



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Diseño de una arquitectura IoT para el control de sistemas hidropónicos

MEMORIA PRESENTADA POR:

Sara Hinojosa Pinto

GRADO DE INGENIERÍA INFORMÁTICA

Autor: Sara Hinojosa Pinto

Tutor: Juan Ramón Rufino

Convocatoria de defensa: *Junio 2019*

Diseño de una arquitectura IoT para el control de sistemas hidropónicos

Sara Hinojosa Pinto

Resumen

Vivimos en un sistema en el que cada vez se valora más la alimentación sana así como el respeto por el medio ambiente y los recursos naturales. Es por ello que cada vez son más utilizadas alternativas como los cultivos hidropónicos, caracterizados por utilizar los recursos justos que necesitan las plantas evitando derrochar tanto agua, como nutrientes.

De acuerdo a lo anteriormente expuesto, los principales objetivos de este trabajo son: acercar al usuario doméstico los cultivos hidropónicos y crear un sistema capaz de monitorizar los datos del cultivo (niveles de temperatura, Ph, humedad, etc) permitiendo al hidroponicultor saber en qué condiciones se encuentra el sistema en cada momento y de esta manera, ayudarle a tomar decisiones para optimizar el rendimiento y calidad de su cosecha.

Palabras clave

Internet de las cosas (IoT), hidroponía, sensores, monitorización.

Dirección

Juan Ramón Rufino

Contenido

1. Introducción	4
1.1. Antecedentes	4
1.2. Objetivos	4
2. Estado del arte	5
2.1. Hidroponía	5
2.1.1. Técnicas de hidroponía	6
2.1.2. Técnica NFT (Nutrient Film Technique)	6
Componentes y materiales necesarios para técnica NFT	6
2.1.3. Ventajas e inconvenientes del cultivo hidropónico	7
2.3. Empresas	8
Ikea	8
GroHo y Tecniccat	8
Verdifresh (proveedor de Mercadona)	8
2.4. IOT y la producción agrícola	9
3. Implementación	9
3.1. Estructura física	12
3.2. Sistema de recirculación del agua	12
3.3. Tecnología	13
3.3.1. Orange Pi	13
Conexiones al Arduino	14
3.3.2. Arduino NANO (CH340)	14
3.3.3. Sensores temperatura DS18B20	15
Protocolo 1-Wire	15
Configuración de los sensores	16
Lectura de los valores leídos por el sensor	18
Conexiones al Arduino	18
3.3.4. Sensor de humedad DHT11	18
Lectura de los valores leídos por el sensor	19
Conexiones al Arduino	19
3.3.5. Sensor de luz BH1750	19
Lectura de los valores leídos por el sensor	21
Conexiones al Arduino	21
3.3.6. Bomba de agua	21
LEER Y modificar estado de la bomba	22
Conexiones al Arduino	22
3.3.7. Sensor de Ph 4502C	23
Lectura de los valores leídos por el sensor	24
Conexiones al Arduino	25
3.3.8. Buzzer	25
Notificaciones sonoras	25
Conexiones al Arduino	25
3.3.9. Sistema de iluminación	26
Leer y modificar estado del Sistema de iluminación	26
3.3.10. Nivel agua	26
Lectura de los valores leídos por el sensor	27
Conexiones al Arduino	27



3.3.11. Circuito completo	27
3.4. Software	28
3.4.1. Arduino NANO	28
Librería DallasTemperature	28
LIBRERÍA DHT11	29
Librería BH1750	29
Driver ch340	29
Comunicación entre el Arduino y la Orange Pi	29
3.4.2. Configuración de la Orange Pi	31
Instalación y Configuración de apache, PHP y Mysql	32
Instalación de servicio de correo electrónico	33
Creación de la base de datos	33
Daemon para comunicación con el arduino	35
Servicios para la aplicación web	37
3.4.3. Aplicación web.	39
Estructura de ficheros	39
Diseño y funcionamiento de la aplicación	39
Pantalla Mi hidropónico	42
Pantalla Niveles y notificaciones	44
Pantalla Histórico	44
Pantalla Configuración	45
4. Implementación práctica	45
4.1. Montaje del circuito	45
4.2. Montaje de la estructura	45
5. Conclusiones	46
5.1. Conclusiones personales	46
5.2. Posibles aplicaciones y ampliaciones del sistema	46
6. Presupuesto	47
7. Bibliografía	49
Anexo 1. Herramienta nmap para escaneo de dispositivos en la red wifi	52
Anexo 2. Protocolo MQTT (Message Queue Telemetry Transport)	52
Anexo 3. Código del Daemon en la Orange Pi para la comunicación con el Arduino.	54
Anexo 4. Código del Arduino.	54
Anexo 5. Código de la página web.	54
Anexo 6. Programas y entornos de programación utilizados	54

1. Introducción

1.1. Antecedentes

Tras finalizar todas las asignaturas de Grado de Ingeniería Informática de la Universidad Politécnica de Valencia en el campus de Alcoy, desarrollo el siguiente proyecto cuya propuesta es el diseño de una arquitectura IoT para el control de sistemas hidropónicos así como la implementación de una aplicación web para facilitar la configuración y monitorización del sistema.

Fue en una de las asignaturas del último curso donde nos hablaron del IoT o *Internet de las Cosas*, y me llamó mucho la atención el hecho de integrar tecnología en objetos de la vida cotidiana para que estos se interconecten entre ellos a través de Internet. Es entonces cuando pensando posibles implementaciones me vinieron a la cabeza los huertos urbanos y decidí investigar por esta rama.

El motivo principal por el que he decidido realizar este trabajo viene determinado por el deseo de profundizar en el desarrollo de aplicaciones que unan hardware y software y desarrollar herramientas de utilidad que hagan más fácil la vida de la gente, junto a un interés personal en una alimentación más sana a través de cultivar en casa.

Por lo tanto, me propuse hacer una combinación de todos estos temas, con el objetivo de desarrollar un sistema hidropónico desde cero, creando una aplicación web para controlar el estado del sistema, tratar datos entre los distintos componentes mediante los protocolos adecuados y promover los cultivos sostenibles en nuestras casas. Es por esto, que considero que este trabajo me permitirá consolidar y aplicar los conocimientos adquiridos durante mis estudios a una aplicación real.

1.2. Objetivos

- ▶ Facilitar y acercar al usuario doméstico el cultivo hidropónico de sus propios alimentos.
- ▶ Captar datos (temperatura, humedad relativa, Ph, etc) de un sistema hidropónico para el autoconsumo.
- ▶ Almacenar los datos para su posterior consulta por el usuario mediante una aplicación web.

2. Estado del arte

Antes de comenzar con el desarrollo del trabajo va a introducirse qué es un sistema hidropónico y presentarse algunas empresas que ya han apostado por estos sistemas, tanto para actividades industriales y producciones masivas de productos agroalimentarios, como para usuarios finales.

Hoy en día están muy de moda los sistemas ecológicos y cada vez más las ciudades buscan ideas con las que renovarse y tener más 'espacios verdes', es por ello que muchas de ellas han implementado sistemas hidropónicos en las fachadas, tejados de los edificios e incluso en parques, automatizando su mantenimiento [1].

Por otra parte, cada vez las personas se están preocupando más por el origen de los alimentos que consumen y muchos están optando por cultivar en sus propias terrazas o áticos a nivel individual o vecinal fomentado por muchos ayuntamientos y asociaciones de vecinos.

2.1. Hidroponía

Se trata de una técnica que permite cultivar y producir plantas sin emplear suelo o tierra, con esta técnica se puede obtener hortalizas de excelente calidad y sanidad, asegurando un uso más eficiente del agua y fertilizantes. Los rendimientos por unidad de área cultivada son altos, por la mayor densidad y elevada productividad de la planta.

Los primeros trabajos formales sobre este sistema de producción comenzaron en fechas cercanas al año 1600, no obstante, la historia de la hidroponía se remonta a 3000 a.C en los famosos jardines colgantes de la antigua Babilonia, considerados como una de las siete Maravillas del Mundo antiguo, los cuales se alimentaban del agua que corría por medio de canales. Asimismo, hace más de 1000 años ya se practicaba la hidroponía en China, India, los aztecas en Mexico y los Egipcios a orillas del río Nilo.

Algunas de las primeras informaciones sobre cultivo hidropónico fueron escritas por Francis Bacon en 1627, sugiriendo que el principal papel de las raíces es absorber agua y nutrientes no anclar la planta en el suelo. En 1699 John Woodward hizo crecer plantas en diversos recipientes con medio líquido. Sachs 1860 y Knop 1861 lograron aislar una planta por completo del suelo haciendo que crezca por medio de una solución de elementos minerales. Pero fue el profesor de fisiología vegetal de la Universidad de California, Dr. William Gerike, el cual durante los años 1929 y 1930 potenció la aplicación de la técnica con fines comerciales.

Hoy en día los cultivos hidropónicos se han extendido por todo el mundo, incluso llegando al espacio, pues La NASA está desarrollando programas de cultivo hidropónico en el espacio.

2.1.1. Técnicas de hidroponía

En el mercado encontramos varios tipos de técnicas de hidroponía, entre las que destacan las siguientes [5, 6]:

- NFT (técnica de película nutritiva), que consiste en hacer circular el agua por tubos de PVC con bombas de agua y con el uso de sistemas automatizados.
- Raíz flotante, utilizan contenedores de diversos materiales y las plantas quedan por encima del agua flotando, mientras que las raíces están sumergidas en el agua. En este caso la oxigenación del agua se hace de forma manual.
- Sustrato sólido, la raíz de la planta se sostiene con el sustrato, que aporta humedad y drena el agua que aporta la solución nutritiva.
- Aeroponía, técnica en la que las raíces de la planta quedan suspendidas en el aire, y son pulverizadas constantemente con el agua que contiene los nutrientes, de manera que tiene tanto los nutrientes que necesita como oxígeno.

La técnica propuesta en el proyecto es la técnica NFT, quedando expuesta a continuación, las anteriormente mencionadas quedan fuera de los alcances del proyecto.

2.1.2. Técnica NFT (Nutrient Film Technique)

El sistema de recirculación de solución nutritiva fue desarrollado en el Glasshouse Crop Research Institute, Inglaterra, en la década de los sesenta [4, 7].

Las plantas se colocan en unos agujeros realizados a unos tubos de PVC, de manera que las raíces se quedan ocultas y con contacto directo con el agua que circula por el sistema con los nutrientes disueltos, y las hojas y el fruto de la planta quedan por fuera.

Dichos tubos necesitan contar con un desnivel para que sea posible la recirculación de la solución nutritiva. Este sistema destaca la producción de lechugas con un alto número de rotaciones anuales y la de tomates, con un período extendido de producción que permite la obtención de muy altos rendimientos.

Una de las ventajas que ofrece el sistema NFT es su mayor eficiencia en cuanto a la utilización de los elementos minerales esenciales para el crecimiento de las plantas, de agua y oxígeno, ya que maximiza el contacto directo de las raíces con solución nutritiva que es constantemente renovada por lo que el crecimiento va más acelerado siendo posible obtener en el año más ciclos de cultivo.

COMPONENTES Y MATERIALES NECESARIOS PARA TÉCNICA NFT

A continuación se detallan las características más relevantes de cada elemento y los materiales factibles de utilizar para su implementación.

- Estanque colector: almacena el agua con la solución nutritiva para suministrar al cultivo. El volumen del estanque debe escogerse en función del número de plantas, especies a cultivar y modalidad de corrección química de la solución nutritiva (sistema de corrección manual o automático). Debe permanecer cubierto para evitar el desarrollo de algas, las cuales consumen oxígeno de la solución, aumentan la degradación de compuestos químicos y favorecen su contaminación con restos orgánicos.
- Canales de cultivo: tiene dos funciones, la sujeción de las plantas y que la solución nutritiva pase a través de ellos alimentándolas.
- Bomba: uno de los componentes claves del sistema, pues se encargará de impulsar la solución nutritiva desde el estanque colector hasta la parte alta de los canales de cultivo. Dependiendo de la magnitud del módulo de producción y grado de supervisión deberían considerarse dispositivos de alarma que indiquen un mal funcionamiento (falta de agua, fallo de la bomba, fallo eléctrico, etc).
- Red de distribución: tuberías y mangueras desde la bomba impulsora hacia la parte superior de los canales de cultivo.
- Tubería colectora: recoge la solución nutritiva desde los canales de cultivo y la lleva de retorno hacia el estanque colector.

2.1.3. Ventajas e inconvenientes del cultivo hidropónico

Entre las principales ventajas que presentan los cultivos hidropónicos encontramos que se trata de cultivos sanos y responsables con el medio ambiente, ya que se mantienen con las dosis necesarias de agua y nutrientes para las plantas sin desperdiciar recursos que de otro modo serían absorbidos por la tierra si son utilizados con exceso cuando se riegan los cultivos tradicionales [2].

Son aptos para lugares pequeños, así como paredes o terrazas. Ya que pueden cultivarse de manera vertical.

Presentan un mayor rendimiento en menos tiempo. Según estudios, su tasa de crecimiento puede llegar a ser entre un 15 y un 25% más rápida, ya que las plantas disponen de todo lo necesario para crecer más saludables y de esta manera producir más y de mejor calidad.

Dichos sistemas se pueden construir a partir de materiales reciclados, sin necesidad de invertir grandes cantidades de dinero para montarlos.

Se reducen los gastos tanto en fertilizantes como en herbicidas, preparación del suelo y por lo tanto en mano de obra.

En los cultivos hidropónicos se cultiva con gran facilidad lechugas, albahaca, perejil, tomates cherry, pimientos habaneros, espinacas, fresas e incluso plantas florales como orquídeas [3].

En cuanto a inconvenientes mencionar que cada planta debe ser tratada manual e individualmente. Un fallo en el sistema de recirculación del agua podría echar a perder todo el cultivo, de ahí la utilidad de este proyecto para

monitorizar el sistema, notificando al propietario antes de que esto ocurra.

2.3. Empresas

IKEA

Cualquiera puede tener un jardín. Así es como presenta IKEA el lanzamiento de un huerto hidropónico para cultivar alimentos en casa durante todo el año [8]. Su sistema permite germinar y hacer crecer las plantas sin suelo, ya que las semillas brotan en unos tapones de espuma absorbente que incluye el sistema, manteniéndolas húmedas. Dicha bandeja incorpora un sensor de agua para que el usuario sepa cuándo necesitan más agua las plantas. El precio de estos kits completos con abonos y semillas oscilan entre los 50 y 150 euros.



Figura 1. KRYDDA/VÄXER - IKEA

GROHO Y TECNICCAT

GroHo [9] se dedica a la investigación y producción de equipos hidropónicos. Desarrollando nuevos productos especializados en la hidroponía y dando soluciones agrícolas sin suelo, tanto horizontales como verticales. Entre sus actividades, proporcionan soluciones para el cultivo de alimentos caseros, sanos, instalan sistemas autosostenibles en cualquier lugar y por último, forman a las personas y las conciencian de la importancia de la sostenibilidad a través de talleres, cursos y seminarios.

