

**Universidad Internacional de La Rioja (UNIR)**

**ESIT**

**Máster Universitario en Industria 4.0**

# IoT aplicado al ámbito de la agricultura inteligente en el hogar.

**Trabajo Fin de Máster**

**presentado por:** Peón Lamparero, César

**Director/a:** Saez Manzano, Aurora

## Resumen

Desarrollo de un huerto inteligente conectado a una plataforma IoT para la gestión del cultivo. Se realiza la fabricación casera de un huerto inteligente a bajo coste con una propuesta de desarrollo para que un particular lo pueda llevar a cabo sin necesidad de tener conocimientos técnicos. Se han utilizado gran parte de los habilitadores digitales de la Industria 4.0 y se han conectado entre ellos. Se ha seguido una metodología de trabajo de prueba de concepto (POC) y SCRUM.

**Palabras clave:** Huerto inteligente, Internet de las Cosas, Industria 4.0, Agricultura inteligente

## Abstract

Development of a smart garden connected to an IoT platform for crop management. The homemade manufacture of a smart garden is carried out at low cost with a development proposal so that an individual can carry it out without the need for technical knowledge. Much of the digital enablers of Industry 4.0 have been used and connected to each other. A proof of concept (POC) and SCRUM work methodology has been followed in this project.

**Keywords:** Smart garden, Internet of Things (IoT), Industry 4.0, Smart agriculture

## Índice de contenidos

1.	Introducción .....	10
1.1.	Motivación .....	10
1.2.	Planteamiento del trabajo.....	11
1.3.	Estructura de capítulos.....	11
2.	Contexto y estado del arte .....	13
2.1.	Descripción general del contexto del proyecto.....	13
2.2.	Proyectos relacionados con el tema del TFM .....	13
2.2.1.	Productos en venta.....	13
2.2.2.	Proyectos con microcontroladores .....	16
2.3.	Tecnologías relacionadas con el tema del TFM.....	20
2.3.1.	Sensores y microcontroladores .....	20
2.3.2.	Impresión 3D .....	21
2.3.3.	Internet of Things.....	21
2.3.4.	Ciberseguridad .....	22
2.3.5.	Big data.....	23
2.4.	Conclusiones sobre el estado del arte .....	24
3.	Descripción general de la contribución del TFM.....	28
3.1.	Objetivos.....	28
3.2.	Metodología del trabajo .....	28
3.2.1.	POC.....	28
3.2.2.	SCRUM.....	30
3.3.	Descripción general de las partes o componentes de la propuesta.....	32
4.	Desarrollo específico de la contribución .....	36
4.1.1.	SPRINT 1: Investigación.....	36
4.1.2.	SPRINT 2: Desarrollo I .....	38
5.1.2.	SPRINT 3: Desarrollo II .....	58

5.1.3.	SPRINT 4: Pruebas .....	84
6.	Conclusiones y trabajos futuros .....	85
6.1.1.	“Líneas de trabajo futuras” .....	86
	Referencias bibliográficas .....	88
Anexo A.	Código implementado .....	91
Anexo B.	Links de los materiales .....	120

## Índice de figuras

Figura 1: Click and Grow huerto inteligente .....	14
Figura 2: Huerto Biohort.....	14
Figura 3: Blumfeldt.....	15
Figura 4: Click and Grow Huerto vertical .....	15
Figura 5: Prototipo final TFG Ata Jaber, 2017 .....	17
Figura 6: Resultado final Carlos Alexander.....	19
Figura 7: Integración de las tecnologías . .....	33
Figura 8: Arduino.....	38
Figura 9: Accesorios Arduino I .....	39
Figura 10: Accesorios Arduino II .....	39
Figura 11: Sensor flujo de agua .....	40
Figura 12: Pantalla RGB .....	40
Figura 13: Sensor de humedad de tierra, bomba de agua y relé.....	41
Figura 14: Placa solar Arduino .....	42
Figura 15: Sensor fotorresistencia .....	42
Figura 16: Sensor dht11 de humedad y temperatura .....	43
Figura 17: Módulo wifi ESP8266-ESP01 .....	44
Figura 18: Placa ESP8266 .....	45
Figura 19: Configuración DHT11 a Arduino .....	46
Figura 20: Instalar bibliotecas Arduino .....	47
Figura 21: Librería DHT11 Arduino .....	47
Figura 22: Resultado DHT11 .....	47
Figura 23: Sensor LDR a Arduino .....	48
Figura 24: Datos del sensor de luminosidad.....	49
Figura 25: Conexión módulo WIFI ESP8266-ESP01 .....	49
Figura 26: Configuración NodeMCU ESP8266.....	51

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Figura 27: Conexión correcta al WIFI .....	51
Figura 28: Comunicación serial entre Arduino y ESP8266 .....	52
Figura 29: Blynk en el navegador .....	54
Figura 30: Blynk en un móvil Android .....	54
Figura 31: Comunicación bidireccional Arduino-NodeMCU ESP8266.....	55
Figura 32: Arduino con led azul .....	56
Figura 33: Pila 9V y adaptador.....	57
Figura 34: Esquema final Arduino - NodeMCU ESP8266.....	57
Figura 35: Resultado final Prueba de Concepto .....	58
Figura 36: Tablas de madera de pino .....	59
Figura 37: Tablas cortadas .....	60
Figura 38: Estructura del huerto.....	60
Figura 39: Estructura principal .....	61
Figura 40: Modelo 3D tanque de agua.....	62
Figura 41: Opción de impresión 3D en cura .....	62
Figura 42: Construcción del tanque de agua .....	63
Figura 43: Resultado final del tanque de agua .....	63
Figura 44: Tanque de agua y sensor de flujo de agua.....	64
Figura 45: Tanque de agua y tuberías conectadas al huerto.....	64
Figura 46: Huerto con tierra y techo.....	65
Figura 47: Almacén en el huerto .....	66
Figura 48: Electrónica final huerto .....	67
Figura 49: Relé y tanque de agua.....	67
Figura 50: Resultado final .....	68
Figura 51: Datos Blynk final .....	68
Figura 52: Estructura sensores básicos.....	70
Figura 53: Estructura sensores utilizados .....	71

Figura 54: Conexión Arduino a NodeMCU8266.....	72
Figura 55: Seleccionar placa Arduino .....	72
Figura 56: Arduino y sensor de humedad de tierra .....	73
Figura 57: Botones en IDE Arduino .....	74
Figura 58: Nuevo template en Blynk .....	75
Figura 59: Código para conectar NodeMCU ESP8266 con Blynk .....	75
Figura 60: Placa NodeMCU ESP8266 en herramientas .....	76
Figura 61: Configurar labels en Blynk .....	77
Figura 62: Widget en Blynk .....	78
Figura 63: Dashboard final Blynk .....	78
Figura 64: Token de autorización de Blynk.....	79
Figura 65: Esquema final .....	79
Figura 66: Cortes de las tablas de madera .....	80
Figura 67: Estructura final del huerto .....	81
Figura 68: Sistema eléctrico conectado .....	82

## Índice de tablas

Tabla 1: Elementos necesarios huerto.....	69
---	----

# 1. Introducción

## 1.1. Motivación

El problema principal del estudio es el desarrollo de un huerto inteligente en el hogar, desde el cuál el usuario que desee construirlo, pueda hacerlo a bajo coste. Gracias a la ayuda de las nuevas tecnologías presentes en la industria 4.0, se puede llevar el cultivo de casi cualquier hortaliza del campo a la casa, sin necesidad de grandes cantidades de espacio. Incluso, si se quisiese, se podría realizar un cultivo vertical, con lo cual los metros cuadrados utilizados para dicho cultivo, podrían generar mayor cantidad de hortalizas, al poner un huerto encima de otro.

Actualmente es muy complicado tener una parcela de tierra para poder crear un huerto ecológico propio, o tener suficiente espacio en la casa donde poder poner varias macetas para poder plantar lo que se necesite. Y aunque se tuviesen estas condiciones, sería complicado cultivar hortalizas fuera de su temporada habitual, ya que el clima no lo favorecería. Por esto, la mayoría de las compras de verduras, frutas y hortalizas, se realizan en supermercados, donde estos alimentos vienen en plásticos y han perdido gran parte de su sabor original. Este proyecto plantea una solución alternativa al cultivo original y pretende aumentar el consumo de alimentos recién obtenidos de la tierra, así como la posibilidad de poder cultivar cualquier tipo de hortaliza en cualquier momento del año.

Otro aspecto importante es el consumo de las hortalizas. Cada vez más, el número de personas es mayor, por lo que crece el consumo y se necesita abastecer a todos. También, se desperdicia gran cantidad de comida porque no se vende toda la que se produce, y como las hortalizas se estropean rápidamente, se acaban tirando. Si cada persona pudiese cultivar la comida que se necesite, no habría problemas de abastecimiento ni de desperdicio de comida. Este proyecto no pretende arreglar este problema, pero tener la posibilidad de llevar un huerto a cualquier casa, podría ser una parte de una solución que lucha por la sostenibilidad.

La tecnología avanza rápidamente, y las ciudades futuristas que se veían en el cine antiguamente, ya no quedan tan lejos. Antes parecía imposible poder teletransportarse de un lugar a otro, y ahora gracias a la telefonía móvil y el internet, se puede hacer una video llamada desde cualquier sitio en tiempo real. Los coches son eléctricos y empiezan a ser autónomos, y los objetos de las casas empiezan a comunicarse entre ellos. Se tiene el control completo de los vehículos y de las casas desde el mismo teléfono. De hecho, ya no es necesario tener que salir a la calle para poder trabajar. Sin duda alguna, la alimentación es otro sector que

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

también avanza con las nuevas tecnologías y este proyecto es un claro ejemplo de esto. Antiguamente, se necesitaba de una parcela de tierra donde poder cultivar las hortalizas y tener que ir a regarlas de manera diaria. Cuidar de que otros animales no intentasen comérselas y proteger a las plantas de las variaciones del clima y de plagas. Hoy en día, es posible tener un huerto dentro de una casa, con mayor producción por metro cuadrado gracias a la posibilidad de realizar un cultivo en vertical, con un riego automatizado y sin preocuparse de los contratiempos del clima.

## 1.2. Planteamiento del trabajo

En este proyecto se van a explicar paso a paso como fabricar un huerto inteligente desde cero, para que cualquier persona pueda hacerlo. El tamaño del huerto inteligente depende de la hortaliza que se quiera cultivar, por esto es importante fabricarlo todo de manera manual y elegir el tamaño según la hortaliza. En primer lugar, se creará un macetero normal con las medidas necesarias para la hortaliza que se quiera cultivar. En segundo lugar, se montará el sistema electrónico con la robótica necesaria y después se conectará mediante un sistema IoT. Por último, una vez el huerto este construido, se realizará una primera prueba para comprobar que todo funciona correctamente, se llenará la maceta con abono y se plantarán las semillas.

A lo largo del proyecto, se explicarán los problemas encontrados y como solucionarlos.

Además de la construcción de la maceta y el sistema electrónico para poder crear el huerto electrónico, se pretende utilizar un sistema IoT en el que, en función de los valores obtenidos por los diferentes sensores, se regule automáticamente el riego del huerto. También, se quiere almacenar en una base de datos compartida en la nube, los valores que necesitarían las diferentes hortalizas para tener un consumo óptimo. De esta forma, cada vez que un usuario quiera plantar una hortaliza en su huerto inteligente, solo tenga que consultar los parámetros necesarios para sus sensores, y no tener que preocuparse de nada durante el proceso de crecimiento. Solo de añadir agua al deposito cuando sea necesario. En caso de que se tenga una impresora 3D, se podrá utilizar para algún elemento prescindible del huerto inteligente.

## 1.3. Estructura de capítulos

A continuación, se van a enumerar y explicar de forma resumida, la estructura de los capítulos en este trabajo.

### 1. Contexto y estado del arte

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

En este primer apartado se explica el contexto en el que se encuentra este trabajo. Se va a realizar una búsqueda de proyectos de agricultura inteligente y de otros huertos inteligentes que sirvan de apoyo para el desarrollo del TFE. Además, se van a analizar las diferentes tecnologías habilitadoras de la industria 4.0 que forma parte de este proyecto.

## 2. Descripción general de la contribución del TFM

En el segundo apartado se van a establecer los objetivos del proyecto y las diferentes metodologías que se van a utilizar para el desarrollo del trabajo.

## 3. Desarrollo específico de la contribución

Según se hayan establecido las metodologías de trabajo del segundo apartado, se realizará la estructura del tercer apartado, el cual es el grueso principal del trabajo. En este apartado se realiza el desarrollo del huerto inteligente, desde la adquisición de los diferentes materiales, construcción, montaje y pruebas finales.

## 4. Conclusiones y trabajos futuros

Por último, se explicarán las conclusiones obtenidas del desarrollo del proyecto, así como diferentes líneas de trabajo donde poder continuar con el desarrollo de este proyecto.

## 2. Contexto y estado del arte

### 2.1. Descripción general del contexto del proyecto

El proyecto se encuentra en medio de un sector que está sufriendo un gran crecimiento tecnológico debido a la automatización de la industria. Cada vez más se pretende automatizar la mayoría de los procesos además de aportar mayor autonomía y personalización a todo. El huerto inteligente es un claro caso, en el que se podrá cultivar lo que se quiera, cuando se quiera y de forma autónoma. Para esto, se hace uso de las nuevas tecnologías como IoT, big data, inteligencia artificial, impresión 3D, ciberseguridad, y el uso de sensores y microcontroladores. El sector alimentario ya ha iniciado diversos proyectos, haciendo uso de las tecnologías referentes a la industria 4.0, para mejorar su producción. Se ha mejorado la calidad de los productos, la seguridad de cada proceso y el rendimiento de la tierra, tanto en tiempo como en espacio, y cada vez va a más. Las grandes empresas ya apuestan por grandes invernaderos completamente automatizados, cultivos hidropónicos y producciones verticales, pero esto apenas ha llegado al usuario particular. La mayoría de los avances que se han desarrollado a nivel particular ha sido mediante personas con altos conocimientos en estas tecnologías que han desarrollado proyectos propios y los han compartido con una comunidad mediante un foro o una red social. Esto hace que gran parte de los proyectos que estarán relacionados con el tema del TFM provengan de este tipo de sitios, en vez de “fuentes oficiales”.

### 2.2. Proyectos relacionados con el tema del TFM

Los proyectos relacionados con el tema del TFM se dividen en dos partes. La primera parte son los productos que ya están desarrollados, pero con un coste elevado, y la segunda parte, proyectos donde se utilizan sensores como los que se van a utilizar en este proyecto, así como proyectos teóricos parecidos a este proyecto.

#### 2.2.1. Productos en venta

De los productos en venta, hay que hacer una pequeña separación, entre los huertos inteligentes individuales (como el que se va a desarrollar en este proyecto), y los jardines verticales

##### 2.2.1.1. Huerto inteligente individual

La empresa click and grow tiene varios modelos de huertos inteligentes. Tienen un diseño minimalista y adaptado al hogar. Su coste es bastante elevado. (Click & Grow, 2021)

IoT aplicado al ámbito de la agricultura inteligente en el hogar.



**Figura 1: Click and Grow huerto inteligente. Fuente: (Click & Grow, 2021).**

La empresa biohort, tiene este modelo de huerto inteligente. Su diseño es muy grande y adaptado para el exterior. Su coste vuelve a ser muy elevado (Biohort GmbH, 2021).



**Figura 2: Huerto Biohort. Fuente: (Biohort GmbH, 2021).**

El último ejemplo es de la empresa blumfeldt. Este huerto inteligente no necesita tierra para el cultivo de la planta, ya que se trata de un cultivo hidropónico. Tanto la iluminación como el suministro de nutrientes en el agua se realiza de forma inteligente, por lo cual, pertenece al grupo de huertos inteligentes. Su diseño es bonito y adaptado al hogar. Tiene un coste menos elevado que los anteriores ejemplos (blumfeldt, 2021).



**Figura 3: Blumfeldt. Fuente: (blumfeldt, 2021).**

#### 2.2.1.2. Jardines verticales inteligentes

La empresa Click and Grow, mencionada también en el anterior apartado, además de los huertos inteligentes individuales, también tienen a la venta varios modelos de jardines verticales. Estos jardines son equivalentes a varios modelos de huertos inteligentes individuales, pero colocados uno sobre otro. Gracias a esto, se puede tener mayor producción en los mismos metros cuadrados (Click & Grow, 2021).



