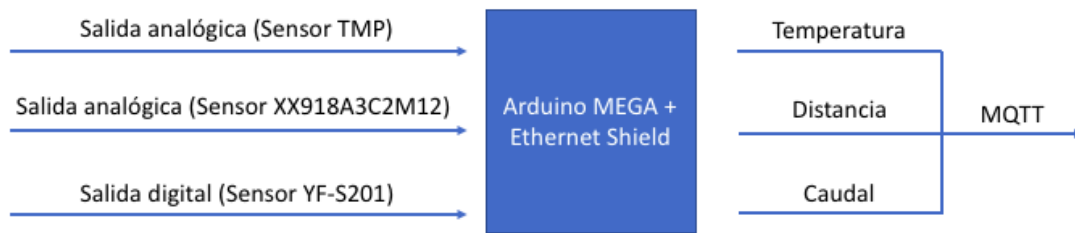


Figura 10: Gráficos ThingSpeak

## 10.2. Gestión de sensores



Como se puede observar en el diagrama anterior, para poder llevar a cabo este bloque ha sido necesario el uso de los siguientes elementos:

- Arduino MEGA
- Sensor analógico de temperatura (LM35DZ)
- Sensor analógico de distancia (XX918A3C2M12)
- Sensor digital de flujo de líquido (YF-S201)
- Arduino Ethernet Shield

A continuación, se explicará cada elemento y su aplicación en el proyecto de forma precisa.

### 10.2.1. Arduino MEGA



Figura 11: Arduino MEGA



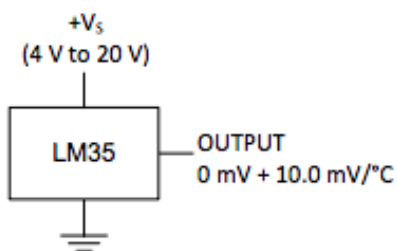
- Descripción

El sensor LM35DZ es un dispositivo de temperatura con un circuito integrado de precisión. Su salida es analógica, es decir, proporciona un voltaje proporcional a la temperatura. El dispositivo LM35DZ no requiere ninguna calibración externa para poder proporcionar las típicas precisiones de  $\pm 1/4$  °C a temperatura ambiente y  $\pm 3/4$  °C en un rango de temperatura entre 0 y 100°C.

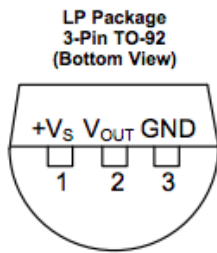
- Características

- Calibrado directamente en Celsius (Centígrados)
- Escala de factor lineal de 10mV por cada grado centígrado
- Precisión asegurada de 0.5°C (a 25°C)
- Clasificado para rango completo de 0°C a 100°C
- Adecuado para aplicaciones remotas
- Funciona de 4 V a 30 V
- Consumo de corriente inferior a 60μA
- Bajo autocalentamiento, 0.08°C en aire
- Tiene una precisión de  $\pm 1/4$  °C

- Sensor de temperatura (centígrado) básico



- Configuración y descripción de los pines



Pin	Nombre	Descripción
1	+Vs	Alimentación +Vcc
2	Vout	Datos
3	GND	0V

Tabla 41: Descripción pines LM35DZ

- Programación Arduino

Teniendo un sensor analógico hay que convertir la tensión en temperatura. Para ello Arduino proporciona una función para saber cuanta tensión pasa por ese sensor. Para utilizar esa función hay que indicar el pin en el que está conectada la salida analógica del sensor en el Arduino -> `analogRead(analogPin_LM)`.

Una vez se obtiene la tensión hay que transformarla en temperatura (grados centígrados). Para convertir la señal analógica en digital Arduino dispone de 10 bits y mapea la tensión de entrada entre 0 y 5 voltios (ya que el sensor TMP está alimentando de 5V), por lo que convierte la tensión de entrada en valores enteros entre 0 y 1024 ( $2^{10}$ ). Esto produce una resolución entre lecturas de: 5 voltios / 1024 unidades o 4.9mV por unidad. Una vez sabido esto, se llega a la siguiente conclusión:

$$\text{Volt} = \frac{5}{1024} * \text{analogRead}(\text{analogPin\_LM})$$

Una vez sabidos los voltios, el fabricante del LM35DZ dice que la salida de tensión tiene un factor de escala lineal de 10mV (mili voltios) por cada grado de temperatura (grados

centígrados). Por lo que la formula para transformar tensión en temperatura es la siguiente:

$$\text{Temperatura} = \frac{5}{1024} * \text{analogRead}(\text{analogPin\_LM}) * 100$$

- Conexiones Arduino

Pin Arduino Mega	Pin LM35DZ
VCC (5V)	+Vs
Pin analógico 0	Vout
GND	GND

Tabla 42: Conexiones Arduino LM35DZ

Además de las conexiones de la tabla anterior, se ha colocado un condensador entre Vout y GND para intentar reducir los picos entre lecturas de la señal analógica. Colocando un condensador se consigue que éste sea capaz de almacenar energía eléctrica y así conseguir que en el tiempo entre lecturas no haya mucha desviación de señal. Gracias a este condensador se consigue una tensión más estable. Esto podría equivaler a sacar n muestras y hacer el promedio de ellas.