Por otra parte, Tecniccat [10], se encarga de añadir en los equipos de GroHo la parte de electrónica permitiendo a los usuarios controlar el sistema mediante controladores. Cabe destacar que esta empresa no se dedica únicamente a esta actividad, si no que están especializados en el tratamiento de aguas y diseño electrónico para el servicio y bienestar de las personas.

Ambas empresas han sido una fuente de inspiración para la realización del trabajo, Groho por la simplicidad de sus productos y el hecho de que estén fabricados con materiales reciclados y Tecniccat por sus controladores, aportando nuevas ideas para implementar en el hardware y software que he desarrollado.

VERDIFRESH (PROVEEDOR DE MERCADONA)

El proveedor de ensaladas de Mercadona [12] ha puesto en marcha un sistema de cultivo hidropónico, que no utiliza tierra convencional y ahorra hasta un sesenta por ciento de agua. El grupo producirá 10.000 kilos de verduras y hortalizas a la semana con este nuevo método.

2.4. IOT y la producción agrícola

En las últimas dos décadas, la tecnología ha tenido un papel fundamental en el desarrollo de este sector. El primer avance fue la automatización de los sistemas de fertilización, y 20 años después, gracias a la digitalización y conectividad de los invernaderos, se consigue, cultivar más y mejor y además, minimizar el impacto de la agricultura en el medio ambiente. De manera, que lo que antes se hacía por intuición o experiencia del agricultor y el técnico, ahora se puede hacer basándose en los datos recogidos por los sensores creando un clima óptimo dentro de los invernaderos [13].

Si bien es cierto, los agricultores “de toda la vida” son más reacios al uso de nuevas tecnologías, pero son los más jóvenes los que están apostando por estas innovaciones, un claro ejemplo es Eurosol, que tiene fincas propias con invernaderos multitúnel, cultivo hidropónico, fertilización automatizada por bandejas de demanda y control de las ventilaciones en función de los datos recogidos.

Pretender alcanzar una agricultura sostenible, económica, social y medioambientalmente solo es posible digitalizando los procesos de los diferentes eslabones de la cadena con el fin de optimizarlos y hacerlos más eficientes.

También se están desarrollando softwares de ayuda a la toma de decisiones (DSS) que procesan los datos y “aprenden” con el paso del tiempo y los convierte en información para ayudar en tomar las medidas correctas.

3. Implementación

El sistema se puede dividir en cuatro grupos

- ▶ Por una parte está la **estructura física**, que será donde estén las propias plantas e incluye todo el sistema de recirculación del agua, tanto las tuberías como el estanque colector.
- ▶ Los **sensores** (temperatura, Ph, humedad, luminosidad, nivel de agua), y **actuadores** (relés) comandados por un microcontrolador, que tomará muestras de los diferentes sensores y actuará sobre la bomba de recirculación del agua y el sistema de iluminación. Para este trabajo se ha escogido el Arduino, que facilita la interacción con el hardware. Este transferirá dicha información a un equipo remoto y actuará cuando este se lo pida.
- ▶ Un **equipo remoto** que consta de un sistema de comunicación, con alimentación. En este proyecto se ha utilizado una Orange Pi, que se comunicará con el Arduino para almacenar toda la información que éste le pase en una base de datos, enviará comandos para modificar los estados de la bomba o el sistema de iluminación y actuará como servidor para mostrar en una aplicación web toda esta información.
- ▶ **Aplicación web** que muestra al usuario los datos de su sistema y le permite configurar tiempos de riego e iluminación. Adicionalmente permite al usua-



rio registrar en la base de datos información individualizada de cada una de sus plantas.

Otra posible implementación sería utilizar un servidor y que la Orange Pi solo hiciese de intermediaria entre el Arduino y el servidor, pudiendo también almacenar los datos en una base de datos local por si se perdiese la comunicación con el servidor, esta comunicación sería mediante el protocolo MQTT. En el **Anexo 2** se explica brevemente este protocolo y cómo implementarlo, pero en este proyecto no se ha llevado a cabo de esta manera, ya que se ha utilizado la Orange Pi como servidor.

A continuación se muestra un esquema con todos los elementos del sistema y sus conexiones:

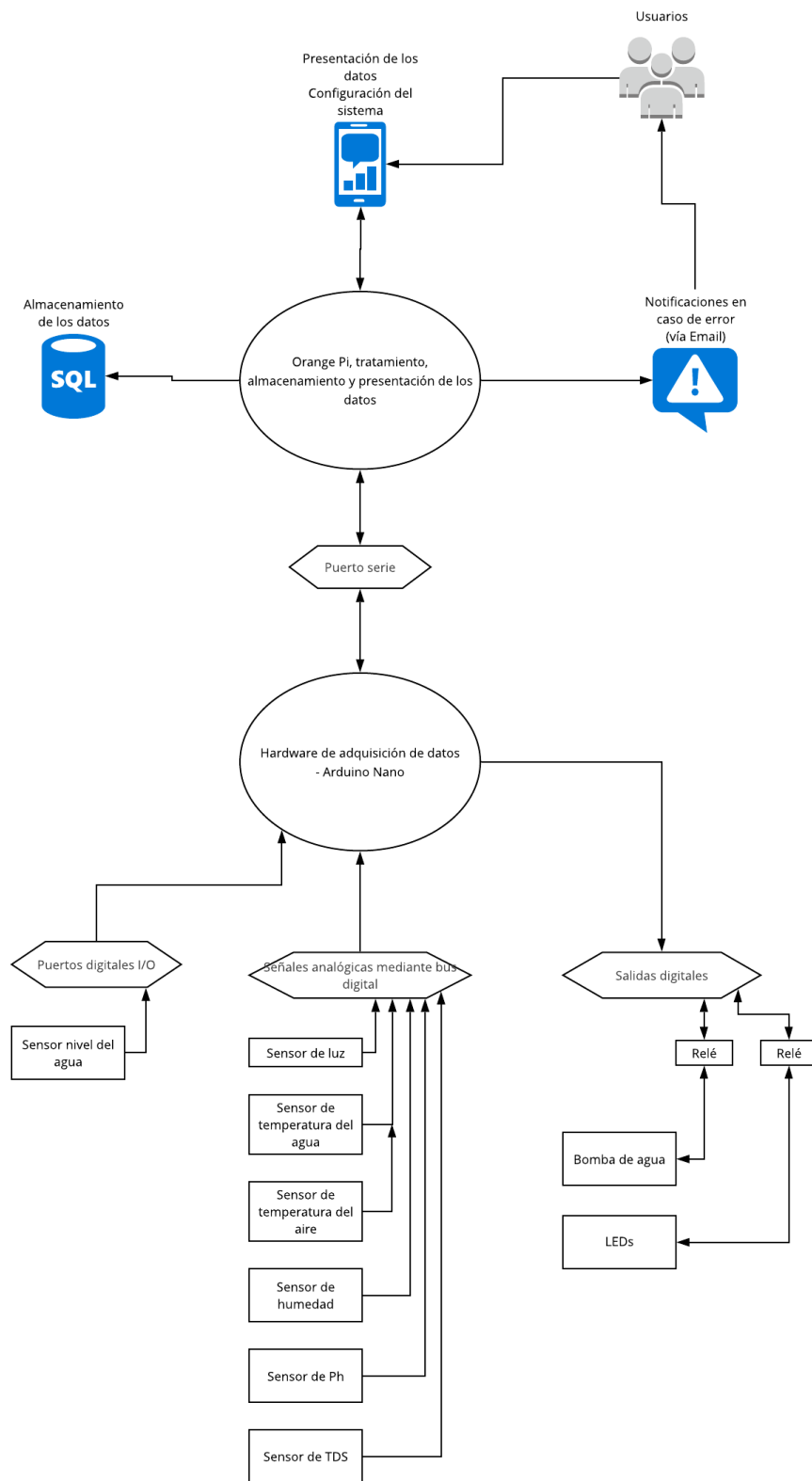


Figura 2. Esquema elementos del sistema

3.1. Estructura física

Se ha decidido montar una estructura física vertical en la que las plantas van en unos agujeros realizados a los tubos de PVC y sujetas mediante esponjas de manera que quedaría algo como lo siguiente.



Figura 3. Prototipo de la estructura física

3.2. Sistema de recirculación del agua

Sobre la estructura física se han añadido mangueras conectando los tubos para permitir la circulación del agua con los nutrientes, en la parte derecha estará el desagüe, y por la izquierda será por donde entra el agua a cada uno de los tubos. En la parte inferior se localiza el estanque recolector con la bomba de agua, junto a los sensores, y actuadores utilizados.

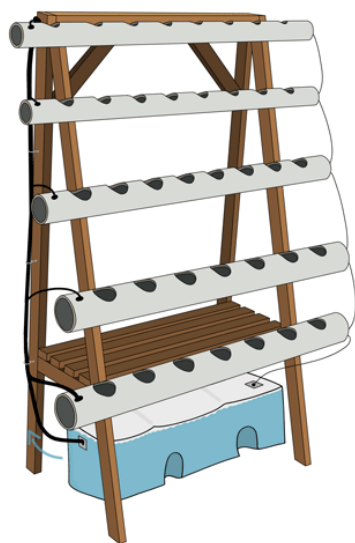


Figura 4. Prototipo con sistema de riego



Figura 5. Prototipo final con plantas

3.3. Tecnología

Para obtener los datos de todos los sensores, así como, cambiar el estado de algunos de los elementos del sistema, se ha diseñado un circuito que interconecta todos los componentes.

A modo de resumen, el Arduino recibirá los datos de los sensores, se los enviará a la Orange Pi por el puerto serie para que esta los almacene y muestre en la aplicación web. Además, se encargará de actuar sobre los controladores según lo que reciba de la Orange Pi.

A continuación se explican detalladamente cada uno de los elementos utilizados para controlar y automatizar el sistema.

3.3.1. Orange Pi

Una Orange Pi es un mini ordenador de bajo coste y de tamaño compacto. Tiene todas las prestaciones que puede aportar cualquier ordenador, conexiones USB, Ethernet y HDMI.

Para la realización del proyecto se ha utilizado una Orange Pi Lite, que nos ofrece las prestaciones suficientes para la realización del proyecto. Se trata de una versión económica que tiene similares características que la Raspberry Pi 3 B+ a un precio inferior.

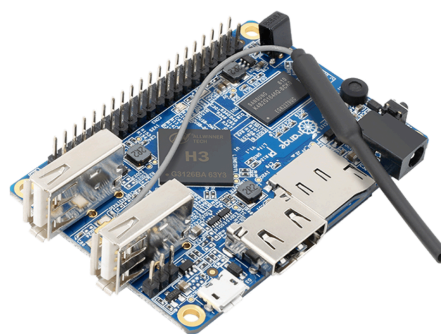


Figura 6. Orange Pi

En la siguiente tabla se observan las principales diferencias entre la Raspberry Pi 3 B+ frente a Orange Pi Lite [16]:

	Raspberry Pi 3 B+	Orange Pi Lite
Precio	35 US\$	12 US\$
CPU	ARM Cortex-A53 (64-bit) 1.2GHz quad core	ARM Cortex-A7 (32-bit) 1.2GHz quad core
GPU	VideoCore IV	Mali-400 MP2
RAM size	1GB	512MB
SD card	MicroSD card slot Max. size: 32GB	MicroSD card slot Max. size: 64GB
Ethernet	1 port Type: 10/100	X
Audio connector	3.5 mm	X
WiFi	802.11b/g/n	Built-in
Bluetooth	Bluetooth 4.1	X

Este tipo de dispositivos se pueden utilizar para infinidad de aplicaciones como servidores de contenido conectados al televisor, intermediarios entre la parte electrónica de un sistema y el servidor que almacena y presenta la información, etcétera. Pero, en este proyecto se utiliza para, por una parte comunicarse con el Arduino tanto para obtener los datos como para modificar el estado de la bomba o del sistema de iluminación, y por otra parte almacenará todos los datos que obtenga el Arduino para presentarlos en una aplicación web.

CONEXIONES AL ARDUINO

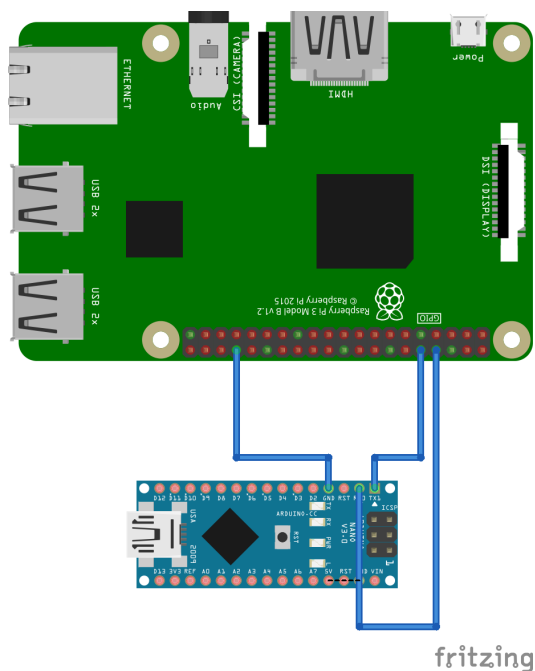


Figura 7. Circuito conexión Orange Pi al Arduino

3.3.2. Arduino NANO (CH340)

Los Arduinos son plataformas hardware y software de código abierto, basados en una placa sencilla con entradas y salidas digitales y analógicas. Están mucho más limitados que los mini ordenadores descritos anteriormente, pero también es cierto que tienen otro propósito, por lo que con ellos también se pueden hacer proyectos de cualquier tipo. Al tratarse de un software abierto, la comunidad de Arduino comparte muchas librerías para diversos sensores y componentes electrónicos por lo que facilitan el desarrollo de proyectos [17].

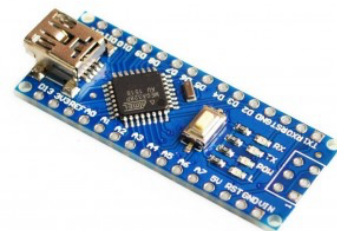


Figura 8. Arduino

Para la realización del proyecto se ha escogido un CH340, se trata de una versión más económica que el Arduino NANO original, pero a pesar de ello, con similares características. Dispone de varios pines digitales y analógicos

y un conector mini USB.

En este proyecto cumplirá la función de recoger la información de los sensores y transmitírsela a la Orange Pi, así como mandar los comandos que reciba desde la Orange Pi a los actuadores del sistema.