**Figura 4: Click and Grow Huerto vertical. Fuente: (Click & Grow, 2021).**

Gracias a los productos que se han expuesto anteriormente, se pueden sacar varias ideas de cara al desarrollo de este proyecto.

En primer lugar, ideas de diseño. Estos productos tienen diseños minimalistas y con colores blancos y madera, sin estar sobrecargados de luces y botones, y en ninguno se ve ningún cable por ninguna parte. En cuanto al tamaño, el primer y tercer proyecto tienen tamaños adaptables al hogar, en cambio el segundo solo se podría poner en el exterior. El tamaño de este proyecto se relaciona más con el primer y segundo producto.

En segundo lugar, ideas de escalabilidad. Una vez que se haya creado un huerto inteligente a bajo coste, y se compruebe su correcto funcionamiento, sería interesante poder construir más hasta tener los huertos necesarios para cubrir el abastecimiento personal de cada individuo. Se podría crear un jardín vertical para aprovechar al máximo el espacio donde se ubiquen estos huertos. Como la idea de este proyecto no es crear un jardín vertical, el diseño del huerto inteligente tiene que permitir colocar otro huerto inteligente justo encima. En el primer ejemplo, no se podría colocar un huerto encima de otro, ya que se caería el de arriba y/o se romperían los leds del de abajo. El tercer proyecto si que podría soportar otro huerto encima suyo.

En tercer lugar, el coste de estos productos. Este proyecto busca realizar productos similares a los que se han visto, pero de forma más económica, personalizada y manual. Para esto, el presupuesto para realizar este proyecto tiene que ser inferior al coste de estos productos. El más económico de estos productos, actualmente tiene un coste de 120€, por lo que el presupuesto para este proyecto debería de ser inferior a 100€ para que sea más rentable fabricarlo que comprarlo.

### **2.2.2. Proyectos con microcontroladores**

#### **2.2.2.1. Proyectos con Arduino**

En el siguiente repositorio de GitHub, se puede encontrar un proyecto realizado hace 7 años por gradyh (gradyh, 2015). En este proyecto se utiliza una placa de Arduino para conectar los sensores de humedad, humedad de tierra y una fotorresistencia para obtener datos del jardín. Además, ha configurado un riego automático al jardín en función de la temperatura de la tierra.

En el siguiente proyecto se conecta una placa de Arduino a una plataforma IoT mediante un módulo de wifi ESP8266 ESP-01 (elifhocarobotik, 2021). La configuración de internet mediante un modulo externo a una placa de Arduino es un elemento muy interesante, ya que IoT aplicado al ámbito de la agricultura inteligente en el hogar.

muchas personas han podido adquirir una placa de Arduino para realizar algún proyecto personal, y con la intención de este proyecto de generar una comunicación mediante una plataforma IoT, por culpa de la conexión a internet, algunas personas tendrían que adquirir una placa de Arduino adicional que de base soportase la configuración a internet mediante wifi, como por ejemplo la placa de Arduino UNO WIFI R2. Gracias a este modulo, cualquier placa de Arduino se puede conectar a internet, lo cual facilita la conexión a la plataforma IoT y abarataría los costes para las personas que ya dispusiesen de una placa de Arduino previamente.

En el proyecto de (Ata Jaber, 2017), se realiza un prototipo funcional de un huerto inteligente. Para llevar a cabo la comunicación con internet, se realiza gracias al modulo ESP8266-ESP01. El prototipo final tiene por separado el huerto, el tanque de agua y por otro lado la placa de Arduino junto con todos los sensores. Se puede ver en la siguiente fotografía.



**Figura 5: Prototipo final TFG Ata Jaber, 2017. Fuente: (Ata Jaber, 2017).**

En el documento de Acosta, Aguilar y Pinzón Trejos (2015), se ha realizado una conexión a internet mediante una placa de Arduino para realizar el envío de diferentes sensores. La conexión a internet se ha realizado mediante la placa Arduino YUN y una conexión a ethernet (conexión por cable) al router. De esta forma, se consigue enviar los datos a la aplicación móvil que ha sido desarrollada mediante Android Studio.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Este proyecto de la propia plataforma de Arduino muestra de la monitorización inteligente de un cultivo a gran escala (Cortina, 2019).

#### 2.2.2.2. Proyectos con Raspberry

El proyecto de Finio consiste en automatizar la función del riego en su jardín (Finio, s.f.). Este proyecto está realizado con una Raspberry y varios sensores de flujo de agua conectados entre sí, y gestionados desde la Raspberry. El creador de dicho sistema conecta la Raspberry al ordenador para encender o apagar los diferentes sensores de flujo de agua, para dejar pasar el agua o cortarla. Este proyecto es interesante por saber como se ha configurado un riego automático mediante una Raspberry, pero el proyecto va más enfocado a un gran jardín con varias zonas donde regar, que para el riego automático de un pequeño huerto.

#### 2.2.2.3. Proyectos con Arduino y Raspberry

En el siguiente proyecto, se utilizan tanto la placa de Arduino como la placa de Raspberry para el desarrollo de un jardín inteligente (hackster.io, 2017). La opción principal por lo cuál utiliza la Raspberry es para incorporar un módulo de cámara en la Raspberry mediante la cuál vigilar el estado de la planta con video, además de tener la aplicación conectada a una plataforma IoT y ver los datos en tiempo real. La placa de Arduino se utiliza para el resto de los sensores, tales como la humedad, la luz y la temperatura.

La fabricación de un huerto hidropónico también se puede realizar utilizando las placas de Arduino y Raspberry (Potter, 2015). Mediante este proyecto se muestra como se puede realizar el cultivo de una planta sin necesidad de usar tierra, solamente con agua y con los nutrientes necesarios para que la planta crezca. Para el desarrollo de un huerto hidropónico, además de que los elementos son más caros que los que se necesitan para el desarrollo de un huerto “tradicional”, los conceptos para entenderlo son más complicados, y la idea de este proyecto es que este al alcance de todos, sin que tenga conocimientos previos sobre los habilitadores tecnológicos de la industria 4.0, ni de jardinería.

#### 2.2.2.4. Proyectos sin Arduino ni Raspberry

En el proyecto TFG de Carlos Alexander (2021) se ha realizado el desarrollo de un huerto inteligente conectado a internet. La conexión a internet se ha realizado mediante una placa

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

de desarrollo NodeMCU ESP 32S a la cuál se le han conectado el resto de los sensores. Para el desarrollo de este prototipo no se ha utilizado ni Arduino ni Raspberry. La aplicación utilizada en este proyecto ha sido desarrollada por el autor mediante el desarrollo de una aplicación móvil con Flutter. El resultado final consiste en la monitorización de una persiana para la entrada de luz, y la monitorización del riego automático. Se puede ver en la siguiente fotografía.



**Figura 6: Resultado final Carlos Alexander. Fuente: (Icaza Guamba, 2021).**

#### 2.2.2.5. Proyectos sin implementación

En el proyecto de Raihan (2016) se muestra un diagrama de como poder realizar un huerto inteligente gracias a una placa Arduino. Este proyecto no se ha llevado a cabo, y solo se ha mostrado un diagrama de como los diferentes sensores y demás elementos deberían de funcionar. Frente al resto de proyectos que se han comentado anteriormente, este proyecto ofrece la posibilidad de conectar la placa de Arduino a un panel solar para poder generar la

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

energía suficiente para el riego automático y para la iluminación. Este es el proyecto más completo de todos los que se han comentado, y solo utilizando una placa de Arduino.

## 2.3. Tecnologías relacionadas con el tema del TFM

Este proyecto va a hacer uso de las tecnologías relacionadas con la industria 4.0, para ello se va a realizar una breve introducción a cada una de ellas, y así ponerlas en contexto.

### 2.3.1. Sensores y microcontroladores

Los principales protagonistas en este proyecto son los sensores y los microcontroladores que se van a utilizar, ya que son los encargados de obtener los datos necesarios para cultivar la hortaliza. Gracias a los sensores se van a obtener los datos necesarios para el correcto cultivo, y los microcontroladores serán los responsables de procesar esos datos. Actualmente hay una gran cantidad de sensores en el mercado, cada uno encargado de obtener un dato en concreto.

Por ejemplo, el sensor DH11 es un sensor encargado de recoger los datos de la temperatura y de la humedad. Este sensor va conectado a una placa de Arduino, la cual se encarga de procesar esos datos.

El proyecto de Arduino nació en el año 2005 donde el coste de una placa de Arduino era de 100\$ (Wikipedia, s.f.), actualmente se puede encontrar en Amazon a bajo coste junto a todo lo necesario para poder utilizarla, como una placa de prueba, 1 cable para poder conectarlo al pc, cables, resistencias y leds.

Otra placa similar a Arduino es la Raspberry pi, la cual es una placa base como la de un ordenador, pero con un tamaño muy reducido. La fuente de alimentación se genera mediante un cable usb-microusb, y el resto de las entradas son las mismas que en un ordenador, hdmi para una pantalla y varios puertos usbs. El almacenamiento, en lugar de ser un disco duro hdd o ssd, se utiliza una tarjeta microsd (como las que se suele utilizar en los teléfonos móviles). Esta placa requiere de un sistema operativo para poder funcionar, a diferencia de la placa de Arduino que no necesita de uno. La placa de Arduino a su vez, necesita estar conectado a un pc para poder programarlo, en cambio la placa de Raspberry se vale por si misma.

El coste de una Rapsberry es más elevada que la de un Arduino.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

### **2.3.2. Impresión 3D**

La impresión 3D es una tecnología que cada vez se está utilizando más gracias a que es viable poder tener una impresora 3D en casa, y las posibilidades que ofrece son infinitas. Las impresoras 3D más comunes son FDM (Modelado por deposición Fundida), o por SLA (estereolitografía). La primera consiste en calentar el plástico hasta fundirlo y dar forma a la pieza que se quiere fabricar capa a capa, es decir, imprimiendo una capa encima de otra. Por otro lado, las impresoras de SLA utilizan el principio de fotopolimeración para crear las piezas a partir de resinas sensibles a los haces de luz.

Para poder imprimir una pieza, primero se tiene que realizar un diseño 3D con un software que lo permita, como Autodesk Fusion 360. Una vez que se tiene el objeto 3D, se tiene que exportar el fichero a otro software que realice un modelado de capas para que la impresora pueda imprimirlo, como puede ser CURA Ultimaker. En este último software también se puede parametrizar lo que se requiera para la pieza, como puede ser la velocidad de impresión, la altura de cada capa, si se requiere soporte, y demás características.

Para realizar un diseño 3D no se necesita de un profesional, se puede aprender de forma autodidacta con cursos gratuitos, y crear los diseños con software también gratuito (como Blender o 3D Builder). Una vez teniendo el diseño 3D, se necesita una impresora 3D para poder imprimir lo que la imaginación permita.

Esto aumenta las posibilidades de poder desarrollar y fabricar cualquier cosa, en el caso de este proyecto, cualquier pieza que necesite nuestro huerto inteligente. De todas maneras, el coste de una impresora 3D es elevado, frente al presupuesto que requiere este proyecto, de tal modo, que las piezas que se fabriquen utilizando esta tecnología siempre serán de forma voluntaria y en ningún caso imprescindibles para el desarrollo principal del huerto electrónico.

### **2.3.3. Internet of Things**

IoT (Internet of Things), o el internet de las cosas, se basa en la comunicación de diferentes dispositivos gracias a internet. Cualquier objeto o dispositivo que se pueda conectar a internet, forma parte del ecosistema IoT. Hoy en día es muy común que el ordenador y el móvil sean dispositivos que están conectados a internet y a nadie le extraña, pero poco a poco son más los dispositivos que se empiezan a conectar a internet: relojes, luces, bombillas, altavoces, televisiones, etc. Hace unos años era imposible pensar que una bombilla o un altavoz estuviesen conectados a internet, pero ahora, gracias a esa conexión podemos encenderlos y apagarlos solo con nuestra voz, y la ayuda de un asistente virtual. Este asistente virtual escuchará nuestro mensaje y lo “traducirá” al lenguaje del dispositivo para que pueda IoT aplicado al ámbito de la agricultura inteligente en el hogar.

entender la orden. A medida que avanza el tiempo, el número de dispositivos que se pueden conectar a internet, y las funcionalidades nuevas que van saliendo, no paran de crecer. En los ejemplos anteriores de la bombilla y el altavoz, ya no solo se pueden encender y apagar, sino que existen un sinnúmero de funcionalidades. En la bombilla se puede elegir el color, o colores, la intensidad de la iluminación, programar unas horas para que este encendida y luego se apague, e incluso, que se adapte a la música que está sonando, o a la película que esta reproduciendo un televisor inteligente. En el caso del altavoz más de lo mismo, ahora se le puede pedir una canción, que pase a la siguiente o vuelva a la anterior, que suba o baje el volumen, que diga el tiempo que va a hacer hoy, o que te cuente un chiste. La evolución que tiene ahora cada dispositivo es increíble, y cada vez más gracias al internet de las cosas, y de como se “comunican” las máquinas entre sí. Por ejemplo, el altavoz puede decir que intensidad de luminosidad tiene la bombilla que esta encendida, y la bombilla adaptarse a la música que sale del altavoz.

De cara a este proyecto, es importante que el huerto se pueda comunicar mediante internet para comprobar que los valores que se obtienen de los sensores son los correctos, y en el caso de que no lo sean, actuar en consecuencia. Si la tierra tiene poca agua, regar de forma automática, o si el foco da mucho calor, apagarlo. Esto se puede hacer gracias al IoT ya que la humedad del agua y la intensidad de los focos variara según la planta que se este cultivando, y esos valores se pueden obtener de internet. Además, si el huerto tiene algún problema (agua, luz, temperatura, etc.), se puede mandar una señal para que avise al usuario mediante un correo electrónico o una notificación en una aplicación para móvil, esto también requiere de conexión a internet.

#### **2.3.4. Ciberseguridad**

La ciberseguridad se encarga de arreglar las posibles brechas de seguridad que haya en internet. Hoy un día, si una empresa recibe un ciberataque puede llevarle a la quiebra si ciertos datos sensibles no están bien protegidos y quedan expuestos. Si esto llegase a pasar, los cibercriminales que han realizado el ataque podrían pedir grandes cantidades de dinero a cambio de no hacer públicos los datos. Por esto principalmente, hay una gran cantidad de hackers buscando constantemente, brechas de seguridad en cada sistema. Los cibercriminales utilizarán esa brecha de seguridad para atacar, y los hackers de sombrero blanco o hackers éticos, utilizarán esa brecha de seguridad para arreglarla o para comunicarlo al afectado para que lo arregle.

Hoy en día hay muchos tipos de malware, o software malicioso, que se pueden “colar” en cualquier sitio, desde el pendrive personal de un empleado, el acceso wifi de un teléfono, o directamente desde un correo electrónico. En realidad, la única forma de estar seguro sería no teniendo conexión a internet, y aún así, no sería del todo seguro. Internet no fue construida pensando en la seguridad, y por ello, hay el ámbito de la ciberseguridad es tan importante.

En el caso del huerto inteligente, una brecha de seguridad puede matar a la planta que se este cultivando, ya que, si se modifican los parámetros de los sensores, se podría regar más o menos de lo que se necesita, e iluminar más o menos de lo necesario.

### **2.3.5. Big data**

El big data, o la cantidad masiva de datos, consiste en recoger una enorme cantidad de datos, para posteriormente analizarlos y poder sacar conclusiones. Cambia por completo el método científico utilizado hasta ahora ya que este se basa en proponer una hipótesis, y trabajar para demostrar que esa hipótesis es posible. En cambio, el big data funciona al revés, se tiene el resultado final gracias a utilizar la gran cantidad de datos, y se sacan conclusiones de los resultados. Hoy en día, las grandes empresas tecnológicas como google, Facebook y Amazon, recopilan toda la información que pueden de cada usuario para su propio beneficio. Para poder aumentar las ventas ofreciendo anuncios personalizados, vender los datos a otras empresas que se encargan de distribuirlos, y tener un mayor control sobre cada individuo, anticipándose a lo que cada uno vaya a hacer. Esto lo consiguen gracias a que buscan similitudes de personalidad entre diferentes usuarios, y cuando uno realiza una acción, al resto de perfiles similares a este, le ofrecen la misma acción, antes de que este individuo lo haya pensado. Por eso da la sensación de que estas empresas saben lo que cada persona quiere antes de ser consciente de que lo quiere. Este análisis tan profundo de los datos ha supuesto una grandísima revolución, llegando al punto de que los partidos políticos han necesitado la ayuda de empresas que analizan datos (como Cambridge Analytica), para conseguir los votos de las personas que dudaban a quien votar, mediante una campaña de anuncios personalizados a cada individuo. Al comienzo de todo esto, ceder los datos personales no suponía ningún riesgo, y el pensamiento común era: ¡Que más me da lo que hagan con mis datos!, hoy en día, viendo que por todos los datos que compartimos pueden llegar a influenciarnos para que tomemos una decisión u otra en un proceso electoral, tenemos que preocuparnos más. Es por esto, que cada vez más los datos personales se tienen que proteger con leyes, para que no se vendan entre empresas tan fácilmente, y proteger al usuario de campañas de este tipo.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Lo mencionado anteriormente es un claro ejemplo de la fuerza que tiene el big data en esta sociedad, y lo importante que es saber utilizar, gestionar y analizar grandes cantidades de datos. Igual que las empresas tecnológicas lo utilizan para su propio beneficio, es importante aprender a usar las mismas herramientas, para usarlo en este proyecto.