### 10.2.3. Sensor de distancia (OsiSense XX918A3C2M12)



Figura 13: OsiSense XX918A3C2M12

- Descripción

El sensor analógico OsiSense es un dispositivo para medir la distancia. Este sensor es ideal para usar en este proyecto ya que es posible la detección de nivel de líquido sin con

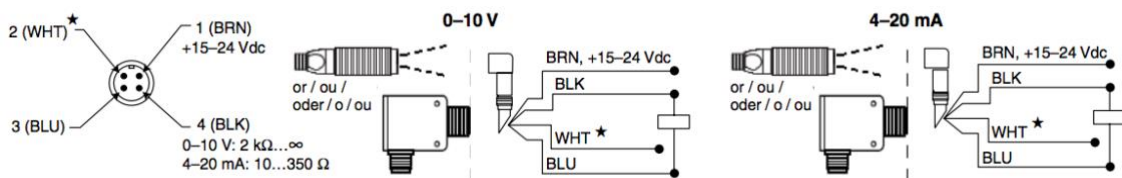
contacto con este mismo. Además, teniendo un rango de distancia desde 0.051m hasta 0.508m proporciona una mayor precisión al proyecto.

- Características
  - Sensor ultrasónico
  - Cuerpo cilíndrico M18
  - Tecnología de escritura 4-wire
  - Salida analógica de 4 a 20mA
  - Tensión de alimentación nominal 12 ... 24V DC con protección de polaridad inversa
  - Rango de detección 0.051 ... 0.508m
  - Ángulo del haz de luz: 6°
  - Grado de protección IP IP67 conforme a IEC 60529
  
- Estado del LED

Color LED	Estado
LED doble color	Asistencia de instalación
LED verde	Estado de alimentación
LED amarillo	Estado de salida

Tabla 43: Estado del LED OsiSense XX918A3C2M12

- Configuración y descripción de los pines





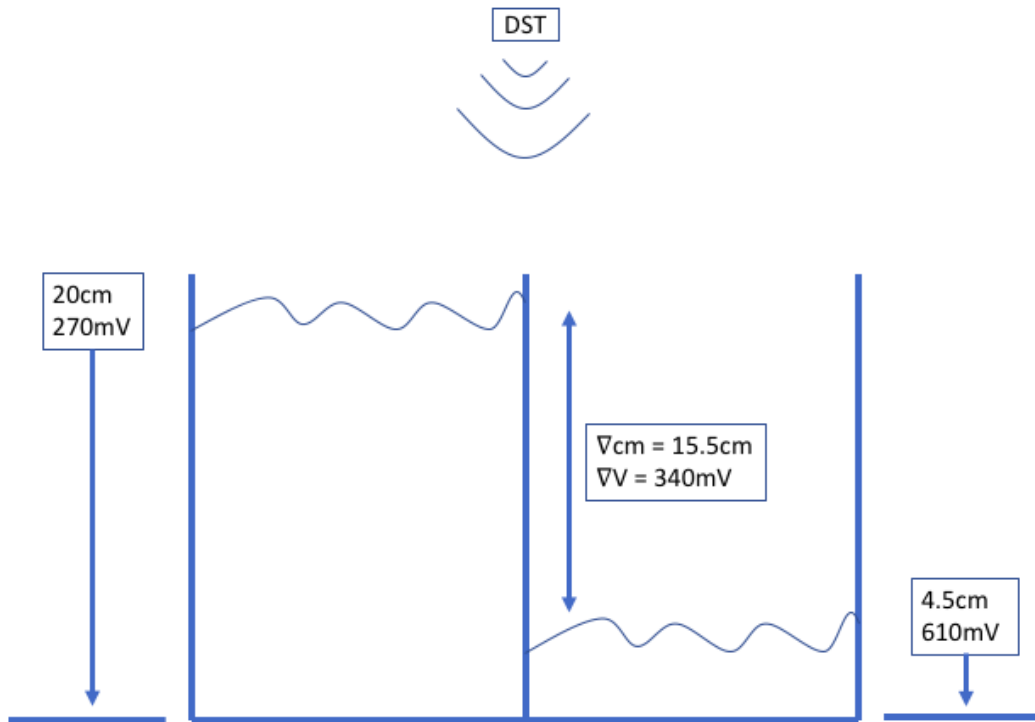
Pin	Nombre	Descripción
1	BRN (marrón)	Alimentación +Vcc
2	WHT (blanco)	Configuración
3	BLU (azul)	0V
4	BLK (negro)	Señal salida

Tabla 44: Descripción pines OsiSense XX918A3C2M12

## Programación Arduino

En la imagen de abajo se pueden ver dos tanques de agua, uno llenado hasta una altura de 4.5cm (altura mínima) y el otro a una altura de 20cm (altura máxima). Se ha determinado que la altura mínima sea de 4.5 cm ya que la bomba de agua debe estar constantemente bajo el agua porque si no se introduciría aire en ella y dificultaría el funcionamiento óptimo del proyecto, además de que la bomba se estropearía más rápido a lo largo del tiempo con un mal funcionamiento.

En el dibujo también se puede observar que a una altura de 20 centímetros el sensor de ultrasonidos marca una tensión de 270mV, gracias a la función previamente comentada, `analogRead()`, y una tensión de 610mV cuando el nivel del agua se encuentra a 4.5 centímetros. Además, se puede ver que la diferencia de centímetros entre el punto máximo y el mínimo es de 15.5 centímetros y la diferencia de voltaje es de 340mV.