3.3.3. Sensores temperatura DS18B20

Se trata de sensores idóneos cuando se desea medir la temperatura en lugares húmedo o incluso del agua, por lo que se han utilizado dos, uno para medir la temperatura del aire, y otro que estará introducido en el depósito del agua, para medir la del agua. Estos sensores se comunican con el Arduino mediante un protocolo poco común, el 1-Wire y tienen una resolución configurable de 9, 10, 11, o 12 bits [19].



Figura 9. Sensor de temperatura

Dispone de 3 pines útiles, el VCC para alimentarlos, GND que es la toma de tierra y DQ que es el pin de datos, por donde se recibirá toda la información. Como bien dice el nombre del protocolo, se utiliza un único cable/pin para conectar varios sensores al Arduino.

Estos sensores tienen un rango de temperatura de entre -55°C y 125°C , pero como todos, tienen errores de medición, y el error no es el mismo en todo el rango abarcado. De modo que para temperaturas de entre -10°C y 85°C puede haber un error de $\pm 0,5^{\circ}\text{C}$ y para el resto de valores de $\pm 2^{\circ}\text{C}$. Para este proyecto no es algo demasiado significativo, ya que el sistema no estará expuesto a temperaturas tan extremas, pero debe tenerse en cuenta.

Por otra parte, disponen de una memoria de 8 bytes, para almacenar el código de identificación único e inalterable del sensor, necesario para identificarlos dentro del bus 1-Wire de comunicación. Internamente dispone de otro espacio de memoria, que entre otras cosas, utiliza el sistema de verificación de redundancia cíclica CRC para la detección de errores en los datos.

También dispone de dos alarmas que se disparan si la temperatura es mayor, o menor, a un umbral de temperatura establecido.

PROTOCOLO 1-WIRE

Se trata de un bus de bajo coste basado en un microcontrolador que se comunica digitalmente sobre un cable con los componentes 1-Wire. Participa un maestro y varios esclavos de una sola línea de la que se alimentan y hacen uso para la transferencia de los datos, y por supuesto, todos ellos conectados a la misma toma de tierra [21].

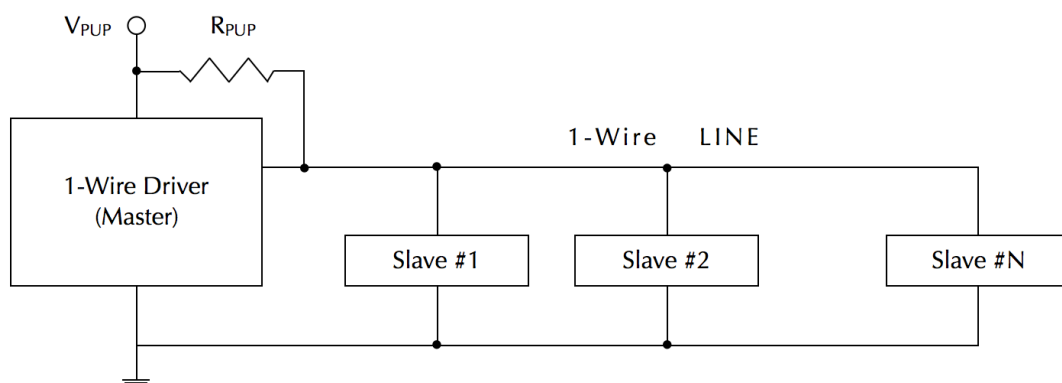


Figura 10. Conexiones dispositivos protocolo 1-Wire

En primer lugar el maestro comienza y controla la comunicación mediante bloques temporales de 60 microsegundos. Los esclavos se sincronizan con el reloj del maestro a través de la línea.

Esencialmente, la comunicación se realiza en tres pasos:

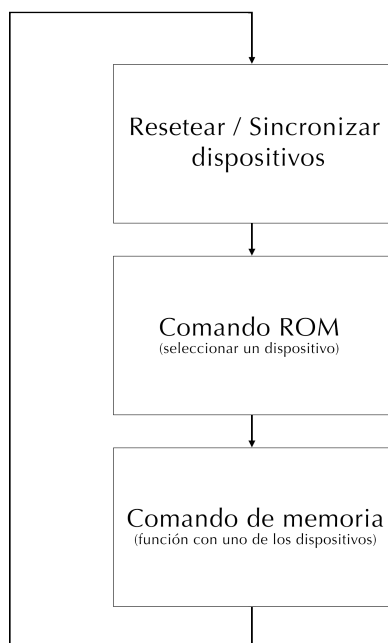


Figura 11. Pasos comunicación 1-Wire

Más adelante, junto a la explicación de la librería DallasTemperature utilizada, se comentan los posibles comandos ROM que proporciona para el uso de estos sensores.

Gracias a este protocolo se consigue robustez en la transmisión de los datos, ya que trabaja con datos digitales mucho menos sensibles al ruido y las señales analógicas y, por lo tanto, permite comunicaciones en distancias más elevadas, garantizando el correcto funcionamiento con hasta 2000 dispositivos.

CONFIGURACIÓN DE LOS SENSORES

Los sensores se conectarán en serie, tantos sensores 1-Wire como quera-

mos, a un único pin del Arduino, por el que se establecerá la comunicación. Tener en cuenta que para la conexión de estos sensores al Arduino es necesaria una resistencia de 4.7 K Ω entre el Vdd y el pin de datos que haga de pull-up, sin ella no se obtienen buenas lecturas del sensor, ya que utiliza un transistor FET de drenaje abierto como una puerta AND, y servirá para tener el pin DQ siempre en modo HIGH [20].

La mejor manera de identificar los sensores es conectarlos por separado al Arduino, para obtener las direcciones únicas de cada uno de ellos, con el método `getAddress()` se obtiene su dirección. El siguiente método lee dichas direcciones.

```
Void direccionSensorTemperatura(int index)
uint8_t deviceAddress[8];
sensors.getAddress(deviceAddress, index);
for (int i = 0; i <8; i++){
    Serial.print(" ");
    Serial.print(deviceAddress[i]);
}
}
```

Fragmento código 1. Método para obtener direcciones de memoria protocolo 1-Wire

Con el método `getDeviceCount()` de la librería se obtiene el número de sensores conectados al bus, y puede servir para comprobar que todo sigue funcionando adecuadamente.

Se han anotado las direcciones para tener claro de qué sensor se van a leer los datos.

Sensor dentro del depósito de agua	0x28 0xFF 0x96 0xF0 0x00 0x17 0x03 0x72 40 255 150 240 0 23 3 114
Sensor colocado fuera	0x28 0xFF 0xCD 0xBD 0x00 0x17 0x03 0xEA 40 255 205 189 0 23 3 234

Tabla 1. Direcciones de memoria sensores de temperatura

Dado que es un poco peligroso leer los valores de los sensores por el índice que ocupan en el bus de comunicación, ya que no hay un control sobre dicho orden, y un cambio en el circuito podría alterar las posiciones, se ha decidido leer cada una de las temperaturas a través de la dirección única de los sensores para estar seguros de que las lecturas provienen del sensor que se desea, por lo tanto la función utilizada para la lectura de las temperaturas es la mostrada a continuación.

LECTURA DE LOS VALORES LEÍDOS POR EL SENSOR

```

float medirTemperatura(DeviceAddress direccionSensor,
  bool &correcto){
  sensors.requestTemperaturesByAddress(direccionSensor);
  correcto = sensors.isConnected(direccionSensor);
  return sensors.getTempC(direccionSensor);
}
  
```

Fragmento código 2. Método para obtener datos de los sensores de temperatura

CONEXIONES AL ARDUINO

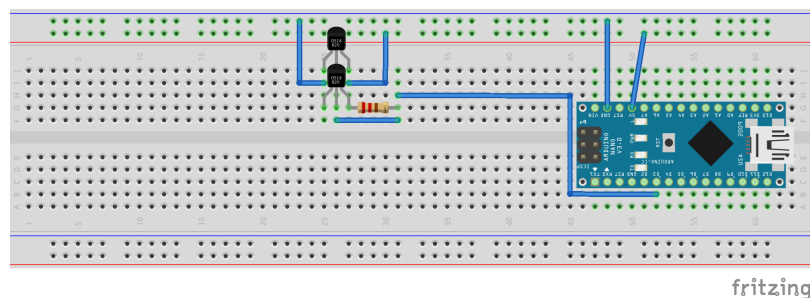


Figura 12. Conexiones sensores de temperatura a Arduino

3.3.4. Sensor de humedad DHT11

Se trata de un sensor que proporciona alta fiabilidad y estabilidad gracias a su señal digital calibrada. Además de leer la humedad, es capaz de medir la temperatura, pero como ya se está tomando esta medida con otros sensores, no se utilizará para eso [22].

Dado se ha escogido un sensor sin PCB, será necesaria una resistencia pull-up de 5 k Ω conectada a la salida digital del Arduino. Y, en ambos casos, es necesaria una alimentación de 3,5V a 5V.

Pines del sensor:

- VCC: alimentación.
- I/O: transmisión de datos
- NC: no conecta, pin al aire
- GND: conexión a tierra

A pesar de conectarse a un pin digital, se trata de un sensor analógico, de manera que es el propio dispositivo el que hace la conversión de analógico a digital, proporcionando una trama de 40 bits correspondiente a la información captada por el sensor:

0011 0101	0000 0000	0001 1000	0000 0000	0100 1001
8 bist de humedad (parte entera)	8 bist de humedad (parte decimal)	8 bist de temperatura (parte entera)	8 bist de temperatura (parte decimal)	Bits de paridad (confirmar datos)

Tabla 2. Trama de bits sensor de temperatura

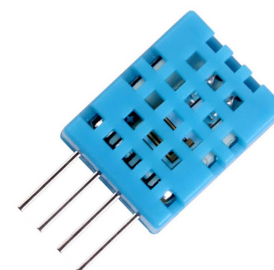


Figura 13. Sensor humedad

Además de medir la temperatura con este sensor, se ha decidido extraer el índice de calor, es decir, la sensación de calor en función de la humedad relativa y la temperatura.

LECTURA DE LOS VALORES LEÍDOS POR EL SENSOR

```

bool medirHumedad(float &humedad, float &indiceCalor){
    humedad = dht.readHumidity(); // Leemos la humedad relativa
    float t = dht.readTemperature(); // Leemos la temperatura en grados centígrados
    (por defecto) Loutilizaremos para obtener el indice de calor

    // Comprobamos si ha habido algún error en la lectura
    if (isnan(humedad)||isnan(t)) {
        return false;
    }

    // Calcular el índice de calor en grados centígrados
    indiceCalor = dht.computeHeatIndex(t, humedad, false);
    return true;
}
  
```

Fragmento código 3. Método para obtener datos de los sensores de humedad

CONEXIONES AL ARDUINO

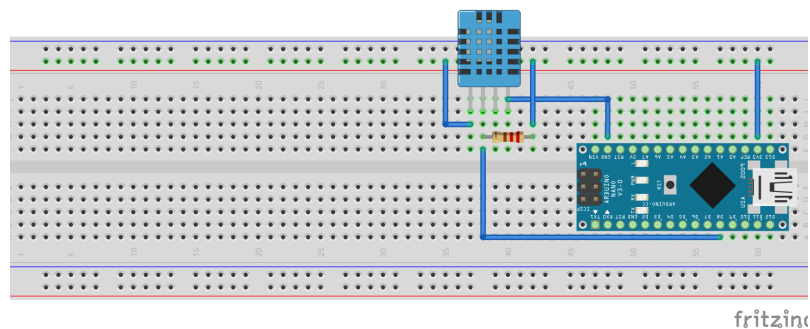


Figura 14. Conexión del sensor de humedad al Arduino

3.3.5. Sensor de luz BH1750

Este sensor proporcionará datos de la cantidad de luz que recibe el sistema a lo largo de los días. Estará localizado junto con el resto de componentes electrónicos. Es útil si queremos encender una serie de LEDs cuando el nivel de lx es inferior a un umbral determinado [23, 24].

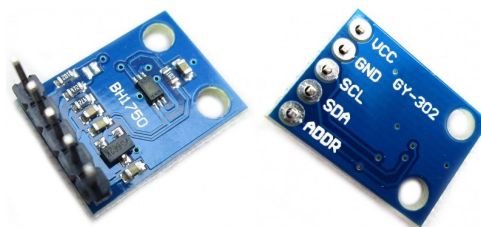


Figura 15. Sensor luminosidad

El sensor BH1750, Se comunica de forma digital con el microcontrolador, con el bus I2C, por lo que es resistente a interferencias si se sitúa a cierta distancia del circuito que lo lee. Su tiempo de respuesta es bastante bajo, menos de 200 ms en las circunstancias más desfavorables y entrega valores

medidos en Lux, unidad de intensidad de iluminación del Sistema Internacional, de símbolo lx, que equivale a la iluminación de una superficie que recibe normal y uniformemente un flujo luminoso de 1 lumen por metro cuadrado. Proporciona una sensibilidad y resolución razonablemente altas, de manera que se comporta frente a la luz visible de una forma equiparable a la del ojo humano y no se ve afectado por la radiación infrarroja ni depende de la temperatura de color del tipo de iluminación, es decir, funciona bien con luz natural y con diferentes tipos de iluminación artificial.

Este módulo tiene un regulador interno de 3.3V, pudiendo alimentarse con 5V sin problemas. A parte de tener los pines de alimentación e I2C, tiene un par de pines más para establecer la dirección.

El pin ADDR tiene internamente una resistencia a GND, por lo que si lo dejamos desconectado la dirección será 0x23. La resolución del sensor puede configurarse en tres modos, cuanto más resolución, mayor tiempo de medición.

Pin ADDR	Dirección I2C
ADDR = HIGH (5v)	0x5C
ADDR = LOW (GND o NC)	0x23

Tabla 3. Pines del sensor de luminosidad

- High resolution Mode2. Resolución de 0.5 lx. 120 ms de tiempo de medición.
- High resolution Mode. Resolución de 1 lx. 120 ms de tiempo de medición.
- Low Resolution Mode. Resolución de 4 lux. 16 ms de tiempo de medición.

Estos modos pueden subdividirse en dos modos de lectura, continua e individual, que se corresponden con dos estados, activo y de bajo consumo o reposo. Si se utiliza el modo de lectura continua el sensor sigue activo tras el muestreo, mientras que de la otra manera, tras realizar una medición individual, entra automáticamente en modo de reposo y bajo consumo. La primera lectura del modo continuo tarda a lo sumo 180 ms y las sucesivas entre 16 ms y 120 ms dependiendo de la resolución. En su hoja de datos se recomienda utilizar la resolución de 1 lx, siendo capaz de distinguir iluminaciones por debajo de los 10 lux (que se corresponde con la luz crepuscular) y que es más inmune al ruido que podría afectar a la medida.