## 2.4. Conclusiones sobre el estado del arte

Gracias a los diferentes proyectos mencionados en el apartado 2.2, se puede comprobar cual es el contexto actual de la fabricación de un huerto inteligente, cuales son las tecnologías utilizadas, cuales han sido algunas propuestas teóricas acerca del desarrollo de este proyecto, y cuales son las opciones comerciales para adquirir un huerto inteligente directamente. Este trabajo tiene el objetivo que se diferencia del resto, de realizar un análisis de las tecnologías para proponer un desarrollo a bajo coste con una solución funcional y autónoma. Como se ha comprobado en proyectos como Click and Grow, ya existen huertos inteligentes caseros pero el coste es bastante elevado. Por ello, se pretende ofrecer una solución más económica apta para usuarios que no tienen grandes conocimientos en estas tecnologías. Para esto, el desarrollo del producto tiene que ser de forma manual, obteniendo todos los componentes por separado y construyendo paso a paso la solución. También se considera importante la estética del producto final, lo cuál puede elevar el presupuesto. En caso de que no se busqué esta estética del producto, y solo la parte económica, también se dará una posible solución, incorporando al desarrollo solamente los productos estrictamente necesarios.

Gracias a los proyectos que se han analizado, se ha llegado a una serie de conclusiones.

- Qué microcontroladores utilizar (Raspberry vs Arduino)
- Qué sensores utilizar (temperatura, luz, humedad, etc.)
- Qué módulo WIFI utilizar
- Materiales para fabricar el huerto (madera vs plástico)
- Estética (colores, altura, escalable)

Además, gracias a la visión aportada en el apartado 2.3 de los diferentes habilitadores digitales, se ha hecho un análisis de los que se consideran imprescindibles para este proyecto.

- Sensores y microcontroladores. Necesarios
- Impresión 3D. No es necesario
- Internet of Things (IoT). Necesario
- Ciberseguridad. Importante, pero no necesario
- Big data. No es necesario.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Gracias a comparar los diferentes proyectos se puede valorar qué productos son necesarios y cuáles son prescindibles para abaratar los costes del huerto inteligente.

- La gran mayoría de los proyectos de huertos inteligentes, utilizan para la medición del correcto funcionamiento del huerto los siguientes sensores: humedad, temperatura, luminosidad y humedad de suelo. Además de estos, otros proyectos incorporan más sensores, pero estos 4 sensores mencionados anteriormente son los considerados como imprescindibles para el desarrollo de un huerto inteligente.
- El sistema de riego necesita de actuadores para poder mover el agua. En el caso del proyecto de Carles Alexander (Figura 6) se puede ver como el sistema de riego consiste en colocar un tanque de agua en altura y que el agua caiga por un tubo, en cambio en el TFG de Ata Jaber (Figura 5), el sistema de riego se basa en una pequeña bomba de agua dentro del tanque de agua que la suba a las macetas. Ambos sistemas de riego son muy útiles y se deben tener en cuenta para el desarrollo del trabajo.
- Una funcionalidad que aparece en algún proyecto muy interesante es la posibilidad de incluir la monitorización del huerto inteligente mediante una cámara IP conectada a la plataforma IoT. Esta cámara solo puede conectarse a una placa de Raspberry y es el principal motivo por el cuál se podría incluir esta placa. El coste de la placa de Raspberry es de alrededor de 40€ y de la cámara de unos 30€, solo incorporar esta implementación supondría el 70% del presupuesto inicial, por lo que debido a este motivo se ha decidido no utilizar ni la monitorización visual mediante cámara IP ni la adquisición de la placa de Raspberry.
- Los módulos para realizar la conexión a internet más utilizados han sido: Arduino YUN, Módulo ESP8266-ESP01 y NodeMCU ESP32S.
  - En el caso del proyecto donde se ha utilizado Arduino YUN además de la placa principal de Arduino. Esto supondría utilizar dos placas diferentes de Arduino, lo cuál elevaría el coste. Además, gracias a esta placa se puede realizar una conexión ethernet, es decir, una conexión por cable a internet. En el caso de este proyecto, esta solución no es compatible, ya que se quiere lograr un huerto completamente autónomo, sin la necesidad de utilizar cables para conectar dispositivos por fuera del propio huerto, ni para conexión a internet, ni para la alimentación de la placa.
  - El módulo ESP8266-ESP01 es un módulo que se puede conectar a la placa de Arduino y proveer comunicación WIFI mediante los comandos AT. Es un módulo muy económico, por lo cuál podría ser utilizado para este proyecto.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

- Placa ESP32S. Esta placa tiene un coste muy económico y no solo permite la conexión a internet mediante WIFI si no que también permite la conexión con diferentes sensores a su placa. Además de la placa ESP32S, hay otra placa muy similar ESP8266, que a diferencia que el módulo WIFI anterior, también permite la conexión de diferentes dispositivos. Es otra opción muy viable.
- Tanto en el prototipo final de Ata Jaber y de Carlos Alexander, se puede comprobar como las diferentes partes del huerto se encuentran separadas. Por un lado, el tanque de agua que vaya a regar el cultivo, por otra parte, la placa de Arduino y los sensores, y por otra parte la maceta con las plantas. En el caso de este proyecto se quiere desarrollar un huerto inteligente que este todo junto en un mismo módulo, en lugar de que haya varias partes separadas.
- Analizando los diferentes proyectos, se quieren mejorar en diferentes partes que no se han encontrado en este análisis.
  - En primer lugar, la autonomía del sistema. Todos los sistemas necesitan de alimentación por cable a un enchufe o al ordenador para poder funcionar. La idea de este proyecto es que el sistema sea completamente autónomo.
  - En segundo lugar, los proyectos que se han analizado llevan a cabo el desarrollo de una aplicación móvil para poder visualizar los datos de los sensores y poder activar los actuadores. Ya que este proyecto esta enfocado para cualquier usuario, no se le va a exigir que se desarrolle una aplicación nativa para un dispositivo móvil (además de que la mayoría de los desarrollos han sido para un solo sistema operativo, o Android o IOS). Se quiere hacer uso de alguna plataforma IoT de código abierto y con licencia gratuita que pueda utilizarse tanto en el dispositivo móvil como en un ordenador, sin necesidad de realizar el desarrollo de la aplicación.

En tercer lugar, la escalabilidad del proyecto. El caso de que se quieran incorporar más huertos para realizar diferentes cultivos no se ha tenido en cuenta en los anteriores proyectos que han sido analizados. En este trabajo se quiere realizar el desarrollo pensando en la escalabilidad de los proyectos elaborando un sistema que permita la incorporación de nuevos huertos inteligentes. La idea principal es poder construir un huerto vertical, colocando un huerto encima de otro, sin la necesidad de adquirir una estantería (como el proyecto de huerto vertical ClickAndGrow).

Por tanto, mediante el análisis de los diferentes proyectos y de los diferentes habilitadores digitales en la industria 4.0, se ha llegado a la siguiente conclusión.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

El proyecto se realizará mediante una placa de Arduino utilizando los sensores de humedad y temperatura DHT11, un sensor de fotorresistencia LDR (sensor de luminosidad) y un sensor para comprobar la humedad del suelo. El módulo WIFI mediante el cuál se realizará la conexión a internet sería o el módulo WIFI ESP8266-ESP01, o una placa ESP32S o ESP8266.

En cuanto a la plataforma IoT que se quiere utilizar, se tiene que buscar una plataforma gratuita que funcione tanto en un ordenador como en dispositivos móviles, ya que en la mayoría de los proyectos analizados se realizaba el desarrollo de una aplicación nativa desde 0.

En cuanto al diseño del huerto, se quiere hacer a base de madera y permitiendo la incorporación de otro huerto encima. Además, todos los componentes del huerto tienen que ir integrados dentro del huerto, no puede haber elementos externos que se salgan del propio huerto, como tanques para el agua o la conexión mediante cables a las placas o al router para el internet.

## 3. Descripción general de la contribución del TFM

### 3.1. Objetivos

#### Objetivo general

Diseñar e implementar un huerto inteligente autónomo de bajo coste conectado a una aplicación IoT.

#### Objetivos específicos

1. Estudiar y analizar las diferentes tecnologías relativas a la Industria 4.0
2. Diseñar un prototipo de huerto inteligente a bajo coste
3. Fabricar un producto final, útil y tangible.
4. Realizar un tutorial para la implementación del huerto.
5. Aprender sobre el desarrollo de un sistema con varios habilitadores digitales conectados.

### 3.2. Metodología del trabajo

Se van a utilizar dos metodologías de trabajo para este proyecto, POC y SCRUM.

La prueba de concepto o POC consiste en el desarrollo de un prototipo de prueba antes de realizar el producto final. Gracias a esto, todos los errores y problemas que surjan durante el desarrollo de la prueba se tendrán en cuenta de cara al desarrollo del producto final, de esta manera se minimiza el tiempo de desarrollo y el coste de sensores o elementos que finalmente no fuesen necesarios.

La metodología SCRUM se utiliza principalmente para el desarrollo software, que en este caso es una parte del proyecto, pero también se puede aplicar para cualquier tipo de desarrollo. En este caso, se va a hacer uso de la metodología para establecer los diferentes objetivos, plazos y reuniones dentro del desarrollo del proyecto.

#### 3.2.1. POC

Las pruebas de concepto se ejecutan en 6 fases diferentes.

- Preparación y configuración. Para realizar la prueba de concepto, se van a conectar los sensores que se quieran utilizar a una placa de Arduino y se va a conectar a la plataforma IoT. La comunicación entre la placa de Arduino y la plataforma IoT tiene que ser bidireccional, de tal forma que la plataforma IoT pueda mostrar datos obtenidos

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

desde la placa de Arduino, y desde la plataforma IoT se pueden mandar datos hasta la placa de Arduino, como por ejemplo, un botón para activar el riego. De esta forma se probará el funcionamiento del sistema electrónico. Una vez este todo correctamente configurado, se llenará una maceta con tierra para probar que los sensores miden correctamente los datos que se quieren obtener.

- Kick-off y planificación. Los tiempos de desarrollo de esta prueba de concepto son: 1 día para la comprobar la funcionalidad de los sensores, 3 días para comprobar los datos en una planta estable, y 2 semanas para comprobar que funciona correctamente en el huerto inteligente. Los recursos que se necesitan son: placa de Arduino con el modulo necesario para la conexión wifi, sensores de medición y maceta para el huerto inteligente. Las funcionalidades esperadas son: el correcto funcionamiento de los sensores, correcta integración de los sensores con una planta, con el huerto inteligente y con la plataforma IoT.
- Implementación. En esta fase, se pone en práctica las pruebas definidas en las fases anteriores. Montaje de sensores y pruebas en el huerto inteligente.
- Ejecución de pruebas. En esta fase se comprueba que los datos obtenidos de la implementación son correctos y coherentes. En caso de que no lo sean, se tienen que arreglar y configurar de forma correcta.
- Evaluación de resultados. Una vez que la fase anterior este completada, se evaluarán los imprevistos que hayan podido surgir y se plantearán posibles mejoras que puedan incluirse. En caso de que haya alguna mejora viable, se introducirá en el proyecto de pruebas, antes de comenzar el desarrollo final.
- Conclusiones. Una vez implementadas todas las mejoras necesarias, y que se haya comprobado durante varios días, que el proyecto funciona correctamente, se dará por concluida la prueba de concepto. En caso de que todo haya salido bien, y el resultado sea el esperado, se procederá al desarrollo del proyecto final.

### 3.2.2. SCRUM

La metodología SCRUM se divide en diferentes plazos de entrega denominados sprints. Estos tienen una duración de 2, 3 o 4 semanas según lo decidan los implicados. Las personas que participan en esta metodología son, la figura del product owner, scrum master y los developers. En el caso de este proyecto, el product owner sería la institución que hace posible el desarrollo de dicho proyecto, la universidad UNIR. El scrum master sería la persona encargada de supervisar el desarrollo del proyecto de los developers y que responde a las peticiones del product owner, es una figura intermediadora entre ambas partes, en el caso de este proyecto es la tutora del TFM, Aurora Saez Manzano. Por último, el equipo de developers consta de una sola persona, que es el autor de este proyecto, César Peón Lamparero.

Al comenzar cada sprint se realiza una reunión llamada sprint planning meeting, donde se planifican los objetivos que se van a realizar durante el sprint y se dividen en pequeñas tareas. Lo óptimo de esta reunión es que exista un documento con todas las tareas pendientes de hacer (backlog), y entre el scrum master y el product owner decidan cuáles son las tareas más prioritarias, y darles una estimación de tiempo a cada una. De esta forma, según la duración del sprint, la velocidad del equipo de desarrolladores y de la duración de cada tarea, se planificará hacer unas u otras.

Entre el equipo de desarrolladores y el scrum master, debería de haber una reunión diaria (daily scrum), para comentar los problemas encontrados con cada tarea, y organizar el día a día de cada desarrollador. Esta reunión no será posible en el desarrollo de este proyecto.

Al finalizar el sprint, se tendrá una reunión entre el scrum master y el product owner para comprobar como ha ido el sprint y si ha dado tiempo a realizar todas las tareas que se habían planificado. Una vez terminada esta reunión, el scrum master se reúne con los desarrolladores para comentar como ha ido la reunión con el scrum master, y estimar el tiempo de las tareas nuevas que se vayan necesitando. Esta reunión se llama sprint retrospective meeting.

Tras esto, se volvería a realizar una reunión de planificación, y se comenzaría un nuevo sprint. Cabe destacar, que cada sprint que se realice la velocidad del equipo y el tiempo en que se tarda en hacer cada tarea se ira mejorando más y más, ya que, en las primeras reuniones, no se sabe con exactitud esos valores.

La metodología scrum separa los diferentes plazos de entrega de los objetivos en sprints. En este proyecto, la duración de cada sprint es de 1 mes aproximadamente desde el comienzo del cuatrimestre.

- Sprint 1. 14 de septiembre – 14 de octubre. Investigación.

El 14 de septiembre es el día en el que se pondrán en contacto el scrum master con el equipo de desarrollo, por lo cuál, es el día en el que se comienza el proyecto. El 14 de octubre es el final del primer sprint, que coincide con la entrega del borrador inicial.

El objetivo principal de este primer sprint consiste en investigar más sobre la propuesta inicial en caso de que el proyecto haya sido aprobado. En caso contrario, se tendrá que pensar otra propuesta diferente siguiendo las indicaciones del scrum master.

Además de investigar, se comenzará con el desarrollo de la memoria del proyecto.

- Sprint 2. 18 de octubre – 25 de noviembre. Desarrollo.

El 18 de octubre se comenzará con el segundo sprint después de una reunión donde el scrum master dará indicaciones sobre el primer sprint al equipo de desarrollo.

El objetivo principal de este segundo sprint, es comenzar con el desarrollo de la prueba POC del huerto inteligente, y avanzar con la memoria del proyecto. Además, se tendrán que modificar las aportaciones comentadas por el scrum master en la reunión.

- Sprint 3. 3 de diciembre – 23 de diciembre. Desarrollo.

El 3 de diciembre se tendrá una reunión similar a la reunión del segundo sprint, en la que se comentará como ha ido el desarrollo del proyecto, y se darán indicaciones para el siguiente sprint.

El objetivo principal del tercer sprint es terminar la prueba POC y comenzar con el proyecto final, cerrando de esta forma también el desarrollo de la memoria. Además, se tendrán que modificar las aportaciones comentadas por el scrum master en la reunión.

- Sprint 4. 10 de enero – 2 de febrero. Pruebas.