Teniendo en cuenta los datos de la imagen, se ha llegado a la conclusión de que por cada milivoltio de diferencia que haya entre el nivel máximo (20cm) y el mínimo (4.5cm) equivale a 0.0455 centímetros ( $15.5\text{cm} / 340\text{mV}$ ).

Una vez sabido todo esto se ha procedido a la programación del sensor. Para ello, se ha vuelto a coger la tensión actual del sensor con `analogRead()` y se le ha restado la tensión mínima (270mV) para así saber la tensión que ha aumentado respecto al mínimo. A continuación, se ha multiplicado esa diferencia por lo equivalente a un centímetro ( $0.0455\text{cm} / \text{mV}$ ). Así se calcula el incremento que ha habido en centímetros y el paso final sería sumar ese incremento a la distancia mínima (4.5cm).

Cabe recordar que este algoritmo se ha gestionado así, debido a las necesidades del proyecto. Por lo que si se está utilizando una maqueta de otras dimensiones o unos datos distintos este algoritmo no sería útil.

- Conexiones Arduino

Como se puede observar en la tabla de abajo se ha tenido que introducir una fuente de alimentación para alimentar el sensor con 15V ya que Arduino solo ofrece 3.3V o 5V. En cuanto a la masa (GND – 0V) se ha conectado el sensor a dos GND, tanto a la fuente de alimentación como al pin de Arduino y posteriormente unir ambas masas. Esto se hace para tener como referencia las dos masas y así poder tratar perfectamente la señal analógica en Arduino. Para ello, primero hay que transformar la intensidad que atraviesa el circuito eléctrico en voltaje. Esto se puede conseguir poniendo una resistencia entre las masas y la salida de datos (BLK) y así poder aplicar la ley de Ohm. Esta ley dice lo siguiente: Es la relación de intensidad que circula por el circuito, la cual es igual a la diferencia de voltaje, entre la resistencia que encuentra la corriente en el circuito.

$$I = \frac{V}{R}$$

I = Intensidad (amperios, A)

V = Voltaje (voltios, V)

R = Resistencia (ohmios,  $\Omega$ )

Entonces, sabiendo que la intensidad del Arduino es de 20mA y el sensor está suministrado por 5V, la resistencia que hay que colocar es de 250 ohmios. Además, al igual que con el sensor de temperatura, se ha puesto un condensador entre la salida de datos y la masa para así, amortiguar la bajada de tensión y obtener una tensión estable entre lecturas.

Por lo que las conexiones tanto con Arduino como con la fuente de alimentación quedarían de la siguiente manera:

Pin Arduino Mega	Pin	Fuente alimentación
	XX918A3C2M12	
	+Vs	VCC (15 V)
A0	Vout	
GND	GND	GND

Tabla 45: Conexión Arduino OsiSense XX918A3C2M12

#### 10.2.4. Sensor de flujo (YF-S201)



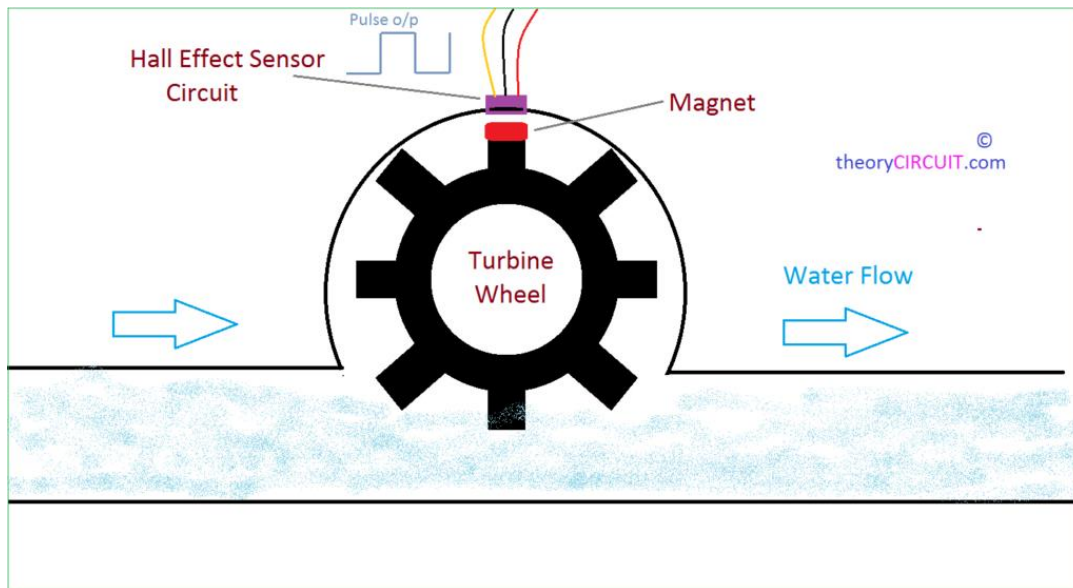
Figura 14: YF-S201

- Descripción

Este sensor está en línea con la corriente de agua y contiene un sensor de molinete para medir la cantidad de líquido que ha pasado tras él. Hay un sensor magnético integrado de efecto hall que emite un impulso eléctrico con cada revolución. El sensor de efecto hall está sellado de la tubería de agua y permite que el sensor se mantenga seguro y seco.