La placa BH1750 utiliza el protocolo I2C para la comunicación, y requiere dos pines para comunicarse con el dispositivo. Para configurar el bus I2C se hace mediante código de usuario, sin necesidad de la librería, esto es así para que se configure una única vez y posteriormente soporte las distintas

opciones para las distintas plataformas. Para la programación y captación de los datos del sensor se utilizará la librería BH1750 escrita por Christopher Laws explicada más adelante.

LECTURA DE LOS VALORES LEÍDOS POR EL SENSOR

```
uint16_t medirLuz(){
  int valor=-1;
  if(luxometreOK){
    valor = lightMeter.readLightLevel();
    if(valor >= 0){
      return valor;
    }
  }
  if(luxometreOK = lightMeter.begin()){ //volvemos a intentar conectar el sensor
    return medirLuz();
  }
  else {
    numErroresLeves++;
    errorSistema += "luz=Error en la lectura de la luminosidad;";
  }
  return -1; //El valor no es correcto
}
```

Fragmento código 4. Método para obtener datos del sensor de luminosidad

CONEXIONES AL ARDUINO

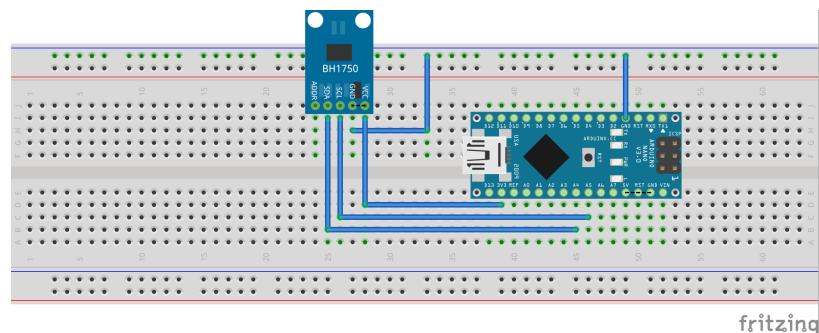


Figura 16. Conexiones del sensor de luminosidad al Arduino

3.3.6. Bomba de agua

Dado que no es necesario elevar el agua a más de 1m de altura, bastará con utilizar una pequeña bomba que irá conectada al resto de la electrónica. El Arduino será el encargado de modificar el estado de la bomba a través de un relé, ya que que la bomba se alimenta a 220V. Las ordenes de conexión las Emitirá la Orange Pi de acuerdo a la configuración del sistema (temporizadores programados o las peticiones del usuario).

LEER Y MODIFICAR ESTADO DE LA BOMBA

```
bool estadoBomba(){  
    return digitalRead(10);  
}  
//-----  
void setBomba(bool newEstado){  
    if(!newEstado){  
        digitalWrite(10, LOW);  
    }  
    else{  
        digitalWrite(10, HIGH);  
    }  
}
```

Fragmento código 5. Métodos para leer y modificar el estado de la bomba

CONEXIONES AL ARDUINO

Se ha sustituido la bomba por un motor.

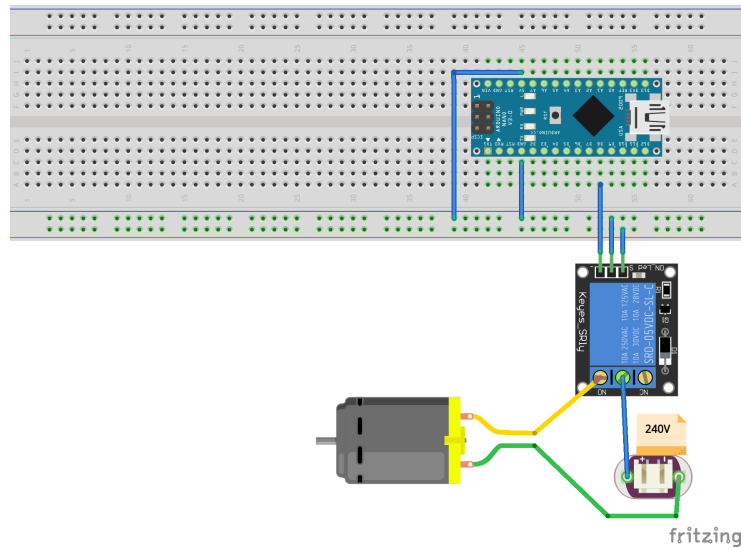


Figura 17. Conexiones de la bomba de agua al Arduino

3.3.7. Sensor de Ph 4502C

Este sensor proporcionará datos sobre la acidez o alcalinidad de una disolución, la escala de pH varía en una escala de 0 a 14. El pH indica la concentración de iones hidrogeno $[H]^+$ presentes en determinadas disoluciones. Este sensor permite medir de forma sencilla el pH de un líquido gracias a su placa controladora que ofrece un valor analógico proporcional a la medición, bastará con leerlo de unas de las entradas ADC del microcontrolador [25, 26].



Figura 18. Sensor de Ph

Los pines que tiene el controlador son los siguientes:

To	Temperatura
Do	Señal del límite de pH
Po	Valor de pH en V
G	Masa del circuito analógico
G	Masa de Alimentación
V+	Alimentación (5V)

Tabla 4. Pines del sensor de Ph

El controlador también dispone de dos potenciómetros que permiten la correcta calibración de la sonda. El que está más pegado al conector BNC regula el offset y el otro el límite de pH.

- Offset: dado que el rango de medida de la sonda oscila entre valores negativos y positivos (el 0 representaría un pH de 7.0, o neutro), el circuito añade un valor de offset al valor medido para poder utilizarlo con el Arduino, de manera que el ADC solo tendrá que tomar muestras de valores de tensión positivos. Por lo tanto forzaremos un pH de 7.0 desconectando la sonda del circuito y cortocircuitando la parte interna del conector BNC con la exterior. Con un multímetro medimos el valor del pin Po y ajustamos el potenciómetro para que sea 2.5V.
- Límite de pH: este potenciómetro sirve para establecer un valor límite al circuito, de manera que el LED rojo se encienda y la señal de pin Do se ponga en ON.

Además se tiene que calcular la conversión del voltaje que dará el sensor de pH, para lo que se necesitan dos valores de referencia de pH y medir el voltaje que devuelve el sensor en el pin Po.

Se han utilizado dos soluciones de pH 3,8 y pH 7.2 y se han obtenido los

voltajes en el pin de entrada del Arduino 2.97V y 2.43V respectivamente. Puesto que se trata de un sensor lineal, tomando ambos puntos se puede deducir la ecuación para convertir el voltaje medido a pH. La fórmula general sería $y=mx+b$, calcular la recta que pasa por ambas coordenadas, obteniendo la ecuación de dicha recta, donde $y=-6.3x+22.5$.

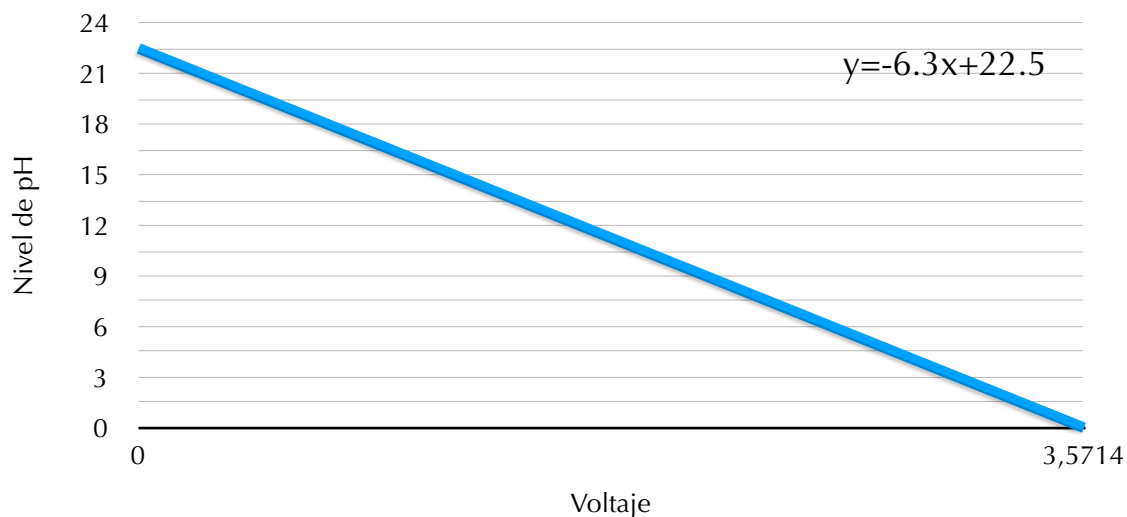


Figura 19. Gráfica ecuación de la recta para obtener los valores medidos por el sensor de Ph

LECTURA DE LOS VALORES LEÍDOS POR EL SENSOR

```

float medirPH() {
  int bufPH[10],
      tempPH;
  unsigned long int avgPH;
  for(int i=0;i<10;i++) {
    bufPH[i]=analogRead(PIN_PH);
    delay(10);
  }
  for(int i=0;i<9;i++) {
    for(int j=i+1;j<10;j++) {
      if(bufPH[i]>bufPH[j]) {
        tempPH=bufPH[i];
        bufPH[i]=bufPH[j];
        bufPH[j]=tempPH;
      }
    }
  }
  avgPH=0;
  for(int i=2;i<8;i++)
    avgPH+=bufPH[i];
  float pHVol=(float)avgPH*5.0/1024/6;
  float pHValue = -6.30 * pHVol + 22.5;

  return pHValue;
}
  
```

Fragmento código 6. Métodos para obtener los valores de Ph

CONEXIONES AL ARDUINO

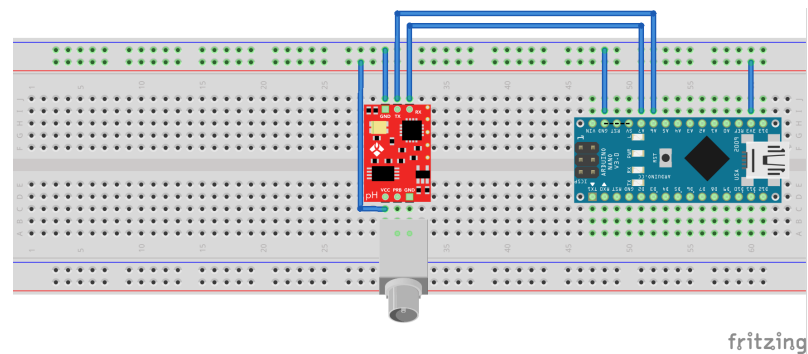


Figura 20. Conexiones del sensor de Ph al Arduino

Para la conexión del circuito al Arduino será necesarias dos entradas analógicas, alimentación y dos GND, que aunque que en la placa estén separados, puede utilizarse la misma.

3.3.8. Buzzer

Se trata de un generador de sonido que emitirá avisos en caso de que algo deje de funcionar como se espera. Podrá programarse para que suene, por ejemplo, cuando el nivel del agua esté por debajo de lo necesario para que el sistema funcione adecuadamente [27].

NOTIFICACIONES SONORAS

```

void beep(int duracion) {
  digitalWrite(12, HIGH);
  delay(duracion);
  digitalWrite(12, LOW);
}
//-----
void notificarError(){
  for (int i = 0; i < numErroresLeves;i++) {
    beep(10);
  }
  for (int i = 0; i < numErroresGraves; i++) {
    beep(300);
  }
}
  
```

Fragmento código 7. Método para emitir notificaciones sonoras

CONEXIONES AL ARDUINO

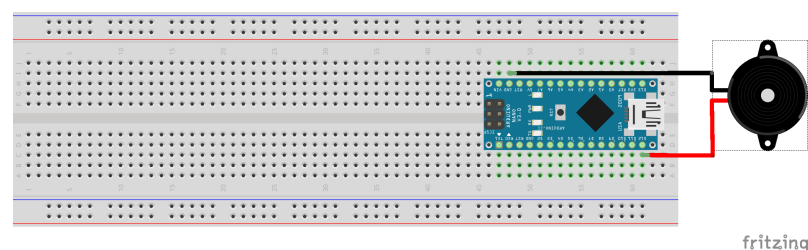


Figura 21. Conexiones del buzzer al Arduino

3.3.9. Sistema de iluminación

Colocados en la parte superior de la estructura y conectado a la electrónica, podrá programarse a la hora que se desea que se enciendan automáticamente, o encenderse en caso de que el sensor de luz detecte una luminosidad por debajo a un umbral previamente establecido.

LEER Y MODIFICAR ESTADO DEL SISTEMA DE ILUMINACIÓN

```

void setLeds(bool newEstado){
  if(!newEstado){
    digitalWrite(9, LOW);
#ifdef DEBUG
    Serial.println(F("Apago LEDs"));
#endif
  }
  else{
    digitalWrite(9, HIGH);
#ifdef DEBUG
    Serial.println(F("Enciendo LEDs"));
#endif
  }
}
//-----
bool getLeds(){
  return digitalRead(9);
}
  
```

Fragmento código 8. Método para encender / apagar sistema de iluminación

3.3.10. Nivel agua

Para la medición del nivel del agua, se utilizará un sensor compuesto por un flotador que funciona como un interruptor. Puede colocarse en la parte superior del depósito, de manera que ajustándolo como queramos, en cuanto el nivel del agua esté lo suficiente bajo como para no elevar el flotador se accione el interruptor. También podría colocarse en el fondo del depósito para que cuando se vacíe cambie de estado del interruptor, pero en este caso no interesa este funcionamiento. Por lo tanto se colocará como se muestra a continuación:



Figura 22. Flotador con interruptor

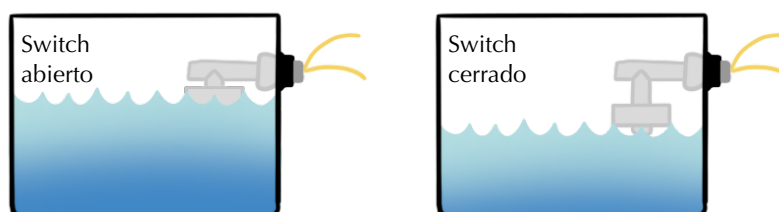


Figura 23. Estados del medidor del nivel de agua

A la hora de conectarlo al Arduino se hace de la misma manera que un interruptor, uno de los pines a Vcc y el otro a un pin digital, el cual debe configurarse como pull-up, es decir, con la resistencia interna del pin del Arduino [28].