El día 10 de enero se tendrá una reunión para comprobar el desarrollo del proyecto y de la memoria y se comentarán los detalles más importantes a tener en cuenta antes de la entrega final.

El objetivo final de este sprint es modificar la memoria con los comentarios del scrum master, y realizar diferentes pruebas con el proyecto final para ampliar la memoria, con posibles errores encontrados, como solucionarlos, y añadir imágenes del desarrollo del proyecto del huerto inteligente.

- Fecha límite de entrega – deathline. 2 de febrero – 9 de febrero.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

En esta fecha se tiene que entregar la memoria del proyecto final, donde se revisará que todo este correcto y se comentarán detalles mínimos para hacer el deposito final del proyecto el día 9 de febrero.

### 3.3. Descripción general de las partes o componentes de la propuesta

La propuesta de este trabajo consiste en fabricar desde cero un huerto inteligente de forma casera, de tal forma que el desarrollo de este proyecto se pueda adaptar a la economía de cualquier persona, sin la necesidad de pagar una gran cantidad de dinero por un huerto inteligente que ya este fabricado por una empresa.

Gracias a este proyecto, se va a poder realizar este huerto inteligente de una forma cómoda y sencilla para cualquier persona, sin necesidad de conocimientos técnicos. Además, se documentarán los problemas que surjan durante el desarrollo, para que el usuario, en el caso de tener un problema, sepa como resolverlo.

#### **Alcance y limitaciones**

El alcance de este proyecto es fabricar un huerto inteligente desde cero. Desde que se compran los componentes, hasta que se puedan plantar una planta, y el sistema monitoree por completo el crecimiento de dicha planta.

Las limitaciones que tiene este proyecto por falta de tiempo, es la creación de una base de datos pública donde cada usuario pueda especificar los mejores parámetros para cada sensor en función de cada planta y añadir mayor seguridad subiendo los valores obtenidos de los sensores a la “blockchain” de IOTA.

#### **Listado de participantes**

En el caso de este proyecto, no hay ningún colaborador ni ninguna entidad externa a la UNIR que participe en el desarrollo.

#### **Tecnologías implicadas**

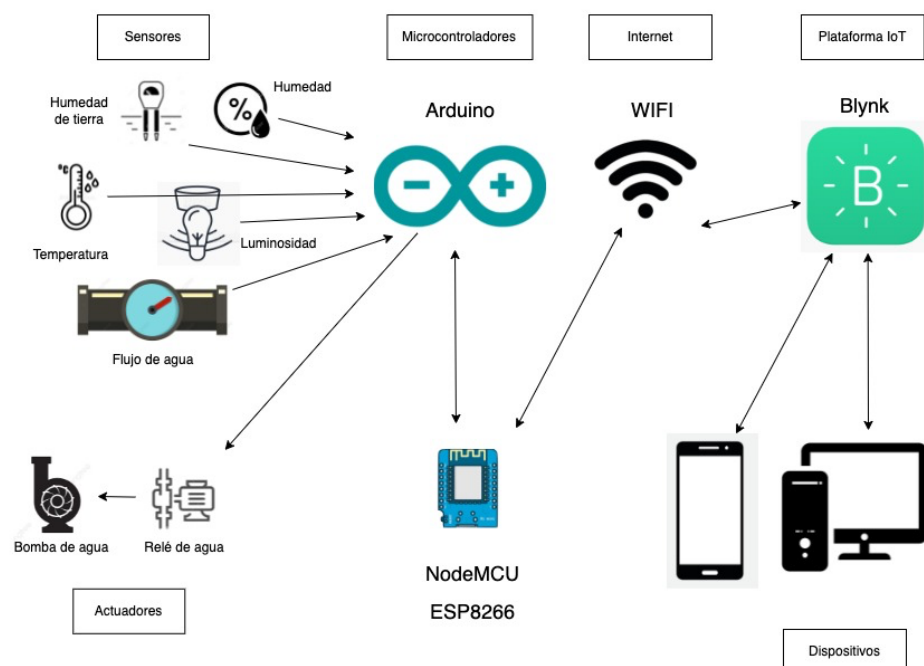
Las tecnologías implicadas principalmente son

- Robótica: Se va a utilizar una placa de Arduino para gestionar la información de los diferentes sensores.
- Sensores IoT: Se van a utilizar varios sensores conectados a una placa de Arduino, para obtener información sobre la planta que se encuentra dentro del huerto inteligente.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

- Internet of Things: Se utilizará una plataforma IoT para la conexión entre los diferentes sensores, y un ordenador/móvil donde el usuario podrá comprobar el estado de su planta.
- Ciberseguridad: En el caso que haya tiempo, una vez terminado el desarrollo completo del huerto inteligente, se almacenarán los datos en la blockchain de IOTA para aportar mayor seguridad a los datos enviados por los sensores.
- Impresión 3D: Esta tecnología se utilizará de forma opcional, en el caso de que el usuario disponga de una en su casa. No es imprescindible para el desarrollo del huerto inteligente.
- Big data: Esta tecnología podrá usarse, en el caso de que, de tiempo, para registrar los parámetros más adecuados a cada tipo de planta. Cada usuario que fabrique su propio huerto inteligente siguiendo este tutorial, mandará automáticamente y de forma anónima, los datos de sus plantas a una base de datos. Con toda esta información, se podrá determinar cuales son los mejores parámetros para cada planta.
- Inteligencia artificial: Unido a la tecnología anterior, y en caso de que diese tiempo a desarrollarlo, se crearían técnicas de machine learning, para que los parámetros se ajusten automáticamente en función de los datos de todos los usuarios que estén utilizando este sistema.

#### Arquitectura, componentes e integración de tecnologías.



**Figura 7: Integración de las tecnologías. Fuente: Elaboración propia.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

### **Resultados esperados**

Los resultados esperados consisten en ofrecer un tutorial que sirva de ayuda para que cualquier persona pueda fabricar un huerto inteligente, y en el desarrollo funcional de un huerto inteligente. A nivel técnico, se espera llevar las nuevas tecnologías habilitadoras de la industria 4.0 a cualquier usuario, y poder enseñar como funciona la agricultura inteligente a pequeña escala.

### **Presupuesto y retorno esperado de la inversión**

El coste de la adquisición de los materiales para poder realizar el huerto inteligente tiene que ser inferior a 100€. El coste esperado ronda los 70€.

En función de la planta que se quiera plantar, tendrá un coste u otro en el mercado, a la vez que diferente tiempo de crecimiento y de maduración del fruto. Además, hay que tener en cuenta, que ya se va a realizar de forma manual, el huerto se puede hacer tan grande como cada uno quiera, por lo que se obtendrá más fruto cuanto mayor sea el tamaño del huerto inteligente. Por todo esto no se puede estimar el retorno de la inversión de forma clara.

Por ejemplo, si se quisiesen plantar fresas, tardan aproximadamente 10 meses en crecer, pero cada planta da aproximadamente un kilo y medio de fresas (Mundo huerto, 2021). Por lo que si se plantan plantas de fresas en un huerto inteligente (ya que la planta no es muy grande), serían 3 kilos de fresas al año. El precio de la fresa por kilo depende de la temporada del año, pero suele rondar los 4€ el kilo. Por lo que estas 2 plantas, equivaldrían a 12€ al año.  $70/12 = 5,8$  años en recuperar la inversión.

Otro ejemplo serían los rábanos, los cuáles tardan en salir 20 días aproximadamente (Hydroenv, s.f.). El precio del manojo del rábano es de 1,8€ en los supermercados. Teniendo en cuenta, que en un huerto similar al anterior ejemplo en el que entrarían dos plantas de fresas, entrarían 6 manojos de rábanos. Por lo que, en menos de un mes, habríamos obtenido lo equivalente a 10,8€. En un año, 356 días, se obtendrían 18 tiradas de 6 manojos de rábanos, lo que serían 108 manojos, que a un coste de 1,8€ cada uno, supone un coste de 194,4€ en un año. Teniendo en cuenta que el huerto inteligente ha costado alrededor de 70€, la inversión se ha recuperado en  $70/194,40 = 0,36$  años, es decir, en 4 meses aproximadamente.

Como se ha podido comprobar gracias a los ejemplos, el retorno de inversión depende mucho de la planta que se quiera plantar, el tamaño del huerto inteligente, la cantidad que se obtenga

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

en cada recolecta y el precio de la fruta de la planta en función de la temporada. Por todo esto, no se puede estimar un retorno de inversión claro al coste de los materiales.

### **Planificación general**

El tiempo estimado de las tareas están definidos por los sprints de la metodología scrum.

- Sprint 1. 14 de septiembre – 14 de octubre. Investigación.
- Sprint 2. 18 de octubre – 25 de noviembre. Desarrollo.
- Sprint 3. 3 de diciembre – 23 de diciembre. Desarrollo.
- Sprint 4. 10 de enero – 2 de febrero. Pruebas.
- Fecha límite de entrega – deathline. 2 de febrero – 9 de febrero.

## 4. Desarrollo específico de la contribución

El desarrollo de este proyecto se ha realizado bajo la metodología SCRUM, la cuál reparte el trabajo a realizar en diferentes fechas de entregas, o sprints. En este punto, se va a describir como se ha realizado el trabajo en cada sprint, y cuáles han sido los problemas encontrados en cada uno, así como el avance que se ha dado.

### 4.1.1. SPRINT 1: Investigación

En esta primera parte del proyecto, se ha buscado información de proyectos similares para obtener referencias de como construir el huerto inteligente. Poder comprobar que sensores se van a utilizar y cuales no son necesarios.

Gracias a las conclusiones obtenidas en el apartado 2 de este trabajo, los elementos que van a formar parte del proyecto son los siguientes

#### Sensores

- Temperatura y humedad: DHT11
- Humedad de tierra
- Luminosidad: Fotorresistencia LDR
- Módulo wifi: ESP8266-ESP01, ESP32 o ESP8266. Ya que estos módulos tienen un coste y funcionamiento similar, se realizarán pruebas con ambos y se determinará cuál es el mejor en la prueba de concepto.
- Placa de Arduino para conectar los sensores.
- Sensor de flujo de agua.

#### Actuadores

- Bomba de agua
- Relé de 1 canal

En cuanto a la plataforma IoT, en la investigación se habían analizado proyectos que utilizaban aplicaciones propias desarrolladas por los propios usuarios que construían el huerto inteligente. En el caso de este trabajo se quiere hacer uso de una aplicación externa gratuita para la visualización y monitorización de los datos de los sensores. Hay varias plataformas que lo permiten.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

- FIWARE: es la plataforma principal para el desarrollo y despliegue de aplicaciones de internet del futuro. En esta plataforma, se tendría que crear una aplicación para poder utilizarse, por lo que no se va a ser utilizada.
- Blynk: esta plataforma permite conectar el mismo dispositivo en el ordenador y en el móvil, y siempre y cuando no se conecten más de 2 dispositivos, es gratuita. En este caso, solo se va a conectar 1 dispositivo, por lo que sería gratuita.
- Adafruit.io: Esta plataforma permite la visualización e interacción mediante la aplicación web, es decir, es accesible desde un navegador, pero no tiene aplicación nativa en los dispositivos móviles.
- Amazon, google y Microsoft. Estas 3 grandes compañías tecnológicas tienen una plataforma IoT dentro de sus respectivos servicios cloud.
  - Amazon Web Service (AWS) es el servicio cloud de Amazon, y dentro se encuentra la plataforma AWS IoT Core destinada al internet de las cosas.
  - Por el lado de Google, Google Cloud Platform es el servicio cloud y Google Cloud IoT es su plataforma para el internet de las cosas.
  - Microsoft Azure es el servicio cloud de Microsoft, y dentro se encuentra su plataforma IoT

El funcionamiento de estas 3 plataformas es muy similar entre sí, el coste es gratuito, o se cobra por el uso de internet que suele ser un coste muy bajo, pero para poder hacer uso de sus servicios hay que registrarse en la plataforma con una tarjeta válida.

Hay muchas plataformas IoT gratuitas como Thingspeak o MyDevices Cayenne, y otras plataformas que son de pago, pero permiten una prueba gratuita durante un tiempo como Temboo, Udibots o Kaa. Pero tanto estas que se han comentado como las mencionadas un poco más en profundidad anteriormente, sirven para el desarrollo de este trabajo, ya que tiene un uso básico. Recoger los datos de los sensores y activar el flujo de agua mediante un botón, para un solo dispositivo.

Finalmente, la plataforma IoT seleccionada en Blynk, ya que permite una suscripción gratuita indefinida con aplicación web para el pc y aplicación nativa para dispositivos móviles, y no requiere el desarrollo y despliegue de una aplicación propia. Además, la propia plataforma ayuda a conectar el dispositivo inicial con la placa que se este configurando con varios ejemplos y con amplia documentación.

## Diseño del huerto

- De madera, ya que es más resistente que el plástico, y permite la posibilidad de realizar un jardín vertical a base de colocar varios huertos uno encima de otro, y la estética de la madera también favorece frente a otros materiales.
- Escalable. Permitiendo que la posibilidad de poder colocar un huerto encima.
- Todos los componentes dentro de la estructura del huerto inteligente. Sin tener ningún módulo externo, como el tanque de agua, o tener la placa de Arduino y los sensores apartados.
- Autónomo. No debe de ir enganchado a la corriente mediante un cable de alimentación, ni conectado a internet por cable (ethernet).

### 4.1.2. SPRINT 2: Desarrollo I

En este segundo sprint se ha fabricado la prueba de concepto (POC) mediante la cuál se pretende saber cuales son los elementos finales que se van a utilizar en el proyecto final.

Para realizar la prueba de concepto se han utilizado los siguientes sensores

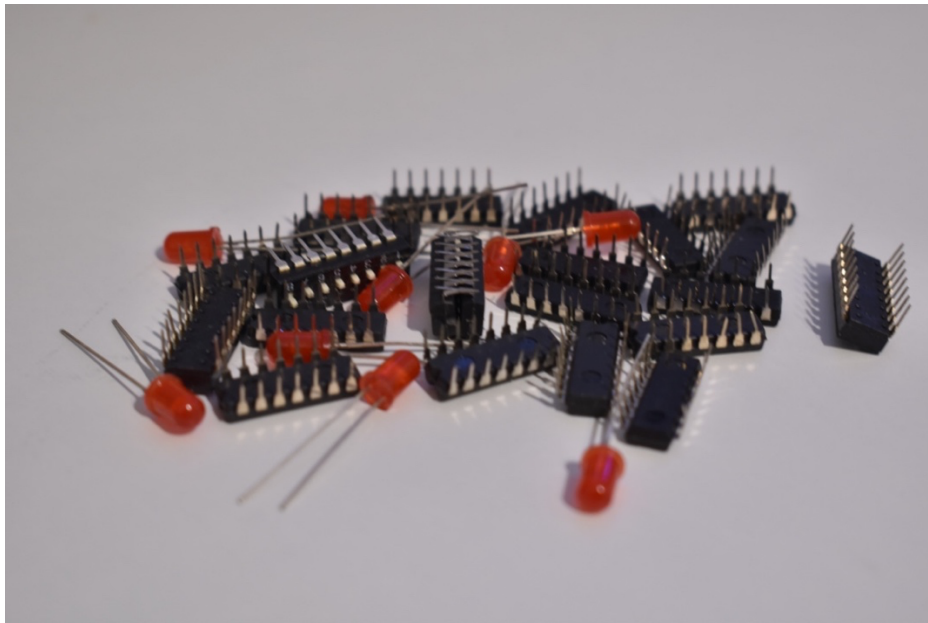
#### 4.1.2.1. Sensores

Kit básico de Arduino sobre el cual se montarán el resto de los sensores. Dentro del kit básico de Arduino, además de la propia placa de Arduino, la protoboard y el cable que se muestran en la siguiente figura, también se incluye una gran cantidad de cables, resistencias y leds.



**Figura 8: Arduino. Fuente: Elaboración propia.**

Además de los leds que vienen en la caja, se han comprado más leds por si fuesen necesarios y diferentes switches para complementar los accesorios de la protoboard, así como pulsadores, leds numéricos y una pila de petaca por si fuese necesario para dar corriente.



**Figura 9: Accesorios Arduino I. Fuente: Elaboración propia.**



**Figura 10: Accesorios Arduino II. Fuente: Elaboración propia.**

El siguiente elemento que se ha comprado es un sensor para el control del flujo del agua, para medir el caudal de agua que pasa por él y poder tomar decisiones sobre cuando dejar de bombear agua.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.