Para calcular el flujo del agua es necesario contar los pulsos de la salida del sensor. Cada pulso es aproximadamente 2.25 mililitros. La señal de pulso es una onda cuadrada simple, por lo que es bastante fácil de registrar y convertir a litros por minuto con la siguiente fórmula:

Frecuencia de pulso (Hz) / 7.5 = tasa de flujo en L / min.



- Características
  - Sensor de efecto Hall
  - Voltaje de trabajo de 5 a 18V DC (voltaje mínimo de trabajo probado 4.5V)
  - Consumo máximo de corriente de 15 mA a 5 V
  - Salida digital de 5V TTL
  - Velocidad de flujo de trabajo de 1 a 30 L / min.
  - Precisión de  $\pm 10\%$
  - Frecuencia (Hz) =  $7.5 * \text{Caudal (L / min)}$
  - Pulsos por litro: 450
  
- Configuración y descripción de los pines

Pin	Cable	Descripción
1	Rojo	Alimentación +Vcc
2	Negro	Masa 0V
3	Amarillo	Pulso PWM

Tabla 46: Descripción pines YF-S201

PWM consiste en la modulación por ancho de pulsos.

- Programación Arduino

Para conseguir el flujo que pasa por el sensor digital YF-S201 se ha conecta el pulso PWM a la entrada digital 2, la que utiliza la interrupción 0. Configurado para que provoque la interrupción cuando el estado del flanco de bajada cambie. Con la función `attachInterrupt` (`sensorInterrupt`, `pulseCounter`, `FALLING`) se configura que en cada interrupción se ejecute la función `pulseCounter()`. Ésta cuenta los pulsos que hay durante un segundo. Ya que solo se calcula el flujo una vez por segundo.

Cuando se esté calculando el flujo con la función `detachInterrupt(0)` se cancela la interrupción para no interferir mientras se está calculando. Para calcular el flujo, lo primero que hay que hacer es calcular la frecuencia con la que pasa el líquido en un tiempo determinado. Una vez se obtiene la frecuencia se procede a calcular el flujo (L / min) con la siguiente fórmula:

$$\text{Frecuencia (Hz)} = 7.5 * \text{flujo (L / min)}$$

Para saber cuantos litros han pasado en un intervalo de un segundo hay que dividir el flujo entre 60 y para finalizar, hay que multiplicar por 1000 para convertirlo en mililitros.

- Conexiones Arduino

Pin Arduino Mega	Cable YF-S201
VCC (5V)	Rojo
Pin digital 2	Amarillo
GND	Negro

Tabla 47: Conexiones Arduino YF-S201

#### 10.2.5. Envío de datos

Para separar claramente los bloques del proyecto, una vez se consiguen los datos de los sensores, éstos se envían al Gateway IoT2020 para que gestione él los datos y así en el

bloque de los sensores solo se haga la lectura de los mismos. Para enviar los datos se ha utilizado el protocolo, anteriormente citado, MQTT y para ello se ha utilizado el siguiente software y hardware:

- Librería Ethernet (SW)
- Librería PubSubClient (SW)
- Ethernet Shield (HW)

A continuación, se van a explicar todos los elementos que se han utilizado para llevar a cabo esta tarea, tanto lo anteriormente citado como el algoritmo:

Para poder enviar datos a un servidor es necesario que el Arduino tenga salida a Internet. Para ello se ha colocado una Ethernet Shield y así poder conectarse a una red Ethernet. Esta placa implementa el protocolo de red TCP/IP y es necesaria la librería Ethernet para poder leer y escribir los datos que pasan por el puerto ethernet y así permitir al Arduino conectarse a internet.

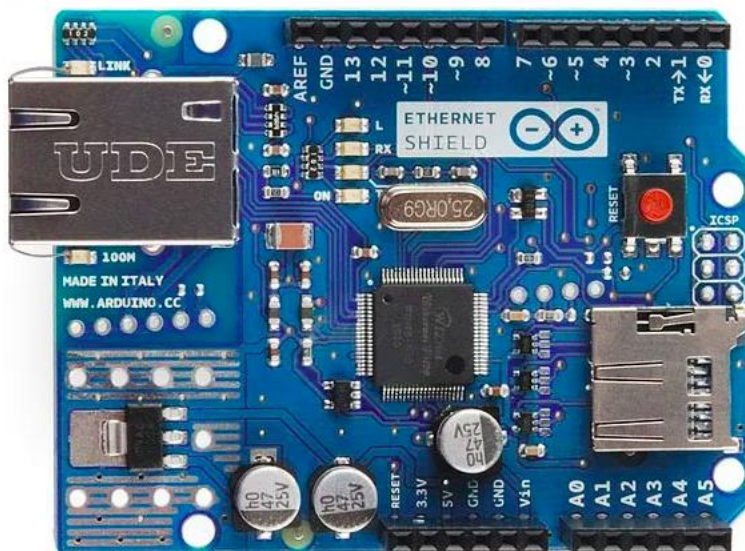


Figura 15: Arduino Shield

La placa se puede utilizar tanto para montar un servidor que acepte conexiones entrantes como para ser un cliente que haga las conexiones salientes. En este proyecto se ha





- MISO (Master In Slave Out) - La línea esclava para enviar datos al maestro
- MOSI (Master Out Slave In) - La línea Master para enviar datos a los periféricos,
- SCK (Serial Clock) - Los pulsos de reloj que sincronizan la transmisión de datos generada por el maestro

y una línea específica para cada dispositivo:

- SS (Slave Select): el pin que el maestro puede usar para habilitar y deshabilitar dispositivos específicos.