LECTURA DE LOS VALORES LEÍDOS POR EL SENSOR

```

bool medirNivelAgua(){
  return digitalRead(11);
}
  
```

Fragmento código 9. Método para saber nivel del agua

CONEXIONES AL ARDUINO

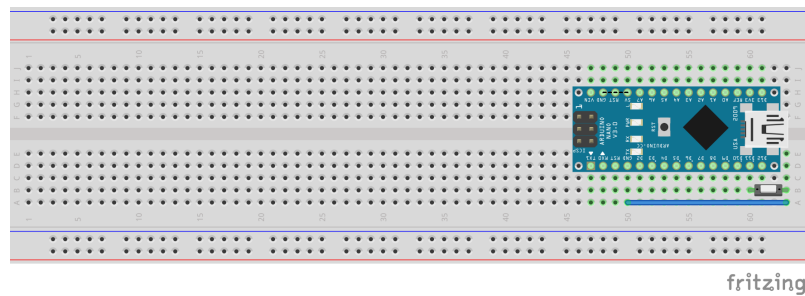


Figura 24. Conexiones del nivel de agua al Arduino

3.3.11. Circuito completo

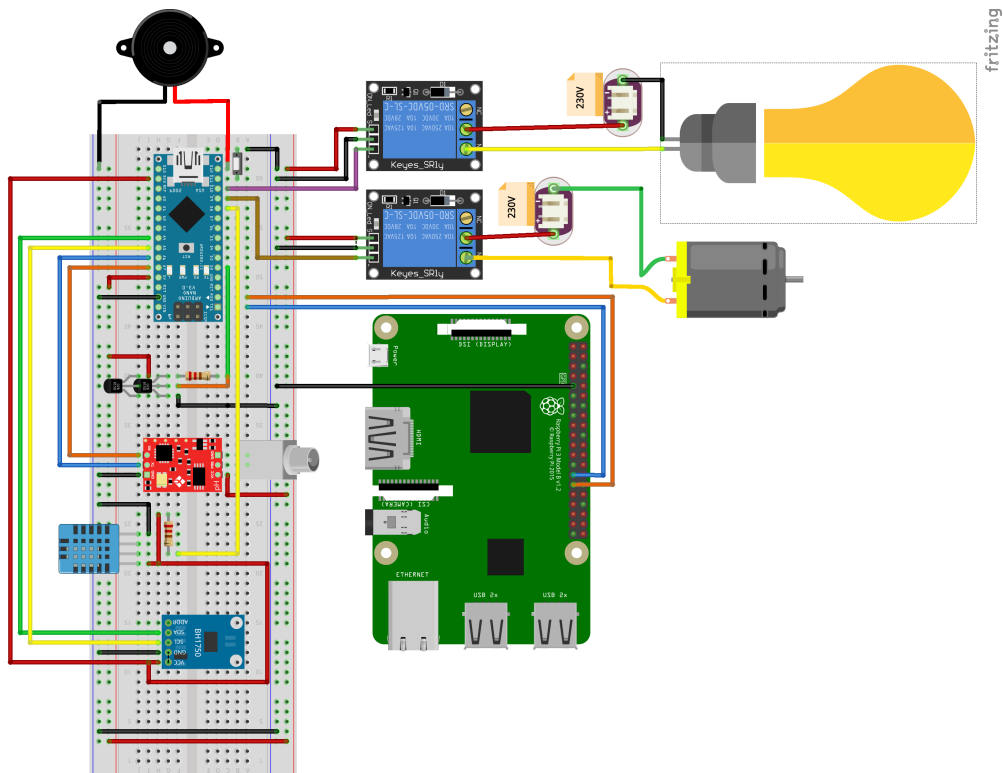


Figura 25. Circuito completo

3.4. Software

3.4.1. Arduino NANO

LIBRERÍA DALLASTEMPERATURE

Se trata de una librería de la comunidad de Arduino, que facilita y simplifica a los nuevos usuarios su uso e implementación par el uso de dispositivos 1-Wire, en este caso, los sensores de temperatura utilizados [29].

Para instalar la librería se han seguido los siguientes pasos:

1. Descomprimir el contenido de la librería en mi caso */Users/Sara/Documents/Arduino*. Esta ubicación se encuentra en el software de Arduino File>Preferences.
2. Después de copiar el directorio de DallasTemperature en *\Libraries*, reiniciar el software.
3. La librería debe de aparecer en Programa > Incluir Librería > Contributed.

Los métodos que incluye la librería y que en su mayoría, se han utilizado para el desarrollo del software para la lectura de los datos de los sensores se comentan a continuación:

- *uint8_t getDeviceCount(void)* Devuelve el número de dispositivos conectados al bus 1-Wire. Sirve para saber que todos los sensores están conectados correctamente.
- *bool isConnected(const uint8_t*)* Comprueba si un sensor está conectado al bus 1-Wire dado una dirección única
- *uint8_t getResolution()* Obtiene la resolución global de todos los sensores. Devuelve un valor entero sin signo de 8-bit con la resolución, ésta puede ser de 9-bit, 10-bit, 11-bit o 12-bit.
- *void setResolution(uint8_t)* Establece la resolución de un sensor dado su índice en el bus 1-Wire. No devuelve ningún valor
- *uint8_t getResolution(const uint8_t*)* Devuelve el valor de la resolución de un sensor dado su dirección única.
- *void requestTemperatures(void)* Envía los comandos a los sensores para que hagan la lectura de la temperatura.
- *bool requestTemperaturesByAddress(const uint8_t*)* Igual que la función anterior pero este solo envía los comandos para que un solo dispositivo haga la lectura de temperatura dado una dirección única
- *float getTempC(const uint8_t*)* Obtiene la temperatura en grados Celsius dada la dirección única del sensor.
- *float getTempF(const uint8_t*)* Obtiene la temperatura en grados Fahrenheit dada la dirección única del sensor.

LIBRERÍA DHT11

Descargada del propio entorno Arduino.

Librería para obtener los datos del sensor de humedad utilizado (también podría utilizarse con el DHT22) [30].

LIBRERÍA BH1750

Descargada de <https://www.arduino-libraries.info/libraries/bh1750-fvi> [31].

Librería que se encarga de gestionar las comunicaciones en el bus I2C para obtener los datos de la medición de la luz ambiental. A continuación se explica las distintas funciones que presenta la librería:

- `begin(mode)` Inicializa el módulo, es decir, inicializa la comunicación I2C de Arduino y configura la resolución del BH1750, los modos disponibles se describen en el apartado `_____sensor bh1750_____`. Si no se especifica el modo, por defecto, se configura con el modo `BH1750_CONTINUOUS_HIGH_RES_MODE` (alta resolución continua).
- `configure(mode)` Como el método anterior, pero con la diferencia de que esta función no inicializa la comunicación I2C, por lo tanto, ya deberá estar inicializada para poder utilizar esta función. Los modos son los mismos mencionados anteriormente.
- `readLightLevel()` Realiza una lectura de la iluminancia a la que está expuesta el sensor, devuelve un entero con el valor de la medición en Lux.

DRIVER CH340

Para poder utilizar la placa ch340, ha sido necesario instalar el Driver correspondiente. Se ha obtenido de la página <https://sparks.gogo.co.nz/ch340.html> y seguido las instrucciones que se indican para su instalación. Esto permitirá la comunicación por el puerto serie con la placa para poder programarla.

COMUNICACIÓN ENTRE EL ARDUINO Y LA ORANGE PI

Para el intercambio de datos entre la Orange Pi y el Arduino, se utilizará el puerto serie. La Orange Pi podrá o bien solicitar información al Arduino, o modificar el estado de cualquiera de los actuadores del sistema, como puede ser la bomba de agua o los LEDs. Además se encargará de enviar al Arduino los valores de los parámetros de la configuración, que revisará periódicamente, y en caso de haber algún cambio desde la última consulta en la base de datos volverá a enviarle la configuración. Se adjunta el código encargado de realizar tal tarea en el Anexo 3.

Los tipos de mensajes que enviará el Arduino son

Enviar datos del sistema

- Datos nombreProyecto: `sensor0=valor0;nombreSensor1=valor1;`

- Ejemplo: `Datos TFG: tempAire=21;temAgua=16;bomba=1;`

Solicitar la configuración

- Config nombreDelSistema

- Ejemplo: `config TFG`

Envío de errores

- Error nombreDelSistema: `sensor0=msajeError0; sensor1=msajeError1;`

- Ejemplo: `Error TFG: agua=El nivel del agua está bajo;`

En cambio la OrangePi, puede

Mandar mensajes para solicitar datos:

- Ejemplo: `get;`

Modificar el estado de los actuadores:

- Ejemplo: `set bomba=ON;`

Enviar la configuration:

- *Config: parámetro0=valor0/parámetro1=valor1/*

- Ejemplo: `config: tiempoEntreErrores=20/estadoBomba=ON`

Convenio de nombres asignados a los distintos elementos del sistema.

Sensor	Nombre	Valores
Sensor de temperatura del aire	tempAire	Valor en coma flotante
Sensor de temperatura del agua	tempAgua	Valor en coma flotante
Sensor de PH	pH	Valor en coma flotante
Nivel del agua	agua	1 nivel alto, 0 nivel bajo
Estado de la bomba	bomba	1 encendida, 0 apagada
Estado de los LEDs	leds	1 encendidos, 0 apagados
Sensor de Luz	luz	Valor entero
Sensor de humedad	humedad	Valor en coma flotante
Estado del sistema	sistema	OK todo bien, ERR algo mal

Tabla 5. Convenio de nombres de los distintos elementos del sistema

Convenio de nombres de los parámetros de configuración del sistema.

Variable en el Arduino - nombre	Valor por defecto	Valores
tiempoEsperaBomba	1 minuto	Valor entero
tiempoEntreErrores	30 segundos	Valor entero
envioErrores	sí	Valor booleano
alarmaSonora	sí	Valor booleano
estadoBomba	sí	Valor booleano
estadoLeds	sí	Valor booleano

Tabla 6. Convenio de nombres de los distintos parámetros de configuración del sistema

3.4.2. Configuración de la Orange Pi

Para utilizar la Orange Pi se ha instalado el sistema operativo Armbian, que está basado en Debian y Ubuntu y nos permite instalar los servicios necesarios.

El sistema de almacenamiento es una tarjeta Micro SD, que sería el equivalente al disco duro de un ordenador convencional. Para instalar el sistema operativo, grabamos en la tarjeta una imagen del sistema especialmente compilada para la Orange Pi utilizada descargada desde la página www.armbian.com. Basta con insertar la tarjeta en la ranura de la placa, encenderla y empezar a utilizarlo.

La Orange Pi Lite no tiene conexión Ethernet, por lo que para configurarla debe establecerse la conexión a través del puerto serie en modo consola, de este modo, realizaremos la conexión del Wifi posteriormente explicada con más detalle, y así poder acceder por SSH para realizar el resto de configuraciones.

Para configurar el wifi se utiliza la herramienta nmtui (Network Manager Text User Interface) del siguiente modo:

- Edit a connection >> Add y añadir una nueva conexión, nombre del wifi, contraseñas, etc. Además, editar la conexión para que la IP asignada sea siempre fija y no tener que averiguar la IP cada vez que se establezca la conexión [32].

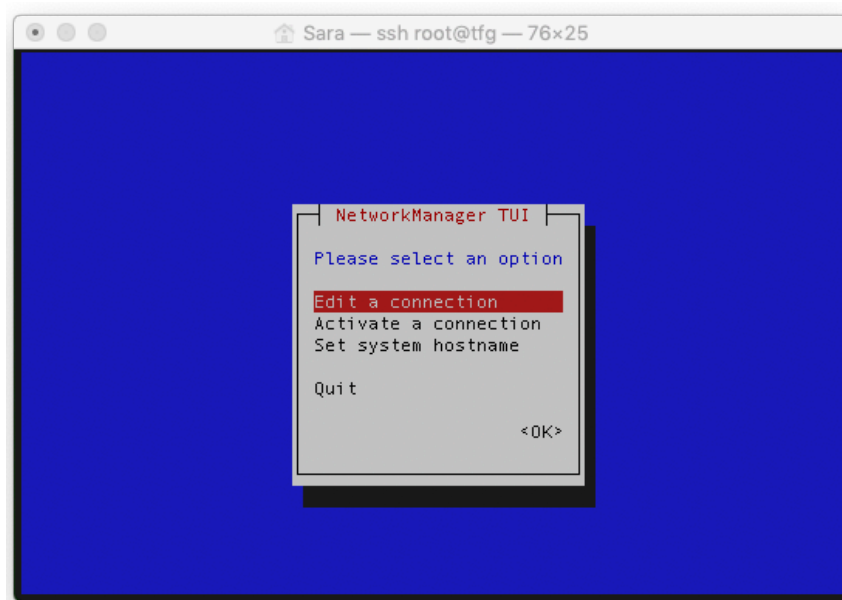


Figura 26. NetWorkManager TUI

De esta manera, cuando se encienda la Orange Pi, detectará la red wifi e intentará conectarse a ella automáticamente. Se ha configurado con una IP privada a la que desde el exterior no hay conexión al sistema.

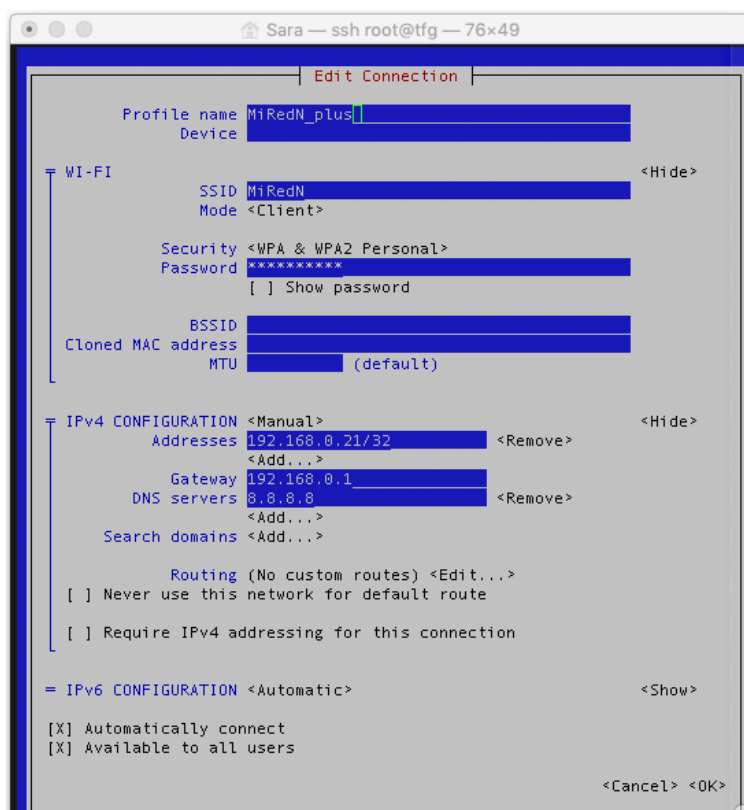


Figura 27. Configuración wifi Orange Pi

Armbian utiliza el gestor de paquetes Aptitude que permite fácilmente instalar aplicaciones y servicios del sistema. Se utilizarán paquetes estándar del repositorio de Armbian.