**Figura 11: Sensor flujo de agua. Fuente: Elaboración propia.**

Se ha comprado una pantalla RGB para mostrar por pantalla los datos obtenidos de los sensores, y a posteriori, utilizar el led para indicar si el usuario tiene que hacer algo con el huerto. Como, por ejemplo, avisar si se le tiene que regar, si la tierra tiene demasiada humedad, o en el caso de que se haya conseguido configurar el riego automático, si el tanque de agua se ha agotado.

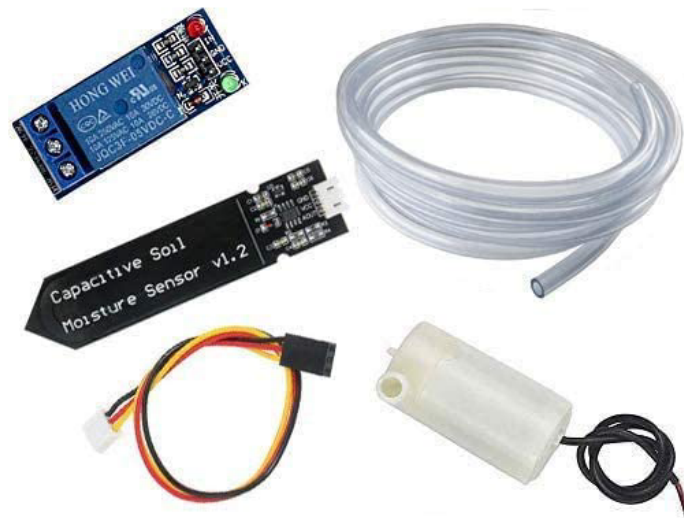


**Figura 12: Pantalla RGB. Fuente: Elaboración propia.**

Para el sistema de riego se va a utilizar un kit que consta de una bomba de agua para mover el agua del tanque por un tubo, de un relé de 1 canal que se encarga de recibir la orden de si

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

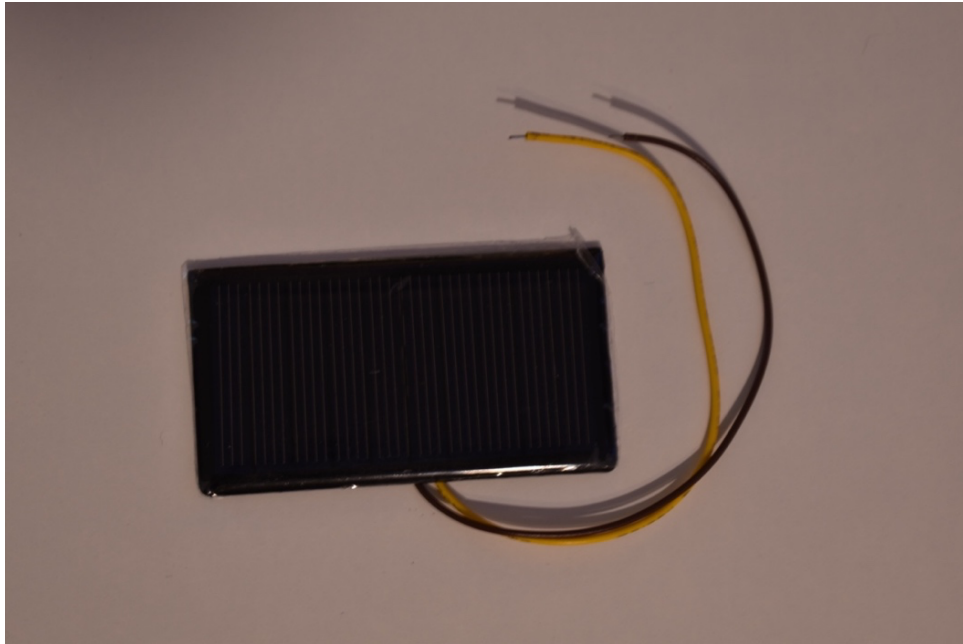
activar o no la bomba de agua, y un sensor de humedad de tierra para poder tomar decisiones en cuanto a la activación de dicha bomba.



**5. Figura 13: Sensor de humedad de tierra, bomba de agua y relé. Fuente: (Runcci-yun, 2019).**

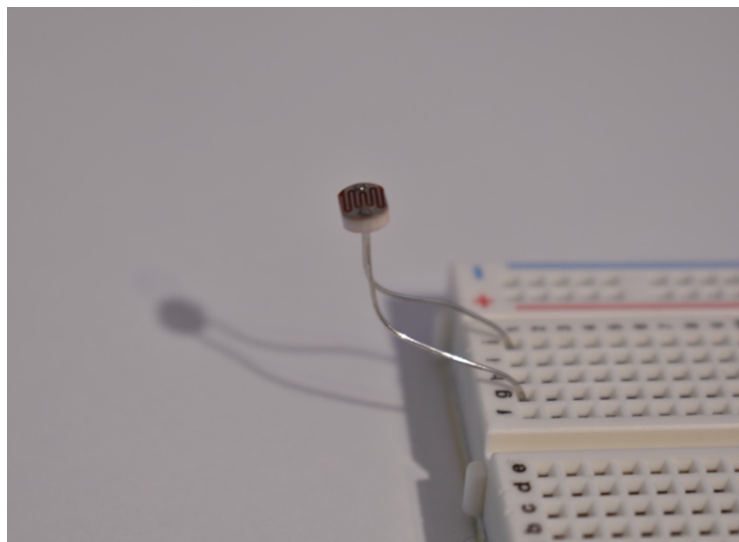
La idea principal es no tener conectada la placa de Arduino constantemente al ordenador, ni para que mande los datos de los sensores ni para que se mantenga enganchada a la corriente. Se quiere que el huerto inteligente sea autónomo y se valga por si mismo para suministrarse energía y para mandar los datos a la plataforma IoT.

Para el suministro de la corriente de forma autónoma, se van a hacer dos pruebas en esta prueba de concepto, la primera mediante pila. En este caso una pila de petaca que ya se ha enseñado en una Figura anterior, y la segunda opción es mediante una pequeña placa solar. Aunque esta opción puede ser mala idea si el huerto se construye en interior y a la casa no le da el sol por ninguna ventana. También hay que comprobar si con la energía que da la placa se mantiene todo activado constantemente. La placa solar que se va a utilizar para hacer las pruebas es la siguiente.



**Figura 14: Placa solar Arduino. Fuente: Elaboración propia.**

De forma general, el huerto inteligente debería de estar lo más cercano a una ventana para recibir la luz solar y aprovechar esa energía. Para comprobar la luminosidad de la zona, se necesita un sensor de fotorresistencia LDR, el cuál tiene una resistencia la cuál se modifica con el aumento o disminución de la luz incidente. El sensor fotorresistencia que se va a utilizar es el siguiente.

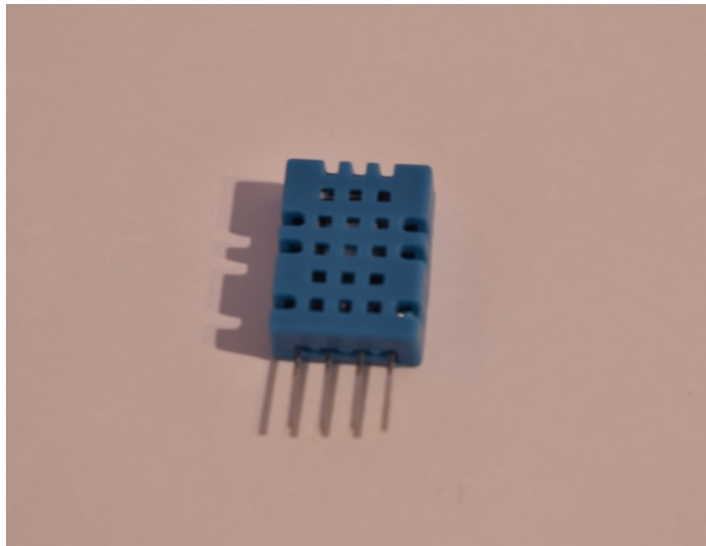


**Figura 15: Sensor fotorresistencia. Fuente: Elaboración propia.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

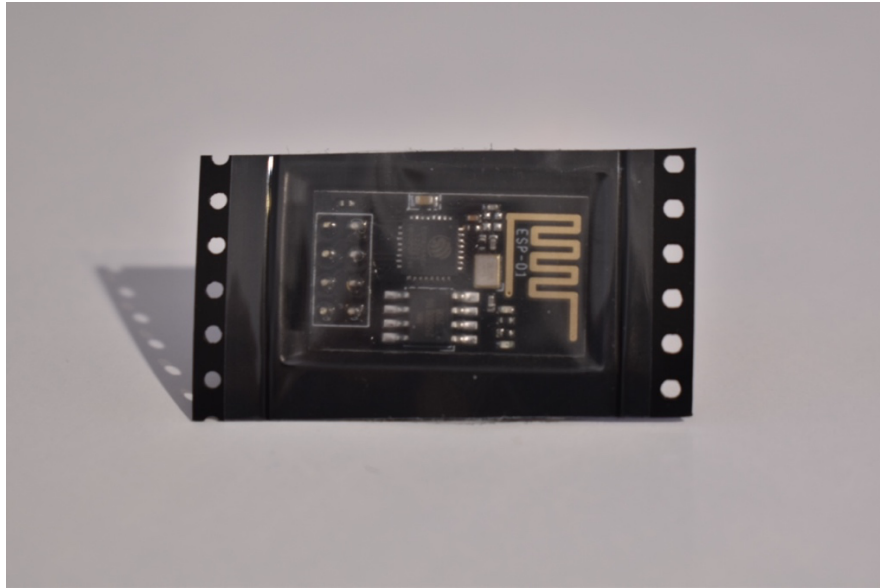
Anteriormente se ha enseñado el sensor que se encarga de medir la humedad de la tierra donde estará la planta y esto comunicara al huerto inteligente si se necesita más agua o no, pero la humedad y la temperatura ambiente también son datos necesarios para la toma de decisiones. Si el huerto se encuentra en una zona húmeda (por ejemplo, dentro de un cuarto de baño después de una ducha, o si se vive en una zona con un clima húmedo), aunque la humedad de la tierra sea baja, no se necesitará echar tanta agua que si se tratase de un clima seco. Por el lado de la temperatura pasa lo mismo, si la estación del año es verano y las temperaturas son altas, la planta necesitará más agua, que si la estación es invierno y las temperaturas son más bajas que en el verano.

Para poder medir la temperatura y la humedad ambiente, se necesita el sensor dht11 que registra ambos datos. El sensor es el siguiente.



**Figura 16: Sensor dht11 de humedad y temperatura. Fuente: Elaboración propia.**

El módulo Wifi ESP8266 (ESP-01) permite a la placa de Arduino que se pueda conectar a internet sin la necesidad de estar conectado al ordenador mediante un cable.



**Figura 17: Módulo wifi ESP8266-ESP01. Fuente: Elaboración propia.**

Otro módulo mediante el cuál se va a realizar las pruebas de conexión a internet es el módulo ESP8266, que es muy similar al ESP32. Gracias a la comparativa mostrada en el siguiente artículo (Descubre arduino, 2021) se puede comprobar como la funcionalidad de ambos dispositivos es similar, y que el ESP32 tiene mejores prestaciones, tiene mayor velocidad del wifi, más pines para realizar la conexión, más canales ADC y permite la conexión por bluetooth, entre otras cosas, pero el coste es algo superior. Como en este proyecto, las prestaciones que ofrece el ESP8266 son suficientes y es más económico, es la placa que se va a utilizar para realizar las pruebas.



**Figura 18: Placa ESP8266. Fuente: Elaboración propia.**

Estos son todos los elementos que se han adquirido para realizar la prueba de concepto (POC), a raíz de las pruebas que se vayan realizando con estos elementos, se comprobará cuales son necesarios y cuales no, cuales no hacen su función correctamente y cuales sí, y si se necesitan otros elementos adicionales para realizar algunas de las funciones básicas que se quieren realizar.

#### 5.1.1.1. Pruebas

Objetivo principal: Probar varios sensores y conectarse a la plataforma IoT para mandar los datos obtenidos.

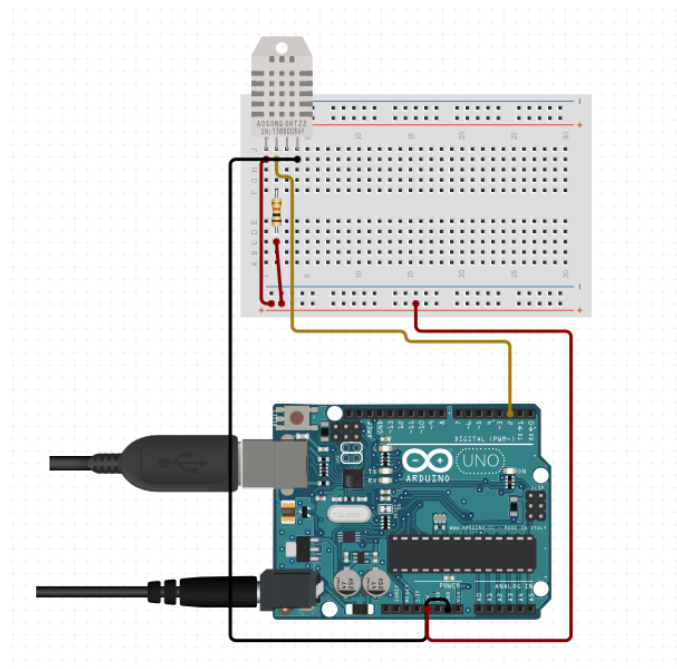
En primer lugar, se realiza la conexión de los sensores con la placa de Arduino, recogiendo los datos obtenidos en la terminal del IDE de Arduino. Una vez que se reciban todos los datos correctamente, se realizara la conexión WIFI mediante las dos placas seleccionadas (ESP8266-ESP01 y ESP8266) para verificar cuál es la correcta para este caso. A continuación, se realizará la transmisión de datos desde la placa de Arduino hacía la placa o módulo WIFI. Después, se conectará a la plataforma IoT Blynk para mandar los datos recibidos de los sensores. Una vez que el sistema este funcionando con el flujo de datos Sensores – Arduino – WIFI – Blynk, se realizarán los ajustes necesarios para poder realizar un flujo de datos bidireccional, de tal forma, que desde una acción realizada desde la plataforma IoT se puedan dar ordenes a los sensores. Esto se quiere hacer para poder activar

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

el sensor de flujo de agua de forma manual si el usuario lo desea, y que no solo se active de forma automática. Una vez que esto se haya conseguido, se habrá dado por finalizada la prueba de concepto.

#### 5.1.1.1.1. Configuración de los sensores con la placa de Arduino

En primer lugar, se va a realizar la conexión con el módulo DHT11 de temperatura y humedad. Este sensor tiene 4 pines que se tienen que conectar a la protoboard. El primero es el que recibe la corriente de 5V, el segundo es el que manda la información al pin 2 de la placa de Arduino y esta conectado a una resistencia de 10k. El tercer pin esta vacío y el 4 es la toma de tierra. Siguiendo el siguiente esquema.



**Figura 19: Configuración DHT11 a Arduino. Fuente: Elaboración propia mediante circuito.io.**

Como en la placa de Arduino, va a haber varios sensores que necesitarán de la toma de corriente de 5V y de la toma de tierra, se conecta un cable desde la salida de 5V de la placa de Arduino, a una fila de color rojo de la protoboard (+), y de la misma forma, se conecta un cable desde la placa de Arduino en el pin GND hasta una fila de color azul de la protoboard (-). Todos los sensores que necesiten corriente o tierra se conectarán a estas filas de la protoboard, en vez de conectarse directamente a la placa de Arduino.

Una vez se ha realizado la conexión física, se pasa a la parte software. En un ordenador, hay que descargar el IDE de Arduino de la página oficial: <https://www.Arduino.cc/en/software>

Una vez descargado y abierto el programa, hay que instalar una librería para poder usarla en el código. Esta librería se instala desde Herramientas/Administrar Bibliotecas.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

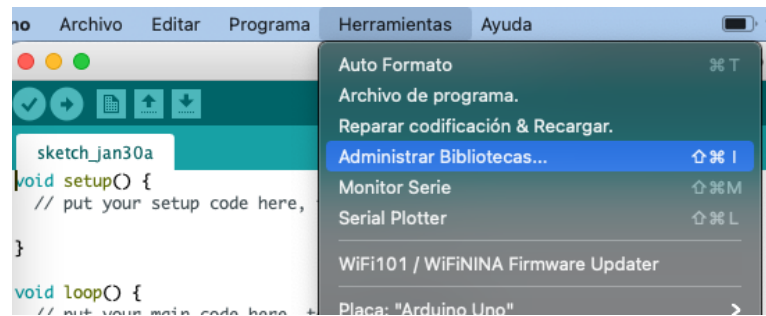


Figura 20: Instalar bibliotecas Arduino. Fuente: Elaboración propia mediante IDE de Arduino.