Por último, solo quedaría enviar los datos mediante el protocolo MQTT. Para ello se ha utilizado la librería PubSubClient. Esta librería proporciona un cliente para realizar mensajes de publicación / suscripciones simples con un servidor que admite MQTT y ha sido desarrollada por Nick O'Leary - [@knolleary](https://github.com/knolleary). Esta librería se puede descargar desde GitHub:

<https://github.com/knolleary/pubsubclient/releases/tag/v2.6>

Se ha utilizado esta librería ya que utiliza la API Arduino Ethernet Client para interactuar con el hardware de red subyacente y la cual es compatible con Arduino Ethernet Shield.

- Programación Arduino

Para enviar los datos mediante MQTT se han creado dos clientes, uno EthernetClient para conectarse a una red ethernet y el otro PubSubClient para publicar los datos en el broker MQTT. A continuación, con la ayuda de las funciones Ethernet.begin(mac) y Ethernet.localIP() se le da una dirección IP al Arduino. Una vez se obtiene la dirección IP se establece una conexión con el broker MQTT en el servidor 192.168.0.200 en el puerto 1884.

Con la conexión establecida al broker, lo único que falta es enviar los datos. Para ello se ha creado una función mqttpublish() que publica los datos mediante la función

`mqttClient.publish(topicBuffer, msgBuffer)`. `topicBuffer` indica el topic en el que hay publicar los datos y `msgBuffer` el mensaje que se quiere publicar en formato JSON.

### 10.3. Gestión actuadores



Para la gestión de actuadores hay dos elementos que son imprescindibles en este proyecto, un PLC SIMATIC S7-1200 para controlar los actuadores y así poder automatizar el proyecto y el TIA Portal para tener un entorno de desarrollo integrado.

#### 10.3.1. SIMATIC S7-1200



Figura 17: SIMATIC S7-1200

El controlador lógico programable (PLC) S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. La CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario,

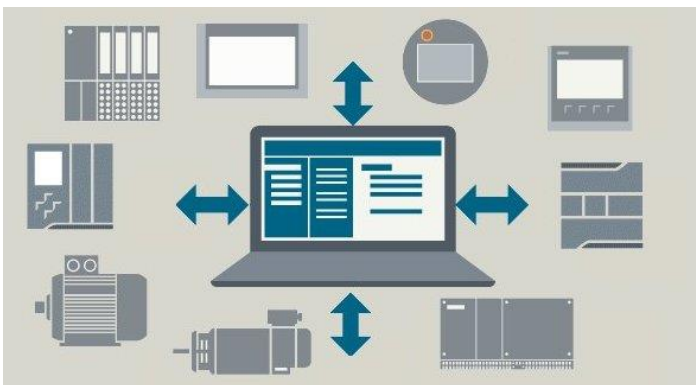
que puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

- Características

	CPU 1214C
Memoria de usuario	
Memoria de trabajo	50 KB
Memoria de carga	2 MB
Memoria remanente	2 KB
E/S integradas locales	
Digitales	14 entradas/10 salidas
Analógicas	2 entradas
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)
PROFINET	1 puerto de comunicación Ethernet
Velocidad de ejecución booleana	0,1 $\mu$ s/instrucción

Tabla 48: Características SIMATIC S7-1200

### 10.3.2. TIA Portal (STEP 7 Basic)



TIA Portal es el innovador sistema de ingeniería que permite configurar de forma intuitiva y eficiente todos los procesos de planificación y producción. Convence por su

funcionalidad probada y por ofrecer un entorno de ingeniería unificado para todas las tareas de control, visualización y accionamiento.

El TIA Portal incorpora la última versión de Software de Ingeniería SIMATIC STEP 7 para la planificación, programación y diagnóstico de todos los controladores SIMATIC, pantallas de visualización y accionamientos SINAMICS de última generación.

El software STEP 7 Basic ofrece un entorno amigable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como PLCs y dispositivos HMI. STEP 7 Basic ofrece dos lenguajes de programación (KOP y FUP) que permiten desarrollar el programa de control de la aplicación de forma fácil y eficiente.

- Configuración

En este proyecto, para que el PLC se pueda comunicar vía S7 con node-red hay que configurar la red del PLC de forma expresa (hay que indicar dirección IP) y además hay que activar la comunicación PUT/GET para permitir el acceso de aplicaciones remotas.

Una vez configurado el PLC, se ha creado un programa principal vacío (OB1 vacío) ya que la gestión del proceso está programada en node-red y éste le indica al PLC que salidas tiene que activar y cuales desactivar. Para ello se ha creado una tabla de observación con los actuadores que se utilizan en el proyecto para la monitorización dentro del propio PLC.

### 10.3.3. Actuadores

Los actuadores que se han utilizado en este proyecto han sido los siguientes, 2 bombas para el trasvase de líquidos, un calentador de líquidos y un agitador.