INSTALACIÓN Y CONFIGURACIÓN DE APACHE, PHP Y MYSQL

Apache es un servidor web de código abierto ampliamente utilizado por millones de aplicaciones web. Permite utilizar PHP como un módulo y facilita la iteración con bases de datos, en este caso gestionadas por el servidor MySQL. En la 'jerga' de Internet, esta configuración se conoce como LAMP (Linux, Apache MySQL y PHP).

En Armbian la instalación de los servicios se realiza con el gestor de paquetes Aptitude y en un solo comando pueden instalarse todos los servicios necesarios.

```
apt-get install apache2 mysql-server mysql-client php php-mysql
```

Fragmento código 10. Comando para instalar servicios necesarios en la Orange Pi

Una vez finalizado el comando ya están instalados con la configuración básica todos los servicios necesarios.

Para personalizar el Apache, crear el fichero `/etc/apache2/sites-enabled/000-tfg.conf` e indicar el puerto por el que va a responder (80), y los directorios donde se encuentran (que fueron creados previamente):

- DocumentRoot: /www/tfg/html
- ErrorLog: /www/tfg/logs/error.log
- CustomLog: /www/tfg/logs/access.log combine

Modificar el fichero /etc/apache2/sites-enabled y añadir 000-tfg.conf, se añaden los tres ceros delante del nombre, porque cuando se interpreta el fichero, se recorren en orden alfabético.

Para que todo este se haga efectivo, reiniciar Apache:

```
service apache2 reload
```

Fragmento código 11. Comando para reiniciar Apache

INSTALACIÓN DE SERVICIO DE CORREO ELECTRÓNICO

Dado que se desea el que sistema notifique al usuario de problemas en su hidropónico, se ha decidido hacerlo a través de correo electrónico. Para ello utilizaremos el sendmail, popular agente de correo que se integra perfectamente en sistemas Debian. La instalación de éste se realiza con el comando siguiente:

```
apt-get install sendmail
```

Fragmento código 12. Comando para instalar agente de correo electrónico

Con esto ya tenemos el sistema preparado para enviar emails a cualquier usuario desde la función `mail()` de PHP.

CREACIÓN DE LA BASE DE DATOS

La base de datos se ha planteado de tal manera que permite que el proyecto sea escalable, ya que con el campo Proyecto que se ha añadido a cada una de las tablas, se pretende tener separados los datos de cada uno de los sistemas que se quieran añadir al proyecto.

- Tabla Usuarios. Permite que haya un control de la gente que tiene acceso a la página web. En futuras versiones, podrían añadirse roles con distintos permisos.

Usuarios						
Field	Type	Null	Key	Default	Extra	
id	int(11)	NO	PRI	NULL	auto_increment	
nombre	char(20)	YES		NULL		
clave	char(20)	YES		NULL		

Tabla 7. Descripción de la tabla Usuarios

- Tabla Datos. Almacena todos los datos leídos por los sensores del sistema, el campo proyecto se ha añadido para que sea escalable, de manera que en caso de haber varios sistemas, sus datos estarían separados por dicho campo, aunque todos en una misma base de datos.

Datos					
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
proyecto	char(16)	YES	MUL	NULL	
parametro	char(16)	YES	MUL	NULL	
valor	text	YES		NULL	
fecha	datetime	YES	MUL	NULL	

Tabla 8. Descripción de la tabla Datos

- Tabla Configuración. Incluye todos los parámetros de la configuración, que al usuario le puede interesar cambiar. Por ejemplo, que el Arduino mande periódicamente mensajes o emita un pitido en caso de tener errores.

Configuración					
Field	Type	Null	Key	Default	Extra
proyecto	char(16)	YES	MUL	NULL	
ambito	char(16)	YES		NULL	
parametro	char(32)	YES	MUL	NULL	
valor	text	YES		NULL	
nombre	char(32)	YES		NULL	
tipo	char(16)	YES		NULL	
descripcion	char(128)	YES		NULL	

Tabla 10. Descripción de la tabla configuración

- Tabla Hidropónico. Información de las plantas que hay en el hidropónico que se muestra en la página principal.

Hidropónico					
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
proyecto	char(16)	YES	MUL	NULL	
fila	char(16)	YES	MUL	NULL	
columna	char(16)	YES		NULL	
tipoPlanta	char(16)	YES		NULL	
fechaPlantada	char(32)	YES	MUL	NULL	
comentarios	text	YES		NULL	

Tabla 11. Descripción de la tabla Hidropónico

- Tabla Plantas. Contiene todas las plantas disponibles para añadir a la figura del hidropónico en la web.

Plantas					
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
proyecto	char(16)	YES	MUL	NULL	
nombreImagen	char(32)	YES	MUL	NULL	
nombrePlanta	char(32)	YES		NULL	
identificador	char(16)	YES	MUL	NULL	

Tabla 12. Descripción de la tabla Plantas

DAEMON PARA COMUNICACIÓN CON EL ARDUINO

Un daemon, en términos informativos, se trata de un proceso especial , o servicio, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario. En este proyecto, su función es establecer una conexión bidireccional con el Arduino, tanto para obtener los datos como para configurarlo, o modificar los estados de los actuadores del sistema.

Se ha decidido programarlo en PHP al igual que la aplicación web, y de este modo poder reutilizar software.

Para ello, crear el fichero serialINO.php (**Anexo 3**), posteriormente habrá que cambiar los permisos y hacerlo ejecutable con `chmod +x`.

Ir al directorio `etc/systemd/system` (servicios de la distribución de Linux) y añadir la configuración del nuevo servicio que se ha creado con el daemon serialINO, llamarlo con el mismo nombre pero cambiando la extensión. El contenido de `serialINO.service` el el siguiente:

```
[Unit]
Description=TFG - conexión Arduino
After=mysql.service
[Service]
ExecStart=/www/tfg/daemon/serialINO.php
Restart=always
[Install]
WantedBy=multi-user.target
```

Fragmento código 13. Contenido del fichero `serialINO.service`

Lo que se indica en esta configuración es:

- Descripción, no es necesaria, pero es conveniente no dejarla en blanco ya que así se pueda saber qué es lo que hace el servicio de una manera fácil.
- 'After' sirve para indicar qué servicios deben haberse lanzado previamente para que este pueda lanzarse, en nuestro caso, será necesario mysql ya que el servicio que se ha creado modifica y consulta la base de datos.
- 'ExecStart' indicar dónde se encuentra el código que debe ejecutar el servicio.
- 'Restart' se ha puesto en always para que cada vez que el servicio se caiga, se intente reiniciar automáticamente.
- 'WantedBy' después de habilitar el servicio, se crea un enlace simbólico del servicio para que el systemd cuando arranque el nivel de ejecución indicado inicie el servicio automáticamente. Al inhabilitar el servicio, este enlace será eliminado.

Habilitamos el servicio

```
systemctl enable serialINO.service
```

Fragmento código 14. Comando para habilitar el servicio

A partir de este momento, cada vez que se arranque el sistema se ejecutará automáticamente el servicio.

Consultar su estado

```
service serialINO status
```

Fragmento código 15. Comando para consultar el estado del servicio

Iniciar el servicio

```
service serialINO start
```

Fragmento código 16. Comando para iniciar el servicio

Ya que el daemon se ejecuta como un servicio del sistema su salida estándar es capturada por el sistema y guardada en los registros del sistema. Podemos monitorizarlo de este modo.

```
cd /var/log  
tail -f syslog | grep serialINO
```

Fragmento código 17. Comando para monitorizar output del daemon

Cada vez que se realice un cambio en el daemon, habrá que reiniciar el servicio.

```
service serialINO restart
```

Fragmento código 18. Comando para reiniciar el daemon

El daemon se encargará de:

- Solicitar al Arduino periódicamente información del sistema.
- Escuchar por el puerto serie por si el Arduino solicita la configuración. O enviársela en caso de que haya habido algún cambio.
- Escuchar por el puerto serie por si el Arduino ha mandado algún mensaje de error o de datos y, en estos casos se introducirlos en la tabla de datos.

SERVICIOS PARA LA APLICACIÓN WEB

A fin de conectar la aplicación web con la base de datos se han creado una serie de servicios utilizando Java Script y PHP.

La estructura es la siguiente.

```
function seleccionarPlanta(x, y, nombrePlanta){
  data = {
    x:x,
    y:y,
    planta:nombrePlanta
  }
  $.ajax({
    url: "getPlantas.php",
    type: "post",
    data: data,
    success: function(obj) {
      editarPlanta(x, y, obj.planta);
    },
    error: function() {
      console.log("Error");
    },
    dataType : 'json'
  });
}
```

Fragmento código 19. Ejemplo de servicio con llamada AJAX

Desde una función de JavaScript, con una llamada AJAX, llamar a un fichero .php, en este caso getPlantas.php. La idea de este ejemplo es que pasándole una posición en el hidropónico (fila, columna) y en nombre de la planta que desea 'plantarse' el servicio introduzca dicha planta en la posición indicada.

El siguiente fragmento de código realiza la inserción en la base de datos, y devuelve a la anterior función, en formato JSON la respuesta obtenida de MySQL.



```
mysqli("insert into hidroponico (proyecto, fila, columna, tipoPlanta) values
( 'TFG', '$_POST[x]', '$_POST[y]', '$_POST[planta]');"");

$r=mysqli_fetch_array(mysqli("select nombrePlanta from plantas where
identificador='$_POST[planta]');""));

class Params {
    function Params($estado, $name) {
        $this->estado = $estado;
        $this->planta = $name;
    }
}
$myObj = new Params(1, $r['nombrePlanta']);
$myJSON = json_encode($myObj);
echo $myJSON;
```

Fragmento código 20. Código PHP para insertar datos en la base de datos

3.4.3. Aplicación web.

Se ha desarrollado una aplicación web simple donde el propietario del hidropónico podrá consultar los datos de su sistema así como cambiar la configuración del mismo, o las preferencias de notificaciones.

ESTRUCTURA DE FICHEROS

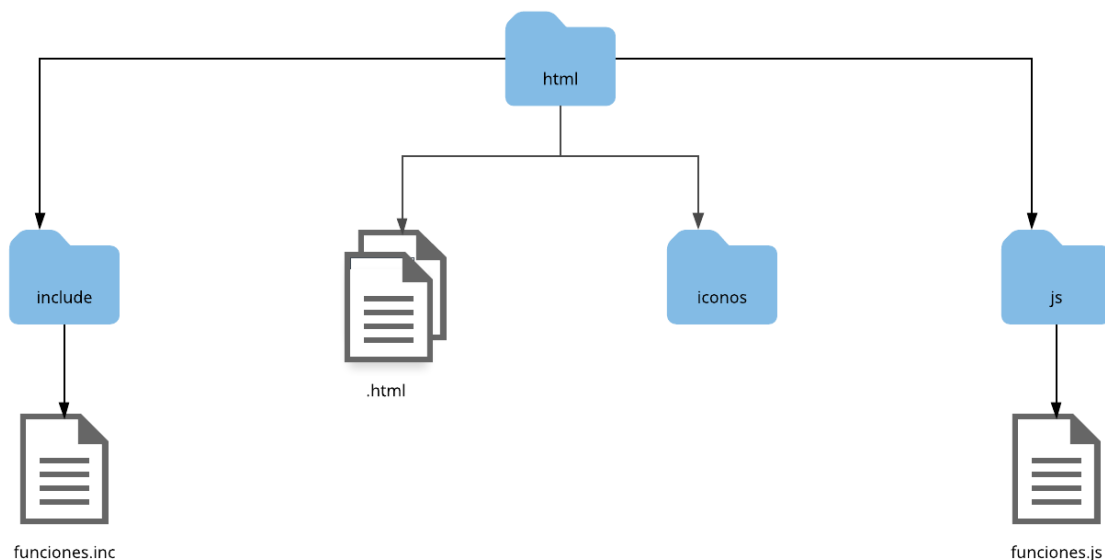


Figura 28. Ficheros aplicación web en la Orange Pi

Se ha creado un directorio html donde se encuentran todos los ficheros necesarios para la pagina web, así como las librerías utilizadas.

/html/include/funciones.inc: funciones generales como acceso a la bbdd, identificación de usuario, cabeceras y las librerías utilizadas, tanto por el daemon como desde le aplicación web.

/html/include/*.html: ficheros html de la aplicación, los cuales incluyen el fichero anterior para mostrar cabeceras, hacer el control de accesos a la página.

/html/iconos/: donde se localizan todas las imágenes de las plantas e iconos que se utilizan en la web.

/htmls/js/funciones.js: fichero que implementa los servicios encargados de acceder a la base de datos y que son consumidos desde los html.

/html/js/highcharts: librería con las funciones necesarias para crear las graficas utilizadas en la web.

DISEÑO Y FUNCIONAMIENTO DE LA APLICACIÓN

En la parte superior se muestra un menú que permite acceder a cada una de las ventanas de la aplicación web.

Diseño de una arquitectura IoT para el control de sistemas hidropónicos

MI HIDROPÓNICO ESTADO ACTUAL DEL SISTEMA HISTÓRICO NOTIFICACIONES Y CONFIGURACIÓN

Figura 29. Cabecera de la aplicación web (común en todas las ventanas)

- Inicio de sesión: únicamente podrán acceder a la web aquellas personas que dispongan de un usuario y contraseña.



Figura 29. Inicio de sesión

- Mi Hidropónico, página principal: muestra un hidropónico con sus respectivos huecos disponibles para seleccionar la planta que se ha plantado.
 - El usuario podrá, añadir una nueva planta al hueco donde pulse.
 - En caso de haber alguna planta en el hueco, mostrará la información de dicha planta (nombre, fecha en la que se plantó, cuidados especiales, comentarios) desde aquí también podrá modificar las características de la planta o eliminarla).
- **En caso de querer cambiar el tipo de planta, tendrá que eliminarse la actual y posteriormente añadir una nueva.
- Niveles y notificaciones: información general del sistema, solo muestra datos, cualquier cambio en las preferencias del sistema, se hará desde la siguiente página.
 - Temperatura del agua y aire, humedad, Ph, TDS, luminosidad, estado de los LEDs, nivel del agua, estado de la bomba y notificaciones.
- Histórico: se muestra en una tabla los datos del periodo de tiempo seleccionado (diario, semanal, mensual, anual, y todo). Se podrá seleccionar un máximo de 3 parámetros para mostrar en la tabla, es posible mostrar todos los que deseemos, pero se ha decidido limitar a 3 para que sea mas legible la información en la gráfica. Desde la parte superior, se seleccionan los parámetros a mostrar y posteriormente, habrá que dar al botón 'Recargar gráfica' para que se refresque la gráfica. **Los datos de la grafica no se ac-



tualizan automáticamente, por lo que si se añadiesen nuevos datos en la base de datos no se verán reflejados en esta ventana hasta que no actualice.