En la barra del buscador, hay que seleccionar la librería DHT sensor library de Adafruit.

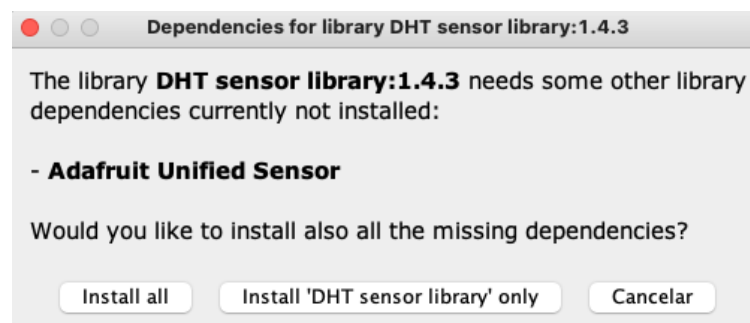


Figura 21: Librería DHT11 Arduino. Fuente: Elaboración propia mediante IDE de Arduino.

Una vez instalada, escribir el siguiente código que aparece en el Anexo A – 1.

Cuando se haya copiado y pegado el código, conectar el cable de datos de Arduino a un puerto usb en el ordenador y configurar la placa de la siguiente forma. En Herramientas/Placa seleccionar Arduino Uno y en Herramientas/Puerto seleccionar el puerto usb en el que este enganchado el cable, por lo general suele poner Arduino Uno entre paréntesis. Una vez configurado, compilar el código pulsando el botón ✓ del IDE de Arduino, y cuando se haya terminado de compilar el código, pulsar en el botón de la flecha derecha, que se encuentra justo a la derecha, y subir el código a la placa.

Cuando se haya terminado de subir, abrir el terminal de Arduino pulsando el botón que hay a la derecha con forma de lupa, y observar que los datos resultantes tienen sentido.

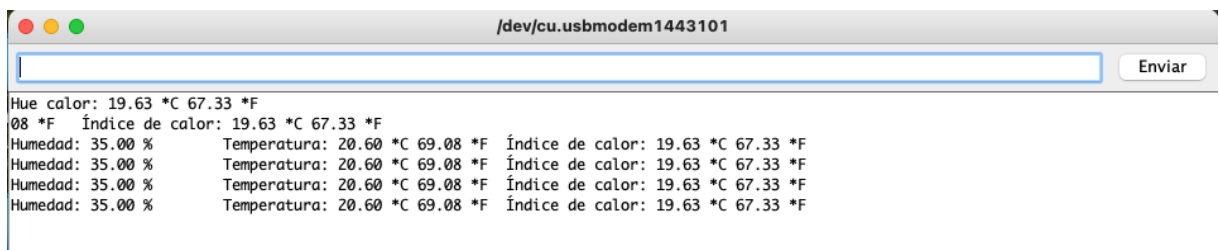
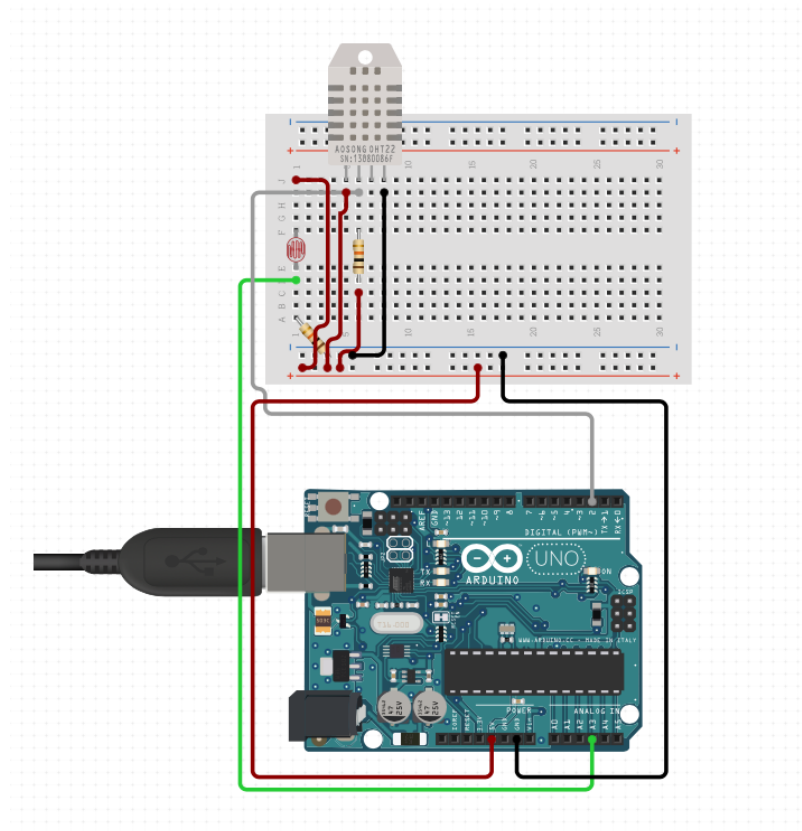


Figura 22: Resultado DHT11. Fuente: Elaboración propia mediante IDE de Arduino.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.


Después de configurar el sensor DHT11 de temperatura y humedad, se va a configurar el sensor de luminosidad. Para esto se va a necesitar el sensor de fotorresistencia LDR y haber configurado el montaje anterior.

En el caso del sensor DHT11 se ha utilizado un pin digital para registrar los datos en la placa de Arduino, en el caso del sensor LDR se necesita un pin analógico. Al igual que antes, el voltaje necesario será de 5V y una resistencia de 10k. La configuración de los cables es la siguiente: Una parte del sensor conectada al voltaje (corriente de 5V de la protoboard), y la otra conectada a una resistencia y al pin de Arduino analógico A3. Desde la resistencia un cable conectado a la parte negativa de la protoboard. El modelo final es el siguiente.

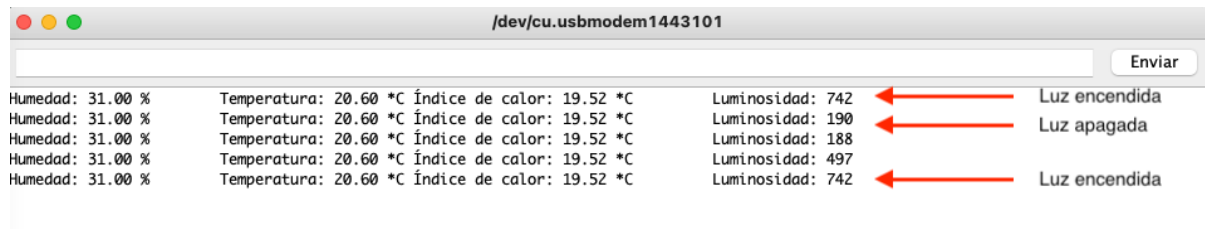


**Figura 23: Sensor LDR a Arduino. Fuente: Elaboración propia mediante circuito.io.**

Cuando la conexión de los cables se haya realizado se escribe el código que se encuentra en el Anexo A – 2. En este script de código se ha configurado el sensor de temperatura y el de luminosidad a la vez, no solo el del sensor de luminosidad.

Al igual que antes, se conecta el cable de datos a la placa de Arduino, se compila el código pulsando el botón  y cuando haya terminado, se sube el código a la placa pulsando el botón de la flecha derecha. Cuando se haya terminado de subir, se abre la terminal pulsando el botón de la lupa, y se tendrían que recuperar unos datos similares a estos.

## IoT aplicado al ámbito de la agricultura inteligente en el hogar.



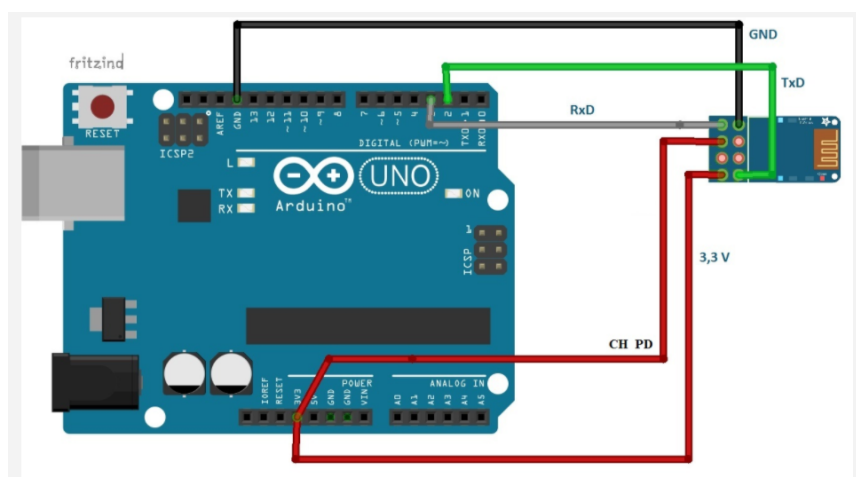
**Figura 24: Datos del sensor de luminosidad. Fuente: Elaboración propia mediante IDE de Arduino.**

Se ha querido comprobar la conexión de dos sensores, uno digital y otro analógico para realizar esta prueba, pero no se considera necesario incluir más en la POC. El resto de componentes se incluirán en el desarrollo de la parte final.

#### 5.1.1.1.2. Configuración de la placa o módulo WIFI

En primer lugar, se van a realizar las pruebas con el módulo WIFI ESP8266-ESP01. Este es un módulo bastante económico que se conecta directamente a la placa de Arduino, al contrario que la placa ESP8266 la cuál se sirve por si misma para funcionar. El módulo ESP8266-ESP01, ESP01 a partir de ahora para diferenciarlo con la placa, obtiene la corriente mediante los cables conectados a la placa de Arduino, en cambio, la placa ESP8266 se puede conectar con un cable microusb para la corriente.

El módulo ESP01 no se puede conectar directamente a la protoboard para la conexión porque los pines no se corresponden con el espacio que tienen los agujeros de la protoboard, por lo que se necesitan cables macho-hembra, para poder realizar la conexión. En el kit básico de Arduino, se incluyen 5 de estos cables, los cuales son los necesarios para realizar la configuración. La configuración de los cables es la siguiente.



**Figura 25: Conexión módulo WIFI ESP8266-ESP01. Fuente: (Crespo, 2017).**

Para la configuración del módulo ESP01 se va a utilizar la otra mitad de la protoboard que no se está utilizando, a la cuál se le da una corriente de 3,3V en vez de 5V como ocurría en la configuración de los sensores. Esto se hace porque 2 pines tienen que ir conectados a la corriente. Otro irá conectado a la toma de tierra GND, y otros dos pines se encargan de realizar la comunicación serial mediante los cables Rx y Tx.

Después de realizar la configuración mediante los comandos AT necesarios para este módulo, y conectarlo a internet surgieron varios inconvenientes.

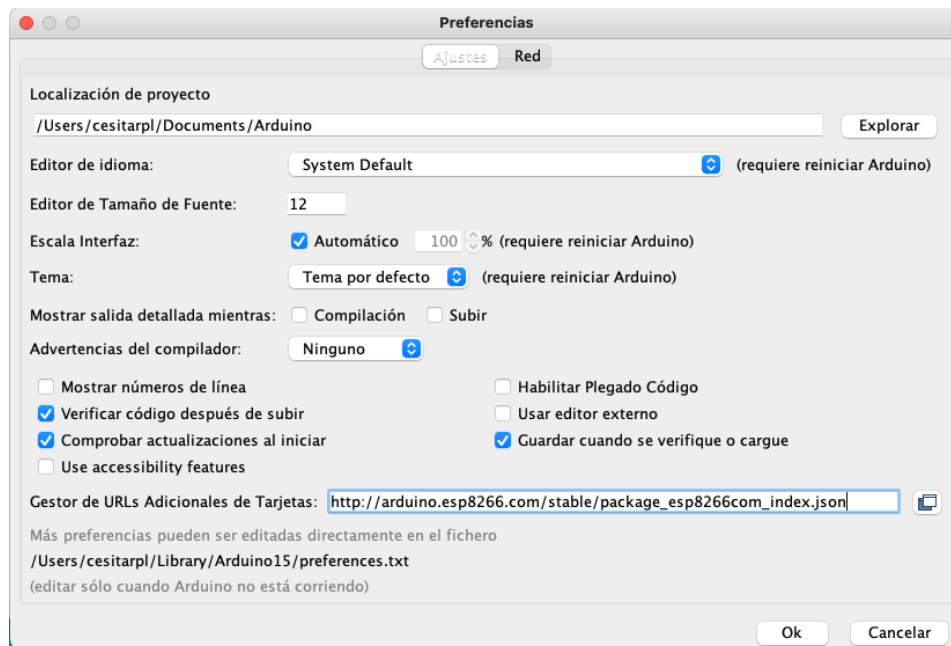
- El primer problema encontrado, es que para configurar el módulo ESP01 se tiene que realizar la configuración puesta anteriormente, pero para la configuración de los sensores, se tiene que realizar un cambio de cables. Esto significa que, para modificar los scripts de los sensores, y el del módulo ESP01, habría que estar cambiando los cables para cada prueba. Además de la configuración de los cables, hay que quitar el microcontrolador de la placa de Arduino con un destornillador, ya que, si se compila un código teniendo este módulo conectado, el código se subiría al microcontrolador de Arduino, en vez de subirse al módulo ESP01.
- El segundo problema, es que la comunicación que se ha mostrado anteriormente funciona en una dirección, es decir, la placa de Arduino puede mandar datos al módulo ESP01 por los cables de Rx y Tx, pero para poder realizar una comunicación bidireccional, se necesitaría conectar otros 2 cables más desde el módulo ESP01 hasta las entradas Rx y Tx de la placa de Arduino, los pines 0 y 1 digitales (los que se encuentran justo al lado del cable gris y verde que ya están conectados). Para esto habría que comprar más cables aparte que no vienen en el kit básico de Arduino.
- En último lugar, la conexión a internet se tiene que realizar mediante los comandos AT comentados anteriormente, para los cuales se tiene que instalar una librería, y abrir la terminal donde se irán poniendo los comandos correspondientes. Desde este lugar se realiza toda la configuración, en cambio, como se verá posteriormente, en el caso de la placa ESP8266 toda la configuración se realiza directamente sobre el script de código. En caso de que en algún momento se modificase la contraseña del router donde este conectado, el módulo ESP01 habría que volver a colocar los cables en forma de configuración, acceder a la terminal y con comando AT reconfigurar la contraseña, con la placa ESP8266 sería mucho más cómodo.

Después de intentar configurar el módulo WIFI ESP01 sin éxito, se procede a la configuración de la placa NodeMCU ESP8266. Para esto, al igual que la placa de Arduino se conecta por

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

un cable USB para poder subir el código, a esta placa se tiene que conectar un cable microusb.