#### 10.3.3.1. Bombas

Las bombas se van a utilizar para traspasar el agua de un tanque a otro, colocando una bomba en cada tanque. De entre las opciones que se han encontrado en el mercado se han

escogido las que mejor se adecuan al proyecto, cuyas características son las que figuran a continuación:

Tensión de alimentación	5-12 V
Caudal	1.67-5.8 L / min.
Potencia eléctrica	0.5-5 W

#### 10.3.3.2. Calentador

Como elemento calentador, hemos elegido una resistencia estanca especial para el calentamiento de líquidos, cuyas características principales son:

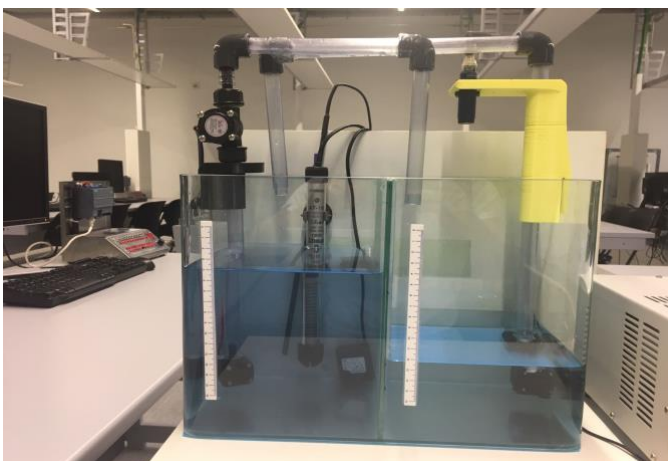
Fuente de alimentación	220-240 V
Caudal	50-60 Hz
Potencia eléctrica	100 W

#### 10.3.3.3. Agitador

Como elemento agitador, destinado a homogeneizar la temperatura del agua en el tanque, hemos escogido una bomba de recirculación de líquidos con las siguientes características:

Tensión de alimentación	220-240 V
Caudal	50 Hz
Potencia	2 W
Caudal	3.33 L / min.

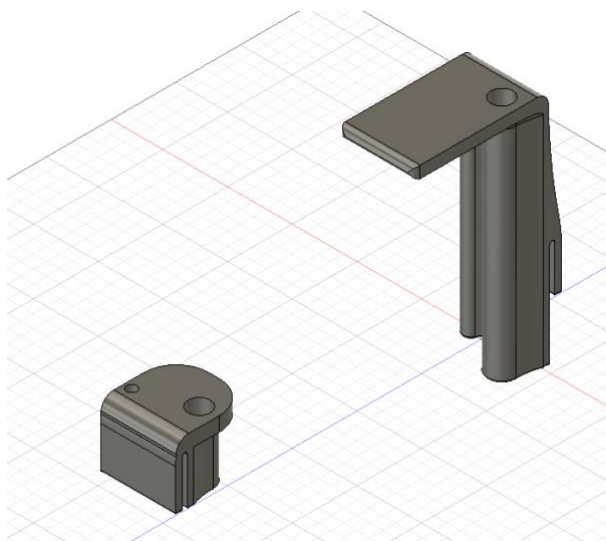
### 10.4. Planta

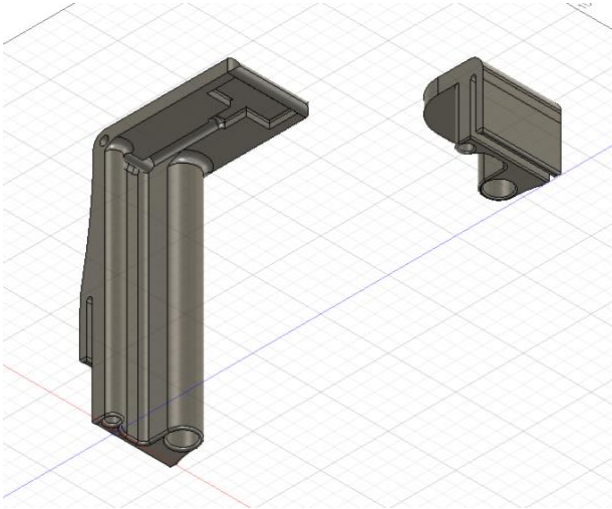


Como se puede observar en la imagen de arriba, la planta, además de incorporar los sensores y actuadores comentados previamente también incorpora 6 tubos de un diámetro de 16 mm, 4 codos para unir los tubos y dos soportes. Uno de ellos está situado en el tanque de la derecha y se utiliza como soporte del sensor de ultrasonidos y del tubo por el que pasa el líquido. El otro, va situado en el tanque de la izquierda y sirve para soportar otro tubo y el sensor de flujo.

A continuación, se va a explicar el procedimiento que se ha llevado para la impresión de las piezas 3D.

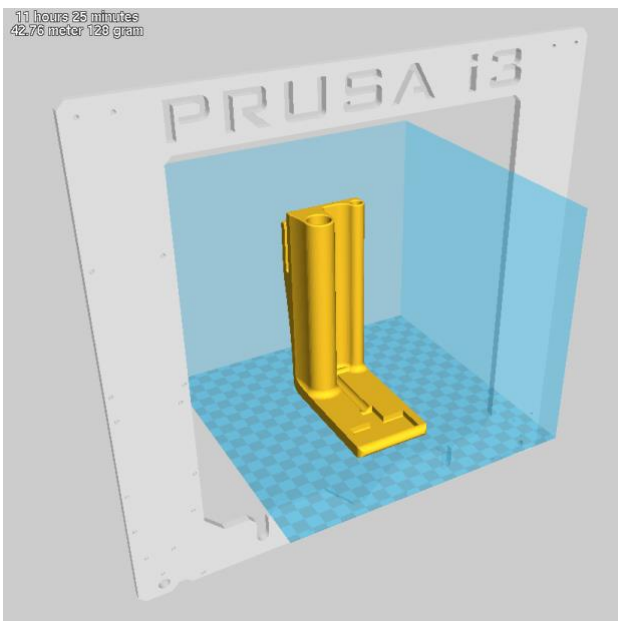
En primer lugar, se han diseñado las dos piezas en el programa, de modelado 3D, Fusión 360. Antes de empezar a diseñar la pieza, hay que tener en cuenta las medidas a la que la impresora puede imprimir. El resultado ha sido el siguiente:



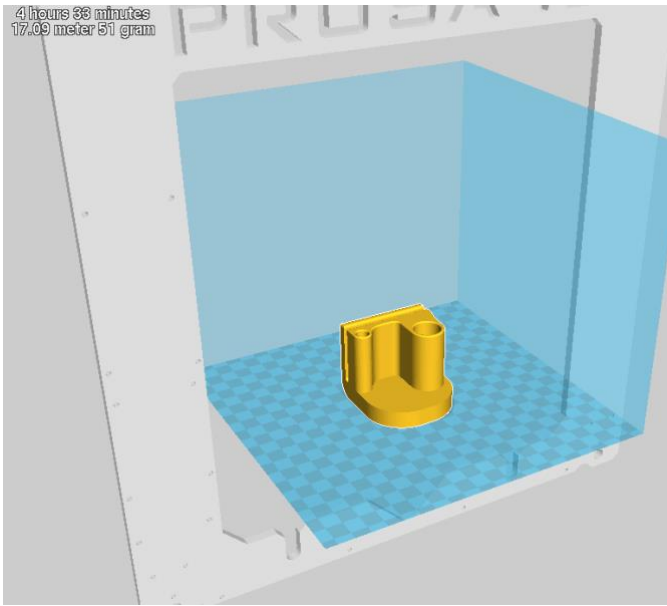


Una vez se obtiene el diseño pretendido, hay que exportar las piezas al formato stl. Este formato sirve para definir la geometría de elementos 3D. Para abrir este fichero se ha utilizado el programa Cura. Gracias a éste se va poder configurar las características deseadas para la impresión, así como el espesor de la pieza, el porcentaje de relleno...

Además, con la ayuda de su interfaz gráfica se puede comprobar que la pieza está lista para ser impresa indicando la posición y orientación de la misma. Una vez se configura la pieza el programa indicará los gramos de hilo que va a utilizar y la duración del proceso de impresión. Tal y como se puede observar en las siguientes imágenes:







## 11. Descripción de los resultados

Para conseguir un proyecto de calidad, así como de aprendizaje, al inicio del mismo se fijaron una serie de objetivos que se han ido cumpliendo con el tiempo. Como el proyecto ha cumplido todos los objetivos propuestos, se puede decir que los resultados han sido los siguientes:

- Programar los distintos sensores (temperatura, distancia y caudal) para un correcto funcionamiento.
- Configurar el Gateway (IoT2020) para la gestión del flujo del programa, además también se ha configurado un servidor en el propio Gateway para la subida de datos de los sensores.
- Además del servidor comentado previamente en el punto anterior, se utiliza otro repositorio para almacenar los datos obtenidos con los sensores en otro servidor proporcionado por ThingSpeak.
- Crear un panel de control para controlar el sistema en todo momento, además, éste también incorpora pequeños gráficos para controlar los valores de los distintos sensores.
- Configurar y conectar los distintos actuadores (bomba 1, bomba 2, resistencia y agitador) en el PLC.
- Diseñar soportes para los sensores y elementos de la maqueta para una posterior impresión en una máquina 3D.
- Montaje de la maqueta para lograr que el proyecto sea realista y así, demostrar que todo lo propuesto en el proyecto se ha llevado a cabo.

Teniendo en cuenta el listado de resultados mencionados anteriormente se ha podido llevar a cabo el proyecto de control de llenado de un tanque de agua con un Arduino y un PLC conectados mediante un Gateway Iot2020.

## 12. Descripción del presupuesto

En este punto se verá la evaluación económica del proyecto, incluido su coste total. Se dividirá en los siguientes apartados:

- Personal:

Para la elaboración del trabajo se necesita el trabajo de un desarrollador. Teniendo en cuenta que un programador junior cobra 6,5€/hora y que el proyecto dura 492 horas, el coste es de 3198€.

- Software:

- **Licencia de Windows 10** gratuita mediante *DreamSpark* (portal de la universidad junto a Microsoft que permite a los estudiantes descarga de software).
- **Gantt Project**: licencia gratuita.
- **Google Chrome**: licencia gratuita.
- **Visual 360**: licencia gratuita.
- **Cura**: licencia gratuita.
- **MQTTlens**: licencia gratuita.
- **Arduino IDE**: licencia gratuita.
- **GitHub IDE**: licencia gratuita.
- **Putty**: licencia gratuita.
- **Eclipse IDE**: licencia gratuita.
- **Dropbox**: licencia gratuita.
- **ThingSpeak**: licencia gratuita con limitación de datos.

**Gastos totales de software: 0€**

- Hardware:

- **Portátil Apple**: valorado en 1000€ con una vida media de 1 año. Haciendo uso prácticamente total del ordenador durante la elaboración del TFG, tanto para desarrollo como para documentación. Con un uso dedicado al TFG del 33% a lo largo del proyecto. Amortización anual:  $1000/1 = 1000€$ . Gasto del ordenador:  $((1000 * 14\text{semanas}) * 0.33/52\text{semanas}) = \mathbf{88.85 \text{ euros}}$ .