- Configuración: mostrará todos los posibles parámetros configurables del sistema. Permite la posibilidad de restablecer los cambios y volver a la configuración por defecto.
 - Estado de la bomba, intervalo entre funcionamiento de la bomba, tiempo de funcionamiento de la bomba.
 - Luz, configurar con atardecer y amanecer, establecer periodos de tiempo, encendido automático con umbral de luminosidad.
 - Notificaciones, que el sistema emita sonidos en caso de error, cada cuanto tiempo actualizar la base de datos con estado del sistema.
- Ventana con los errores: en la parte inferior de la pagina aparece, en caso de haber algún error, un botón que permite desplegar la lista de errores del sistema.

PANTALLA MI HIDROPÓNICO

Pantalla principal en la que el usuario podrá modificar su cultivo añadiendo o cambiando las plantas.



Figura 30. Ventana Mi Hidropónico

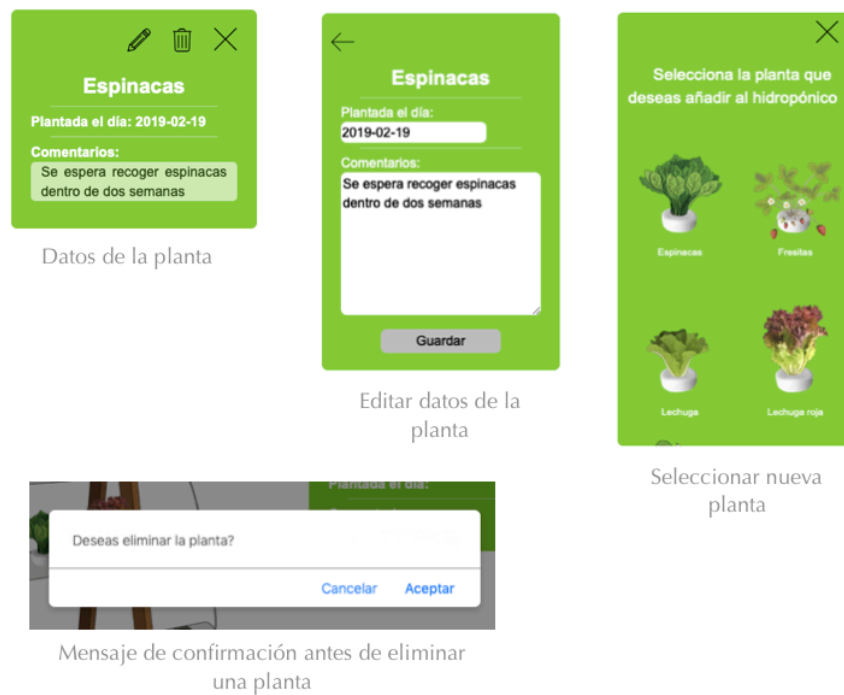


Figura 31. Formularios de la ventana Mi Hidropónico

El usuario tendrá que seleccionar la posición del hidropónico sobre la que quiera, o bien añadir una nueva planta, o en caso de ya haber una, modificarla o eliminarla. El procedimiento para llevar a cabo dichas acciones se explica en el siguiente diagrama de flujo.

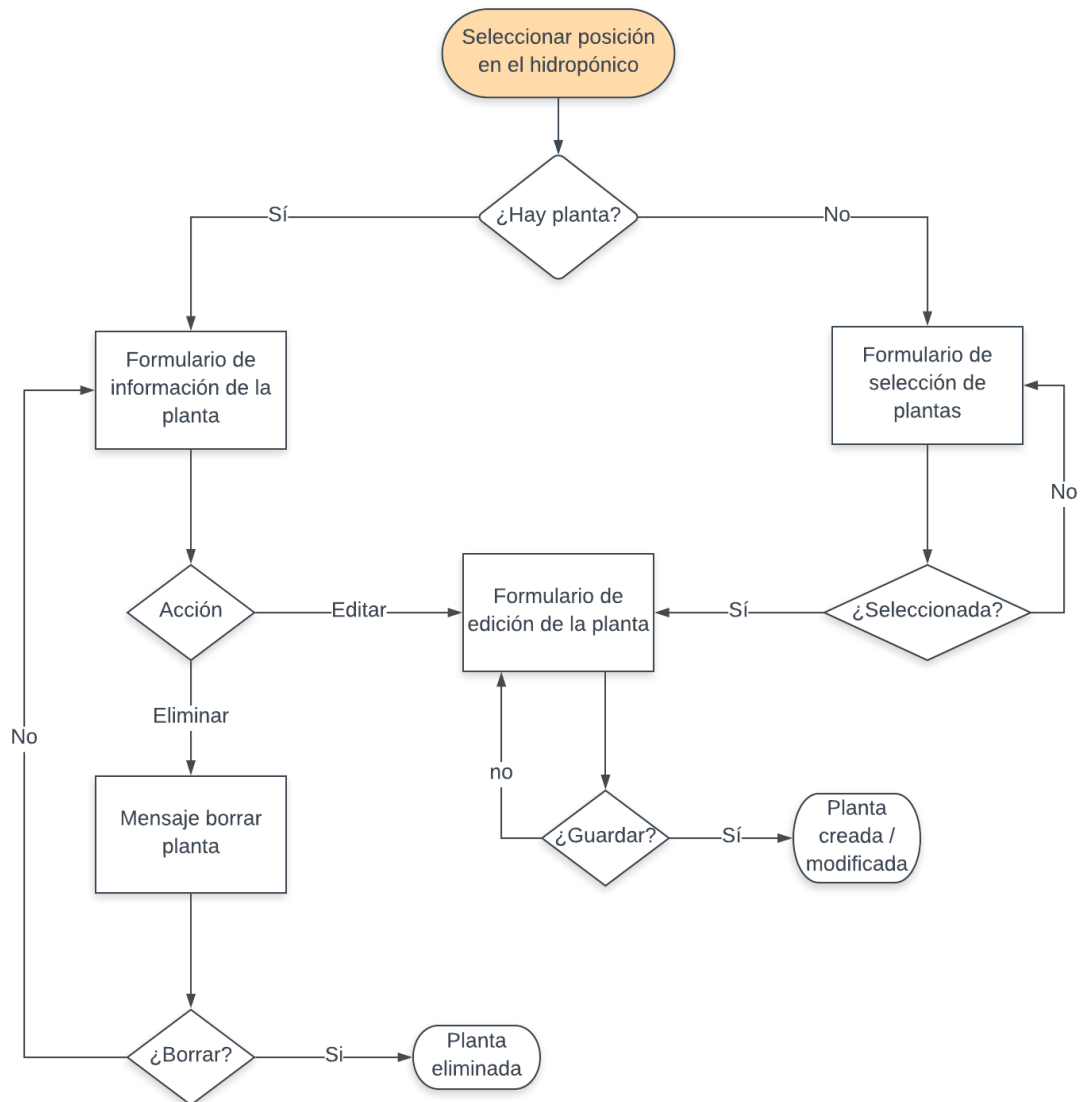


Figura 32. Diagrama de flujo de la ventana Mi Hidropónico

PANTALLA NIVELES Y NOTIFICACIONES

Mostrará de manera muy simplificada los valores de todos los sensores del sistema así como el estado de las notificaciones.

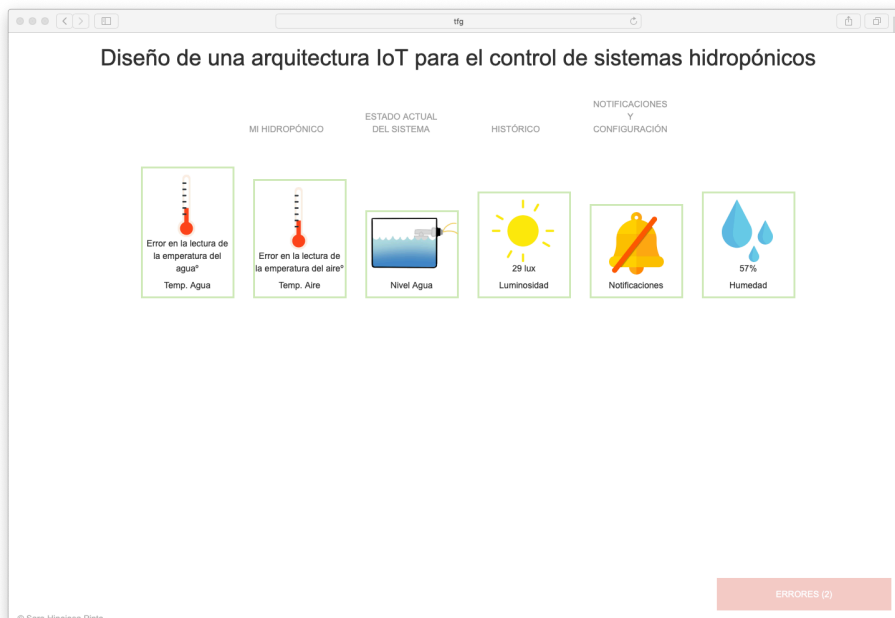


Figura 33. Ventana Niveles y notificaciones

PANTALLA HISTÓRICO

Mostrará una gráfica con los datos recogidos por el Arduino, por defecto se muestran los datos de la temperatura del agua, aire y luminosidad de las últimas 24 horas. Pero el usuario podrá seleccionar qué datos desea que se muestren en la gráfica y el periodo de tiempo.

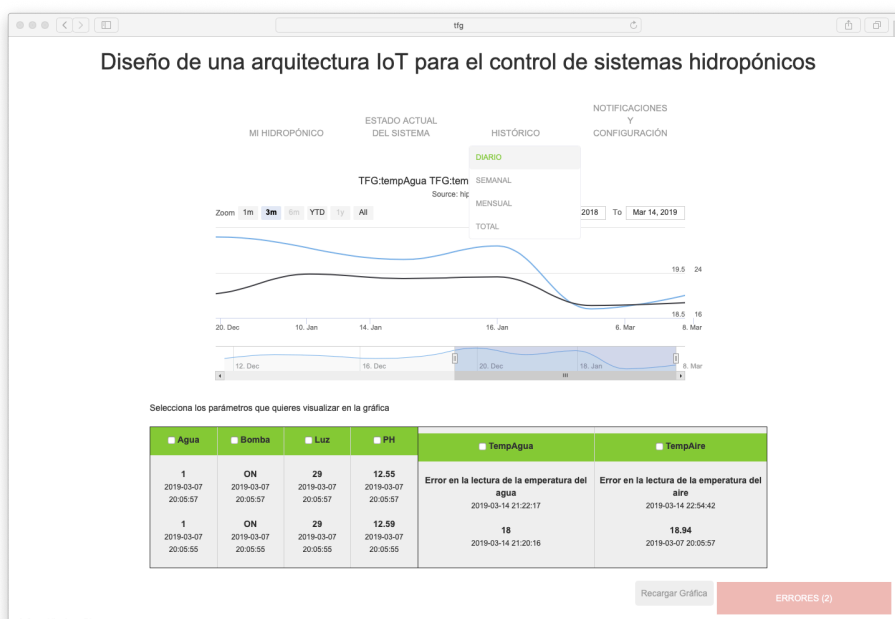


Figura 34. Ventana Histórico

Se ha utilizado la librería HighCharts para mostrar las gráficas.

PANTALLA CONFIGURACIÓN

Permite modificar los parámetros configurables del sistema. Notificaciones, cada cuanto tiempo pedir datos al Arduino, si debe emitir sonidos en caso de que haya algún error, cuando encender y apagar los LEDs, etc.



Figura 35. Ventana configuración

4. Implementación práctica

4.1. Montaje del circuito

Para realizar las primeras pruebas se utilizó una placa board, y una vez decidida la posición de todos los sensores y a qué pines del Arduino iban a estar conectados se soldaron en una placa, de manera que sea más difícil que se desconecte cualquier cable. Todo ello se introdujo en una caja que va colocada en la balda que tiene la estructura, sobre el depósito de agua.

4.2. Montaje de la estructura

Los materiales utilizados son de bajo coste y en su gran mayoría reutilizados. La idea es que se trate de un diseño que no requiera mucha inversión económica por parte del usuario, además se trata de una estructura muy sencilla de montar, compuesta básicamente por dos partes, la estructura vertical (hecha con madera) y el sistema de riego, donde se incluye los tubos de PVC y el depósito de agua en la parte inferior.

5. Conclusiones

5.1. Conclusiones personales

Gracias a este proyecto he aprendido cómo se crea un sistema de monitorización desde cero, así como la presentación de los datos recogidos por los sensores en una aplicación web muy básica. Cosa que me ha resultado muy interesante, ya que he podido aplicar muchos de los conocimientos adquiridos durante el grado en un único proyecto para hacer algo real que cualquiera podría tener en su casa y utilizar.

Por otra parte, considero que podría tratarse de un producto atractivo para aquellas personas interesadas en el tema del cultivo para el autoconsumo en sus propias casas y que además les gusten las nuevas tecnologías, siendo un producto asequible para bastantes familias y que no supone un coste elevado de mantenimiento.

5.2. Posibles aplicaciones y ampliaciones del sistema

El proyecto se ha planteado para el autoconsumo, pero, como ya se ha comentado previamente, la idea es fácilmente escalable, ya que la base de datos permite gestionar información de grandes empresas que puedan llegar a tener uno o varios sistemas hidropónicos, permitiendo unificar toda la información recogida por los sensores o bien, visualizarla por separado.

El hecho de que una gran empresa del sector de la agricultura disponga de tal cantidad de datos recogida automáticamente por los sensores les permite igualar y mejorar las condiciones a las que estuvo sometido el sistema durante una temporada buena.

Por otra parte, los sistemas hidropónicos se están instalando más asiduamente, ya que de esta manera no se desperdicia tanta agua ni tantos recursos, de la misma manera que se pueden cultivar la misma cantidad de productos en espacios más reducidos, permitiendo mejorar y aumentar la producción.

A fin de automatizar el sistema, podrían instalarse electroválvulas que cuando detecten niveles por debajo de los deseados, de agua o partículas en el agua (abono) se abran para añadir a la solución los nutrientes que necesitan las plantas o aumentar el nivel del agua del depósito.

6. Presupuesto

Este proyecto requiere, en primer lugar, tiempo para desarrollar, tanto las comunicaciones entre los dispositivos, como la recogida y almacenamiento de los datos y el desarrollo de una página web donde mostrarlos, es decir el software. También es importante dedicar tiempo al testeo de la página, para detectar posibles errores y corregirlos. Se desglosa a continuación una estimación de los costes.