Antes de poder compilar y subir un script de código a la placa, se tiene que realizar un par de configuraciones en el IDE de Arduino. En primer lugar, en Arduino/Preferencias/Gestor de URLs Adicionales de Tarjetas, se tiene que escribir la siguiente línea:  
[http://Arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://Arduino.esp8266.com/stable/package_esp8266com_index.json)



**Figura 26: Configuración NodeMCU ESP8266. Fuente: Elaboración propia mediante IDE de Arduino.**

Gracias a esto, cuando se vaya a seleccionar la placa en Herramientas, aparecerán los diferentes tipos de placas WIFI, y se podrá seleccionar la correspondiente. Una vez hecho esto, se tiene que instalar la librería que se va a utilizar esp8266. Para esto, entrar en Herramientas/Administrar Bibliotecas..., en el buscador buscar esp8266, y hay que instalar la librería: esp8266 by ESP8266 Community. Una vez que esto se haya instalado, se puede escribir el script de código que se encuentra en el Anexo A - 3. En dicho script solo se comprueba la correcta conexión con internet. Hay dos campos que se tienen que modificar, el primero es la ssid del router y el segundo la contraseña. Una vez cargado, se abre la terminal y se comprueba si se ha conectado o no a internet.

```
WiFi connected
IP address:
192.168.1.105
```

**Figura 27: Conexión correcta al WIFI Fuente: Elaboración propia mediante IDE de Arduino.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

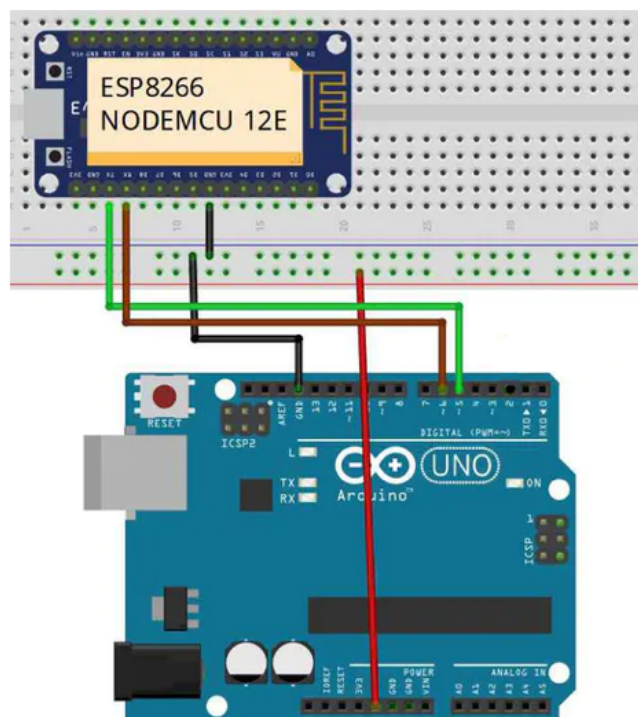
#### 5.1.1.1.3. Comunicación entre Arduino y WIFI

Una vez que se ha comprobado que la placa de Arduino recoge correctamente los datos de los sensores y la placa de WIFI se conecta a internet, es el momento de realizar la comunicación entre la placa de Arduino y la de WIFI para pasar datos de una a otra.

En el último paso se configura la comunicación entre la placa WIFI y la de Arduino, haciendo de esa forma una comunicación bidireccional, pero en este paso, la comunicación va a ir desde la placa de Arduino hasta la placa WIFI.

Para esto, tanto la placa de Arduino como la placa de WIFI van a estar conectadas al ordenador por cable, y cada una va a tener una pestaña del IDE de Arduino con su código independiente. Hay que tener en cuenta, que cada que se subía el código a la placa, había que cambiar la placa y el puerto desde herramientas, al igual que cada vez que se quería comprobar la salida por el terminal.

Para realizar la comunicación, desde la parte física, se han conectado los cables de los pines Rx y Tx de la placa WIFI con las salidas digitales de los 5 y 6 de la placa de Arduino. Ya que Arduino es el que va a mandar los datos por esos pines, y es la placa WIFI la que realiza la conexión serial. La conexión entre las placas se ha realizado gracias a la ayuda del siguiente artículo (DIY, 2020).



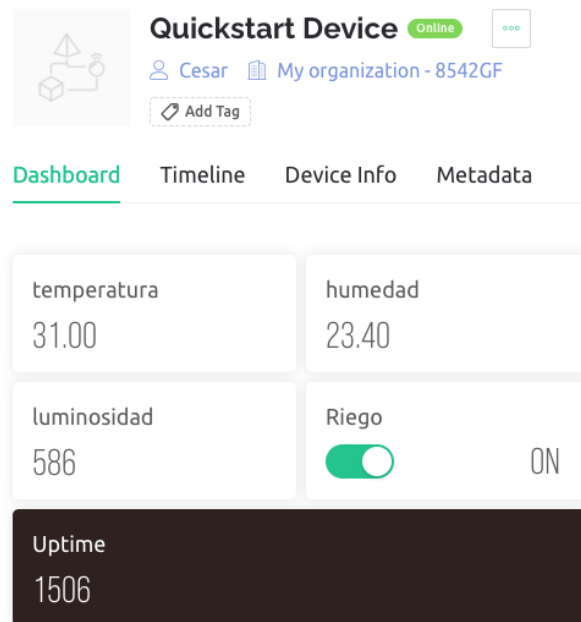
**Figura 28: Comunicación serial entre Arduino y ESP8266. Fuente: (DIY, 2020).**

Después de conectar los cables, se realiza el desarrollo de los scripts de código. El script que se tiene que subir a la placa de Arduino es el Anexo A – 4 y el que se sube a la placa de WIFI NodeMCU ESP8266 es el script del Anexo A – 5. En este código se puede ver como en el script de Arduino, se incluye la librería `SoftwareSerial` y se crea una instancia en los pines 5 y 6. Esto es para comunicar los datos desde esos pines. Por el otro lado, la placa NodeMCU ESP8266 lo escuchara siempre desde los pines reservados para esto, Tx y Rx. Después, la información que se quiere mandar desde Arduino se manda con la llamada a la instancia que se ha creado antes y llamando al método `print`.

Tras copiar y pegar los scripts de código, compilarlo y subirlo a la placa correspondiente, se puede comprobar desde la terminal de cada placa, como la información que se manda desde Arduino, es la misma que se recibe a la placa NodeMCU ESP8266.

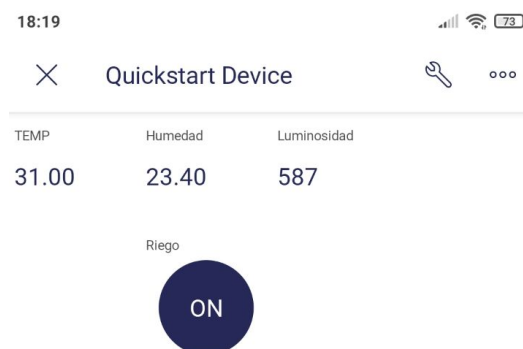
#### 5.1.1.1.4. Conexión con la plataforma IoT: Blynk

Después de conectar la placa Arduino con la placa NodeMCU ESP8266 se va a configurar la conexión con la plataforma IoT Blynk. Para esto, después de realizar el registro en la plataforma, se añade el primer dispositivo siguiendo unos pasos ya configurados en la plataforma, en el tercer paso se ofrece un script de código para copiarlo en el IDE de Arduino, además de pedir la instalación de la librería `Blynk by Volodymyr Shymansky`. Es importante copiar y pegar el script que genera Blynk, ya que la conexión de la placa con el dashboard del usuario se realiza mediante unos campos definidos que ya vienen rellenos, en el caso de copiar y pegar el script de este trabajo, habría que modificar esos campos con los datos personales (`BLYNK_TEMPLATE_ID`, `BLYNK_DEVICE_NAME`, `BLYNK_AUTH_TOKEN`). Una vez descargado el script, se añaden los valores del router, `ssid` y `password` en los campos correspondientes, y se divide la información que se recibe para mandar los valores por diferentes canales. De esta forma, se recibía un string así: `H= 30.00T=18,60L=791`, y ahora se subdivide en 3 valores diferentes: 30,00 / 18,60 / 791. Cada uno se manda por un canal correspondiente, por V4 la temperatura, 30,00 – por el canal V5 la humedad, 18,60 y por el canal V6 la luminosidad, 791. Desde el lado del navegador, se crea un dashboard con el 3 objetos de texto, cada uno en un canal nuevo, uno en V4, V5 y V6. De esta forma, al ejecutar el nuevo script, cuando se reciban los datos de la placa de Arduino, se mandarán por canales separados a la plataforma IoT. El script final se puede ver en el Anexo A – 6, y el resultado final en la plataforma IoT Blynk es la siguiente.



**Figura 29: Blynk en el navegador. Fuente: Elaboración propia mediante Blynk.**

El botón de riego se ha dejado configurado para el siguiente paso, pero no tiene ninguna funcionalidad para este paso. Además de la aplicación de navegador, Blynk también tiene una aplicación nativa para dispositivos móviles, y se va a hacer la prueba de que también se reciben los datos. Para ello, se descarga de forma gratuita desde la store correspondiente y se inicia sesión con los datos que se hayan registrado en la aplicación del navegador. Una vez dentro, hay que crear un dashboard similar al que haya en el navegador. El resultado final es este.



**Figura 30: Blynk en un móvil Android. Fuente: Elaboración propia mediante Blynk Android.**

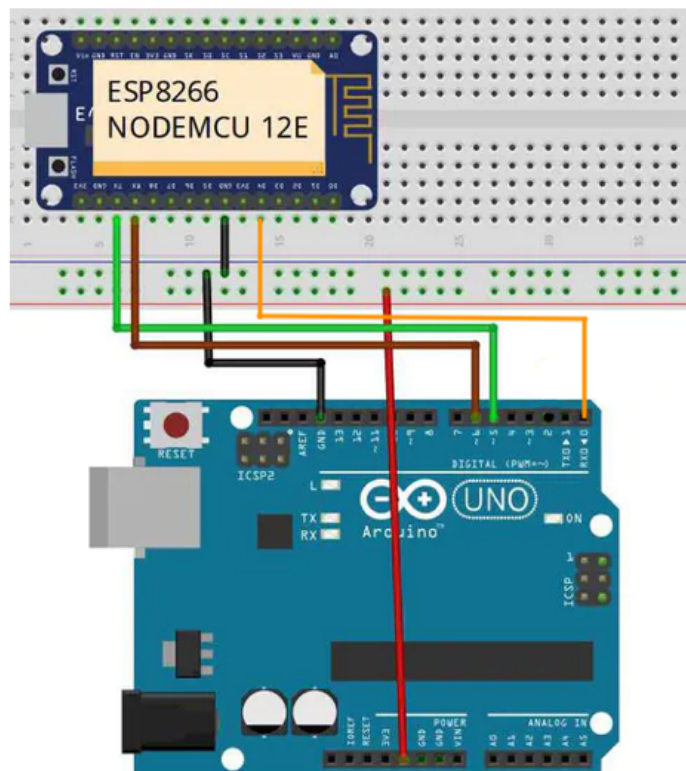
Como se puede comprobar, el nombre del proyecto es el mismo: Quickstart Device, y los datos registrados son similares.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

#### 5.1.1.1.5. Comunicación bidireccional

Una vez que se ha realizado la comunicación correcta entre la plataforma IoT con el dispositivo NodeMCU ESP8266, solo falta realizar una comunicación bidireccional entre la placa WIFI y la placa de Arduino, para que cuando se realice un cambio manual desde la plataforma IoT, se pueda ver reflejado en los sensores. Para esto, se va a realizar una prueba con un led de color azul conectado a la placa de Arduino, y en la plataforma IoT se ha colocado un botón llamado riego. Cuando se active este botón, se tiene que encender el led conectado a la placa de Arduino, y cuando se desactive apagarse. En el desarrollo del proyecto final, esto servirá para activar o desactivar el relé que activará la bomba de agua para regar el huerto.

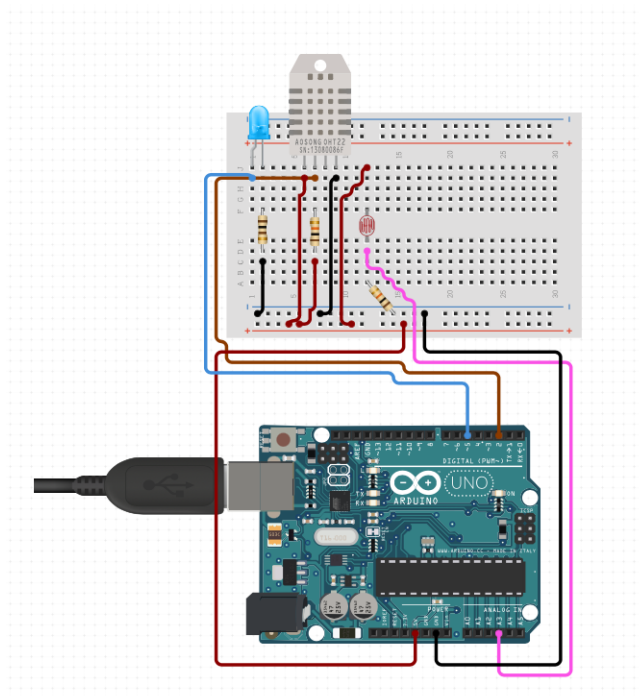
El script que se había proporcionado desde Blynk, ya esta preparado para mandar datos al dispositivo que este conectado mediante el método `BLYNK_WRITE` por el canal correspondiente, de tal forma, que cuando se pulsa el botón de riego, este método se activa. Así que lo que queda por hacer, es mandar esa información desde la placa de WIFI a la de Arduino. Ya que solo se va a mandar información de lectura, no se va a ejecutar nada directamente, no es necesario conectar el cable Tx en la placa de Arduino, solo sería necesario el cable Rx. Este ira conectado al pin 4 de la placa WIFI.



**Figura 31: Comunicación bidireccional Arduino-NodeMCU ESP8266. Fuente: Elaboración propia a partir de (DIY, 2020).**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Por el lado de la placa de Arduino, se ha añadido un led azul conectado por un lado al pin 5 de Arduino, y por el otro lado a una resistencia de 100 y a la tierra.



**Figura 32: Arduino con led azul. Fuente: Elaboración propia mediante circuito.io.**

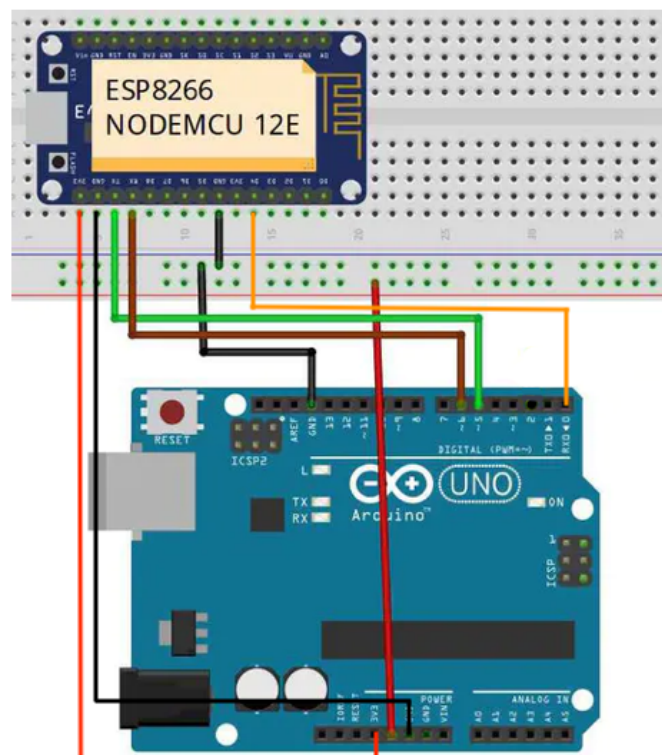
Una vez se han integrado todos los componentes físicos, se han añadido los scripts correspondientes. Para la placa de Arduino el script del Anexo A – 7, y para la placa NodeMCU ESP8266 el script del Anexo A – 8.

Para realizar la prueba final, se quiere dejar conectada las placas de forma autónoma, sin la necesidad de que la corriente llegue por el cable. De esta forma, se hace una primera prueba con la pila de petaca mostrada anteriormente. Dicha pila es de 4,5V lo cuál es un voltaje inferior a lo que se necesita. Según este artículo (Prácticas con Arduino, s.f.) el voltaje recomendado es de entre 9 y 12 V. Por esto, se ha comprado una pila de 9V y un adaptador de esta pila a la conexión Jack de la placa de Arduino.