- **Pecera; precio: 21€/unidad**
- **LM35DZ; precio: 1.99€/unidad**
- **Arduino MEGA; precio: 35.25€/unidad**
- **Ethernet Shield; precio: 4.53€/unidad**
- **Condensador; precio: 10€/unidad**
- **Resistencia 250 Ohm; precio: 1.6€/unidad**
- **Bomba; precio: 10,85€/unidad**
- **Agitador; precio: 10€/unidad**
- **Calentador; precio: 15€/unidad**
- **Pegamento PVC; precio: 4.97€/unidad**
- **Tubo PVC; precio: 5€/m**

**Gastos totales de hardware: 209,04€**

<b><u>Gastos</u></b>	<b><u>Coste</u></b>
Personal	3198
Software	0
Hardware	209.04
Total	3407.04

## 13. Conclusiones

Desde un punto de vista técnico, se ha de subrayar el acierto de Siemens al ofrecer un producto, el IoT2020, que permite a sus usuarios disponer de una pasarela que puede dar servicio a nodos de sensorización de bajo coste basados en Arduino para proveer de información a un conjunto de PLCs. Esto permite que personas con competencias transversales en el ámbito de la electrónica y la informática, pero sin los conocimientos específicos de una determinada marca de PLCs (por ejemplo, Siemens), sean capaces de aportar soluciones económicas y funcionales a un problema concreto de automatización como es control del llenado de un tanque propuesto en este TFG. En este sentido, conviene reflexionar acerca del tiempo de formación que se hubiera requerido para desarrollar el mismo proyecto en TIA Portal, el cual, se estima, hubiera sido mucho mayor.

Asimismo, también se de destacar la gran ayuda que presenta la librería de nodos y el paradigma programación de Node-Red a la hora de leer y publicar datos provenientes/dirigidos de/a dispositivos y/o servicios distribuidos en el fog o la nube, y gestionar flujos de eventos de forma gráfica. Esto permite que el programador pueda centrarse en el diseño y personalización de la solución en lugar de en la programación de bloques básicos de bajo nivel, lo que da lugar a un desarrollo del proyecto mucho más ágil, sencillo y efectivo tal y como el autor de este TFG ha podido constatar.

Desde un punto de vista personal, se ha de destacar que el Grado en Ingeniería Informática de Gestión y Sistemas de Información no ofrece una formación ad-hoc, desde el punto de vista de las competencias específicas (contenidos), para abordar proyectos ubicados en el marco del IoT e Industry 4.0. No obstante, también hay que destacar que la formación recibida es lo suficientemente generalista como para que, a partir de la formación base recibida y a través del aprendizaje personal, un estudiante de este Grado sea capaz de hacer frente a este tipo de proyectos.

## 14. Bibliografía

Adafruit Industries, Unique & fun DIY electronics and kits. Recuperado de <https://www.adafruit.com/>

Aprendiendo Arduino. *Aprendiendo a manejar Arduino en profundidad*. Recuperado de <https://aprendiendoarduino.wordpress.com/>

Bricogeek. Placas Arduino, Robótica, Electrónica, Raspeberry Pi. Recuperado de <http://tienda.bricogeek.com/>

Del Valle Hernández, L. Proyectos IoT con Arduino, las plataformas más importantes. *Programar fácil*. Recuperado de <https://programarfácil.com/podcast/proyectos-iot-con-arduino/> [Consulta: 13/01/2018]

HIVEMQ (Enterprise MQTT Broker). *The Seven Best MQTT Client Tools*. Recuperado de <https://www.hivemq.com/blog/seven-best-mqtt-client-tools>

JavaScript Screencasts and Tutorials. Recuperado de <https://www.javascripttuts.com/>

Morrissey, D (25/04/2010). Sleeping Arduino-Part 1. *Arduino, Zigbee and Embedded Development*. Blogger. Recuperado de <http://donalmmorrissey.blogspot.com.es/2010/04/putting-arduino-diecimila-to-sleep-part.html>

Stack Overflow. Recuperado de <https://stackoverflow.com/>

Sparkfun Electronics (2003). Recuperado de <https://www.sparkfun.com/>

The MathWorks, Inc (1994-2018). Recuperado de <https://es.mathworks.com/>

MQTT. Recuperado de <http://mqtt.org/>

Node-RED. Recuperado de <https://nodered.org/>

SIMATIC S7-1200. Recuperado de [https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/controladores\\_modulares/controlador\\_basico\\_s71200/pages/s7-1200.aspx](https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/controladores_modulares/controlador_basico_s71200/pages/s7-1200.aspx)

SIMATIC IoT2020. Recuperado de <https://w3.siemens.com/mcms/sce/en/simatic-iot2020/pages/default.aspx>

LM35. Recuperado de <http://www.ti.com/lit/ds/symlink/lm35.pdf>

Water flow sensor. Recuperado de <http://www.hobbytronics.co.uk/yf-s201-water-flow-meter>

Sensor distancia ultrasonidos. Recuperado de <http://www.tesensors.com/es/es/product/reference/XX918A3F1M12/>

TIA Portal. Recuperado de  
<https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/tia-portal/pages/tiaportal.aspx>