Costes de las horas de trabajo para el desarrollo del software				
PERSONAL	COSTE / HORA	HORAS	TAREA	TOTAL
Desarrollador back-end	20,00 €	10	Configuración de la Orange Pi, modo servidor	200,00 €
Desarrollador back-end	20,00 €	20	Creación de los servicios de la página web	400,00 €
Desarrollador back-end	20,00 €	5	Diseño y creación de la base de datos	100,00 €
Desarrollador front-end	20,00 €	40	Diseño y desarrollo de la página web	800,00 €
Desarrollador Arduino	20,00 €	40	Comunicaciones entre los dispositivos, recogida de los datos de los sensores	800,00 €
Desarrollador	15 €	5	Comprobar el buen funcionamiento de la página	75 €
Total sin IVA				2375,00 €
IVA (21%)				498,75 €
TOTAL				2873,75 €

Con la estructura física ocurre algo similar, ya que lo más costoso sería el diseño y decisión de qué materiales utilizar, para garantizar una buena calidad y durabilidad del sistema.

Costes de las horas de trabajo para el desarrollo de la estructura física				
PERSONAL	COSTE / HORA	HORAS	TAREA	TOTAL
Diseñador industrial	20,00 €	40	Diseño de la estructura y decisión de los materiales a utilizar	800,00 €
Diseñador industrial	20,00 €	30	Creación de los planos	600,00 €
Total sin IVA				1400,00 €
IVA (21%)				294,00 €

TOTAL	1694,00 €
--------------	-----------

Una vez tenemos esta primera parte desarrollada únicamente se tendría que invertir en costes de soporte técnico, por si hubiese algún problema con el software, además de los materiales de la estructura.

Costes de los materiales del sistema			
MATERIAL	PRECIO	CANTIDAD	TOTAL
MATERIALES PARA LA ESTRUCTURA FÍSICA			
Traviesas madera (2x5,5x50cm)	1,60 €	5	8,00 €
Tubos PVC (5cm diam)	1,90 €	5	9,50 €
Mangueritas (3m)	1,00 €	1	1,00 €
Depósito agua	3,00 €	1	3,00 €
DISPOSITIVOS ELECTRÓNICOS			
Orange Pi	13,00 €	1	13,00 €
Arduino (CH340)	1,70 €	1	1,70 €
Sensor Ph	5,90 €	1	5,90 €
Sensor TDS	4,30 €	1	4,30 €
Sensores temperatura	0,70 €	2	1,40 €
Sensor luminosidad	0,40 €	1	0,40 €
Sensor humedad	0,40 €	1	0,40 €
Sensor nivel del agua	0,30 €	1	0,30 €
Bomba de agua	4,00 €	1	4,00 €
LEDs	3,00 €	1	3,00 €
TOTAL			55,90 €

Dado que en este proyecto ya se ha realizado la primera inversión que básicamente requería tiempo, y apenas costes económicos, puede llevarse a cabo el montaje de tantos sistemas como se desee a partir de los planos diseñados. Por lo que los siguientes sistemas requieren tiempo para su montaje y la configuración de todos los sensores para dejarlos preparados para la recogida de los datos.

7. Bibliografía

Estado del arte

- [1] El Mundo, Agricultores de altura y sin suelo, Agosto 2015, www.elmundo.es/economia/2015/08/16/55c37ceb268e3e184b8b4594.html
- [2] Agua para la producción sostenible de alimentos, <http://www.fao.org/3/y4525s/y4525s05.htm>
- [3] Libro de hidroponía y nutrición, J Smith Meza, https://www.academia.edu/4511736/Libro_de_hidroponia_y_nutricion
- [4] Hidroponia, solidario viaje al futuro, <http://viajes.elpais.com.uy/2017/05/10/hidroponia-solidario-viaje-al-futuro/>
- [5] Web informativa sobre hidroponía, <http://www.cultivarte.com.mx/faq>
- [6] Web informativa sobre hidroponía en nuestras casas, <https://www.hidroponiacasera.net/hidroponia-pdf/>
- [7] La técnica de la solución nutritiva recirculante (NFT), <http://dspace.utalca.cl/bitstream/1950/2927/1/NFT.pdf>
- [9] IKEA y la hidroponía, <https://www.ikea.com/es/es/cat/semillas-germinacion-cultivo-interior-pp002/>
- [10] Web empresa dedicada a la hidroponía, www.groho.es/
- [11] Web empresa dedicada a la hidroponía, www.tecniccat.com/hidroponia/
- [12] ABC, El proveedor de ensaladas de Mercadona diseña un sistema para cultivar sin tierra, Diciembre 2017, https://www.abc.es/espana/comunidad-valenciana/abci-proveedor-ensaladas-mercadona-disena-sistema-para-cultivar-sin-tierra-201712121919_noticia.html
- [13] IoT, la tecnología que puede ayudar a alimentar al planeta, <https://hablemos-deempresas.com/grandes-empresas/iot-en-invernaderos/>

Tecnología y software

- [14] Página web oficial, <http://www.orangepi.org>
- [15] Página web oficial, <https://www.raspberrypi.org>
- [16] Comparación Orange Pi con Rapsberry Pi, <https://www.hackerboards.com/compare/103,107,228/>
- [17] Página web oficial, <https://www.arduino.cc>
- [18] Información del Arduino, <https://aprendiendoarduino.wordpress.com/2016/06/27/arduino-uno-a-fondo-mapa-de-pines-2/>
- [19] Sensor de temperatura, <https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduino/>
- [20] Sensor de temperatura, <https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806>



- [21] Librería 1-WIRE, <http://web.fdi.ucm.es/posgrado/conferencias/MariaEmiliaCambronero2017-slides.pdf>
- [22] Sensor de humedad, <https://programarfacil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>
- [23] Sensor de luminosidad, <https://polaridad.es/bh1750-luz-sensor-iluminacion-ambiental-i2c-medida-luminosidad-medicion/>
- [24] Sensor de luminosidad, https://naylampmechatronics.com/blog/44_Tutorial-módulo-sensor-de-luz-BH1750.html
- [25] Sensor de Ph, <https://scidle.com/es/como-usar-un-sensor-de-ph-con-arduino/>
- [26] Sensor de Ph, <https://electronilab.co/tienda/sensor-analogico-de-ph-de-0-14/>
- [27] Buzzer, <https://hetpro-store.com/TUTORIALES/buzzer-con-arduino-generador-sonidos/>
- [28] Nivel del agua, <https://www.abcelectronica.net/productos/sensores/flujo-agua/>
- [29] Librería Dallas temperature, https://www.milesburton.com/Dallas_Temperature_Control_Library#The_Library
- [30] Sensor DHT11, <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/dht11-con-arduino/>
- [31] Librería BH1750, <https://github.com/PeterEmbedded/BH1750FVI>
- [32] Configuración del wifi Orange Pi, <http://www.3engine.net/wp/2017/12/puesta-en-marcha-y-configuracion-wifi-de-orange-pi-zero/>
- [33] Configuración Orange Pi, <https://www.freedesktop.org/software/systemd/man/systemd.unit.html>

Presupuesto

- [34] <https://www.fitorforestal.com/traviesas-de-tren.html>
- [35] <https://www.baezaonline.com/categorias/riego-agricola/mangueras-planas-flexibles-y-accesorios/manguera-plana-para-riego-.html>

Anexo 1

- [36] Herramienta Nmap, <https://openwebinars.net/blog/nmap-uso-basico-para-rastreo-de-puertos/>

Anexo 2

- [37] Protocolo MQTT, <https://ricveal.com/blog/primeros-pasos-mqtt/>

Anexo 5

- [37] Atom, www.atom.io
- [38] MySQL Workbench, <https://www.mysql.com/products/workbench/>
- [39] Arduino IDE, <https://www.arduino.cc/en/Main/Software>
- [40] Photoshop, <https://www.adobe.com/es/products/photoshop.html>



- [41] KiCad, <http://kicad-pcb.org>
- [42] Fritzing, <http://fritzing.org/home/>
- [43] LucidChart, <https://www.lucidchart.com/pages/>

Anexo 1. Herramienta nmap para escaneo de dispositivos en la red wifi

Se trata de una herramienta de código abierto para la exploración de redes. Fue diseñada para analizar rápidamente grandes redes, pero funciona muy bien con equipos individuales. Utiliza paquetes IP “crudos” para determinar qué equipos están disponibles en una red, qué servicios ofrecen, su sistema operativo, entre muchas otras características [36].

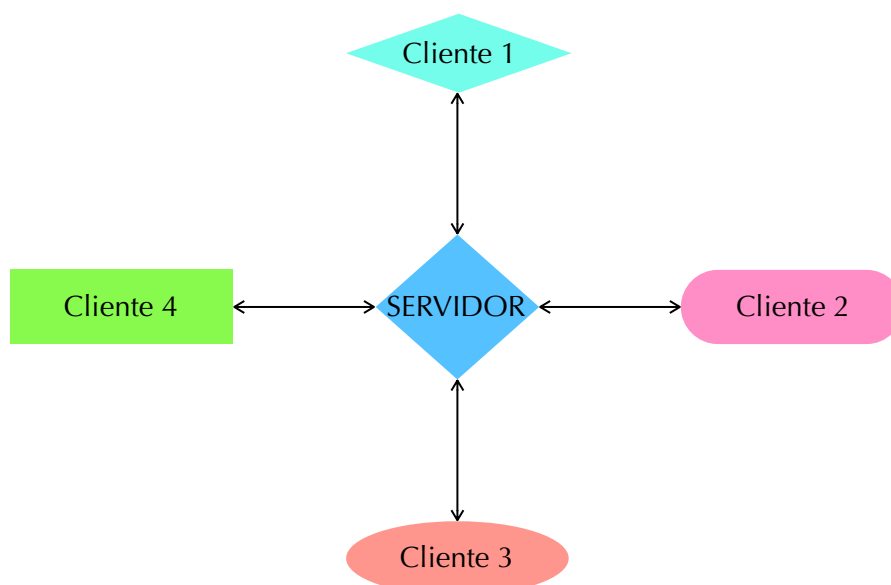
Se trata de uno de los escáneres de red más utilizados en la actualidad, por su eficacia y sencillez.

Para más información de la herramienta, se ha buscado en el propio manual `man nmap`.

Anexo 2. Protocolo MQTT (Message Queue Telemetry Transport)

Protocolo utilizado para la comunicación máquina-a-máquina en el “Internet de las cosas”. Su principal característica es que consume muy poco ancho de banda y puede utilizarse en la mayoría de sistemas empujados con pocos recursos como es el caso de la OrangePi utilizada en el proyecto. Por lo tanto, es un protocolo que está orientado a la comunicación de sensores [37].

MQTT sigue una topología de estrella, es decir, un nodo central hace de servidor y se encarga de gestionar la red y transmitir los mensajes a fin de mantener activo el canal. Los clientes mandan periódicamente un paquete (PINGREQ) y esperan la respuesta del servidor (PINGRESP). Tiene muchas configuraciones, y entre una de las opciones podemos configurar una comunicación cifrada. En nuestro caso, solo tendremos la OrangePi conectada al servidor, donde se almacenarán todos los datos del sistema.



La comunicación está basada en unos temas creados por el cliente donde publica el mensaje, si cualquier otro cliente deseara recibir dichos mensajes, tendrá que suscribirse al tema, por lo que la conversación puede ser de uno a uno o de uno a muchos. Además, los temas están organizados de manera jerárquica, por lo

que un nodo puede estar suscrito a un único tema o a varios.

► Configuración del Protocolo:

- I. Como servidor de MQTT se utilizará Mosquitto (etc/mosquitto.conf).
- II. Editamos el fichero de configuración, para que solo pueda accederse con usuario y contraseña, ya que por defecto no los pide. Se ha creado un dichero .txt que contiene los usuarios y contraseñas que queramos.

```
allow anonymous false
password_file /etc/mosquitto/usuarios
```

III. Probamos el correcto funcionamiento de la configuración.

1. En un terminal escribimos

```
mosquitto_sub -t tópico -u USUARIO -p PASSWORD
```

2. En otro terminal del mismo servidor tecleamos

```
mosquitto_pub -m MENSAJE -t tópico -u admin -P admtfg
```

3. Observamos que el mensaje tecleado en el 2º terminal aparece en el primero
4. Para probar la funcionalidad a través de la red, enviamos un mensaje de prueba a través de otra máquina tecleando el comando

```
mosquitto_pub -m MENSAJE EXTERNO -t tópico -u admin -P admtfg -h tfg.sara.hi-  
pis.es
```

5. Observamos también que el mensaje aparece en el 1º terminal.

► Configuración del demonio que recibe y almacena los datos que envía la OrangePi

- I. En el etc/systemd/system creamos una entrada dataMQTT.service con el siguiente contenido.

```
[Unit]
    Description=TFG - Datos MQTT
    After=mysql.service

[Service]
    ExecStart=/www/tfg/daemon/dataMQTT.php
    Restart=always

[Install]
    WantedBy=multi-user.target
```

De este modo configuramos un servicio que se lance después del MySQL y que se reinicia si ocurre cualquier error. Para habilitar el servicio teclearemos

```
systemctl enable dataMQTT.service
```

- II. Los datos que recibe el demonio, deben ajustarse al siguiente formato

1. El topico debe ser tfg, dado que el proyecto se ha creado con este nombre, si se conectase un segundo sistema hidropónico el tópico podría ser tfg2. Y de este modo tendríamos separados los datos de los sistemas.

2. El mensaje tendrá el formato:

```
tipo_dato1: valor1 {, tipo_dato2: valor2}
```

III. Para probar la funcionalidad definida, desde una máquina externa te-
cleamos

```
mosquitto_pub -m 'tAgua: 20' -t tfg -u admin -P admtfg -h tfg.sara.hipis.es
```

IV. Comprobamos que efectivamente, en nuestro servidor se ha añadido en la base de datos, en la tabla 'Datos' un registro con los datos recibidos y la fecha.

Con este protocolo, comunicaremos la Orange Pi con nuestro servidor, de manera que todos los datos medidos con los sensores, serán almacenados en nuestra base de datos.

Anexo 3. Código del Daemon en la Orange Pi para la comunicación con el Arduino.

Se encuentra en la carpeta comprimida anexos_SaraHinojosaPinto.zip (SerialIno.php).



Anexo 4. Código del Arduino.

Se encuentra en la carpeta comprimida anexos_SaraHinojosaPinto.zip (tfgSW.ino).

Anexo 5. Código de la página web.

Todo el código de la pagina web se encuentra en la carpeta comprimida anexos_SaraHinojosaPinto.zip (carpeta web).

Anexo 6. Programas y entornos de programación utilizados

Atom [37]		Desarrollo del código, en general, .html, .php.
Terminal		Las modificaciones de los ficheros de configuración de la Orange Pi se han hecho directamente desde la ventana de comandos, conectando con ésta por ssh.

MySQL Workbench [38]		La creación de la base de datos, consultas y modificaciones de la misma.
Arduino IDE [39]		Código del Arduino.
Photoshop [40]		Creación de los iconos de la página web.
KiCad [41]		Diseño de las placas electrónicas.
Fritzing [42]		Esquemas del montaje del circuito
LucidChart [43]		Creación de los diagramas de flujo y diagramas de bloques.