**Figura 33: Pila 9V y adaptador. Fuente: Elaboración propia.**

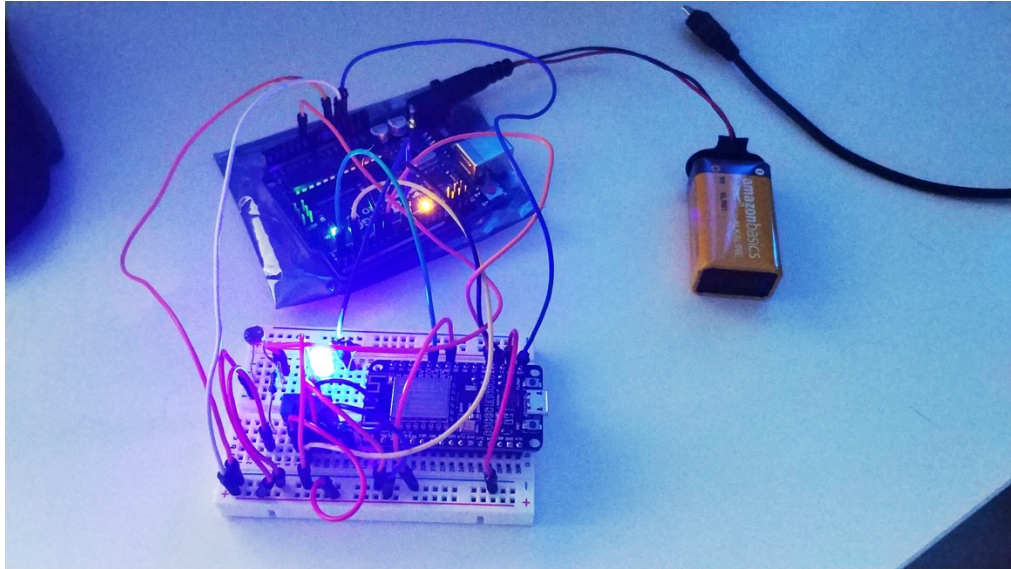
Una vez se han subido los anteriores scripts a cada placa, se desconectan los cables, y como la pila va a dar alimentación a la placa de Arduino, es esta la que tiene que dar la corriente a la placa WIFI. Esta placa no puede recibir más de 3,3V porque si no se puede quemar. Para realizar esta conexión, se lleva un cable desde la salida de 3,3V de la placa de Arduino a la entrada de corriente de la placa de NodeMCU ESP8266, y otro cable que conecte la toma de tierra. El resultado final es este.



**Figura 34: Esquema final Arduino - NodeMCU ESP8266. Fuente: Elaboración propia a partir de (DIY, 2020).**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Una vez esta todo configurado, se realiza la prueba de conectar la pila de 9V a la placa de Arduino, y cuando la plataforma IoT registre que el dispositivo NodeMCU ESP8266 esta conectado a internet, se pulsa el botón de riego para encender el led azul de la placa de Arduino.



**Figura 35: Resultado final Prueba de Concepto. Fuente: Elaboración propia.**

### **5.1.2. SPRINT 3: Desarrollo II**

En el sprint 3 se desarrolla el proyecto final de huerto inteligente, indicando paso a paso como se ha construido y los problemas que se hayan encontrado. Se va a estructurar en diferentes apartados. En primer lugar, la construcción manual del huerto, en segundo lugar, montaje de los sensores y cultivo de la primera planta, después se desarrollará el tutorial para la construcción de este huerto y, por último, recopilación de los problemas que han surgido.

#### **5.1.2.1. Fabricación del huerto**

Para la decisión del material de la infraestructura del huerto, se han valorado diferentes posibilidades para equilibrar la parte estética y funcional con la parte económica.

La forma más económica que se ha encontrado es mediante una botella de 5 litros de agua, a la cual se le corta la parte de arriba una vez esta tumbada. Dentro de la botella se rellena con la tierra y se siembra la planta. La sensorización con la placa de Arduino tendría que añadirse con un modulo aparte. En caso del que presupuesto que se tenga para realizar este huerto sea muy bajo, hasta el punto de que se priorice frente a algo mas funcional y estético, no se va a tener en cuenta.

La conclusión a la que se llegó en la parte de la investigación es que el huerto se va a fabricar con madera, porque funcional y estéticamente es mejor y queda más bonito. Una de las IoT aplicado al ámbito de la agricultura inteligente en el hogar.

principales ideas de este huerto es que se pueda colocar un huerto encima de otro, de tal manera que poco a poco se vaya creando un huerto vertical sin necesidad de adquirir una estantería adicional. La madera es un buen material para aguantar un huerto sobre otro y estéticamente también encaja con el desarrollo de un huerto inteligente. Además, es un material fácil para poder tratar de forma manual, y según el tipo de madera, el coste entraría dentro del presupuesto.

El tipo de madera que se va a utilizar para la fabricación de este huerto es madera de pino ya que es de los precios más económicos, junto a la madera del abeto y de la haya (Majofesa maderas, 2020).

En cuanto a la cantidad y el tamaño de la madera a adquirir cambia en función de la planta que se vaya a cultivar. En el caso del desarrollo de este huerto se va a plantar rábano y cebollino con lo que no se necesita gran altura, y con una profundidad de 20 cm mínimo valdría. El grosor mínimo de la madera debería de ser de 1 cm mínimo, 1,5 cm sería lo deseable y 2 cm como máximo. El grosor deseado no tiene que ser excesivamente grueso ya que, al trabajar directamente con la madera, cuanto menor sea el grosor más fácil será la fabricación. Además, el contenido del huerto es principalmente tierra, por lo que no tiene que soportar grandes cargas de peso.

El tamaño escogido para este huerto han sido 3 tablas de madera de pino de 120 cm y de largo, 30 de ancho y 1,8 cm de grosor. 2 tablas para realizar la estructura principal y la tercera para separar la parte de tierra con la placa de Arduino y montar el “techo” del huerto y posible soporte de un huerto que vaya a ir encima.



**Figura 36: Tablas de madera de pino. Fuente: Elaboración propia.**

Estas tablas se han cortado por la mitad para obtener 4 tablas de 60X30 cm. Una de estas será la base del huerto y dos estarán a los lados a lo largo de la base. De esta forma se consiguen 30 cms de profundidad, lo cual es suficiente para el cultivo del rábano.

La tabla de 60X30 se vuelve a cortar por la mitad, obteniendo dos tablas de 30X30, las cuales irán a los lados a lo ancho de la base.



**Figura 37: Tablas cortadas. Fuente: Elaboración propia.**

Una vez se han terminado de cortar, se colocan para comprobar como quedaría finalmente la estructura del huerto.



**Figura 38: Estructura del huerto. Fuente: Elaboración propia.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Por último, de la tercera tabla se ha cortado una pieza del mismo tamaño que las que están a lo ancho del huerto, de 30X30 para hacer una separación dentro del huerto entre la parte de la tierra, y donde estará la placa de Arduino con los sensores, y un tanque con agua.

El resultado final después de atornillar todas las partes es el siguiente.



**Figura 39: Estructura principal. Fuente: Elaboración propia.**

El resto de la tercera tabla se ha utilizado para hacer el techo con las patas que lo sujetan, y para hacer un pequeño cobertizo donde almacenar cosas que necesite el huerto. Como guantes, humus de lombriz, abono, sustrato o herramientas.

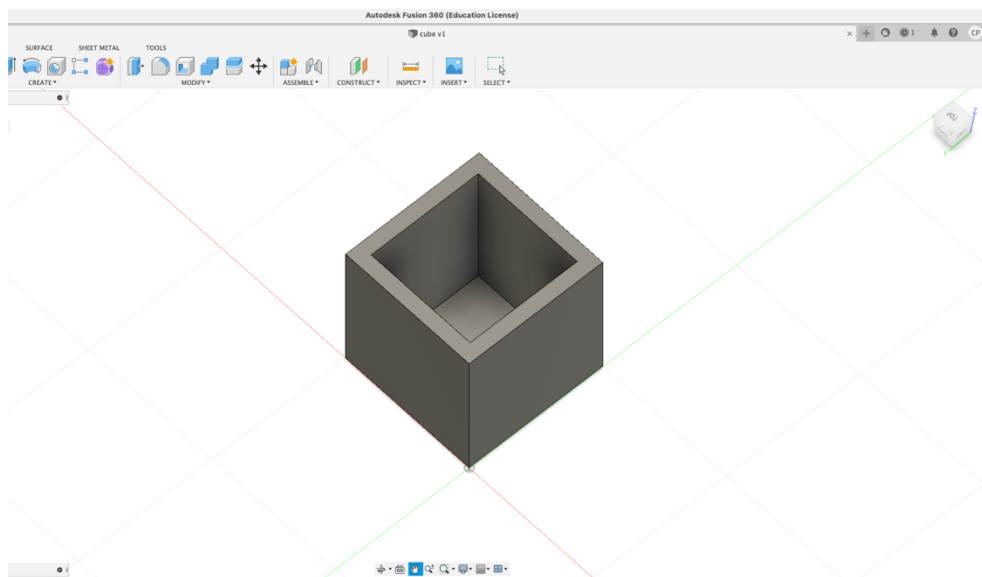
#### 5.1.2.2. Montaje del huerto inteligente

El desarrollo de este huerto inteligente va a necesitar de 2 tanques de agua, uno donde se va a almacenar el agua para regar el huerto, y otro tanque que ira conectado al sensor de flujo de agua para medir el caudal del agua, y poder parar la bomba cuando hayan pasado los litros necesarios. Como este segundo tanque de agua no es estrictamente necesario, ya que si no se quiere utilizar el sensor de flujo de agua no es necesario, se va a diseñar con impresión 3D, la cual tampoco es estrictamente necesaria para el desarrollo del huerto inteligente. Para el tanque de agua principal, se va a utilizar una botella de 2 litros de agua.

Para el segundo tanque de agua se va a realizar un diseño 3D a medida del huerto para optimizar al máximo el espacio.

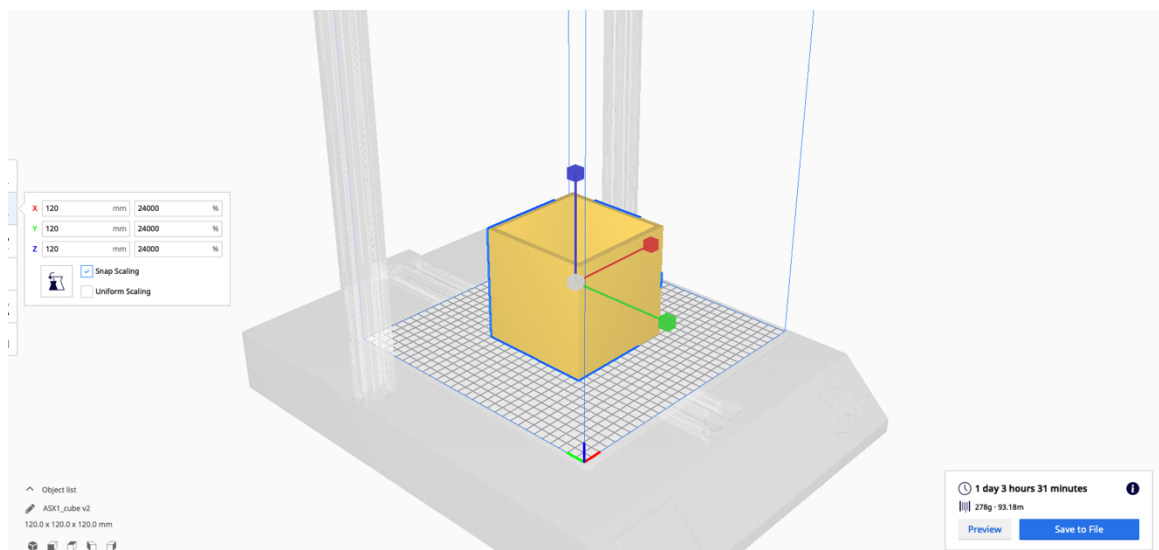
IoT aplicado al ámbito de la agricultura inteligente en el hogar.

El primer paso es el desarrollo del modelo 3D mediante la aplicación autodesk fusión 360.



**Figura 40: Modelo 3D tanque de agua. Fuente: Elaboración propia mediante Autodesk Fusion 360.**

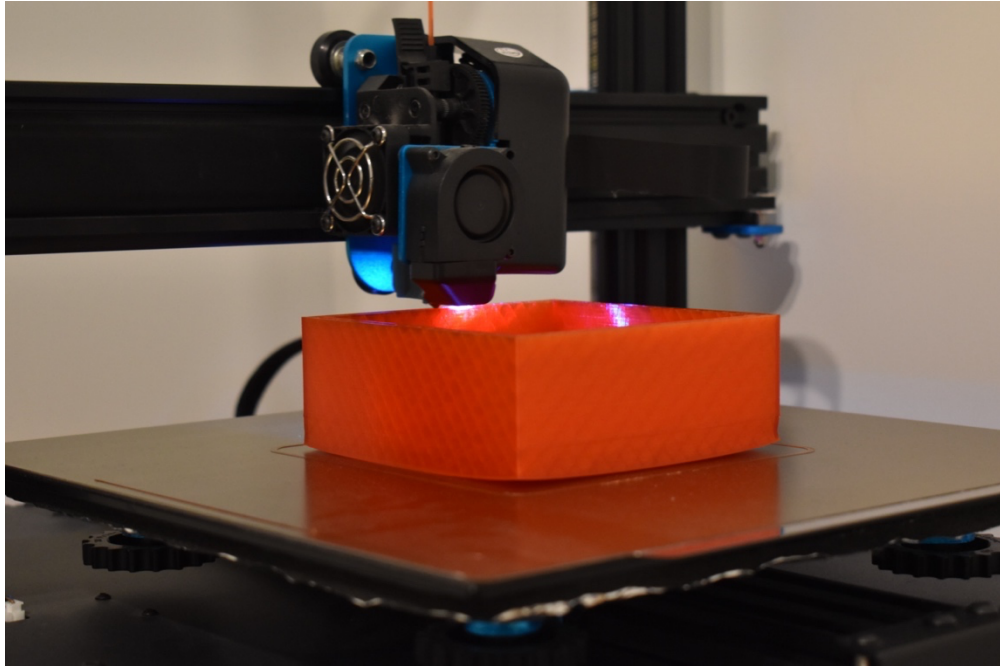
Una vez se obtiene el modelo 3D deseado, se exporta en un fichero .stl. Este fichero se introduce en la aplicación Ultimaker Cura para establecer las medidas del objeto y las características de impresión, como la velocidad de impresión, temperatura del extrusor y de la cama, etc.



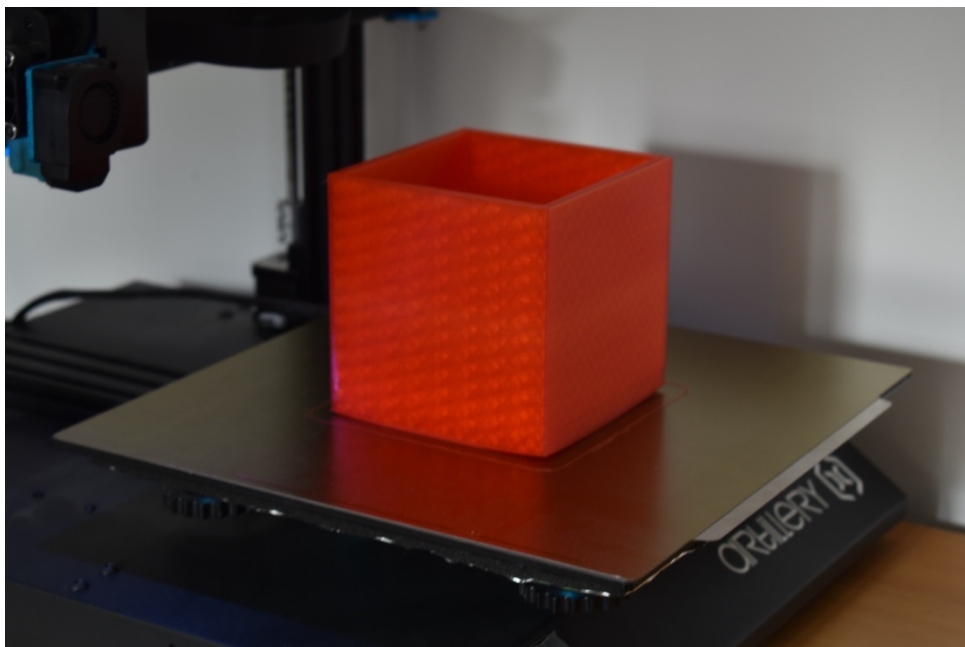
**Figura 41: Opción de impresión 3D en cura. Fuente: Elaboración propia mediante Ultimaker Cura.**

Como se puede comprobar en la Figura en la parte izquierda, el tamaño del objeto es de 120X120 mm. El tamaño que hay en el huerto para depositar el tanque es de 12,2 cm. De esta manera, se aprovecha al máximo el espacio entre las paredes del huerto. Por último, el IoT aplicado al ámbito de la agricultura inteligente en el hogar.

tiempo estimado de impresión es de 1 día, 3 horas y 31 minutos, y se hará un gasto de 218g del filamento PLA.



**Figura 42: Construcción del tanque de agua. Fuente: Elaboración propia.**



**Figura 43: Resultado final del tanque de agua. Fuente: Elaboración propia.**

Una vez se ha enfriado el tanque de agua, se hace un agujero en la parte baja del tanque para colocar el sensor de flujo de agua.

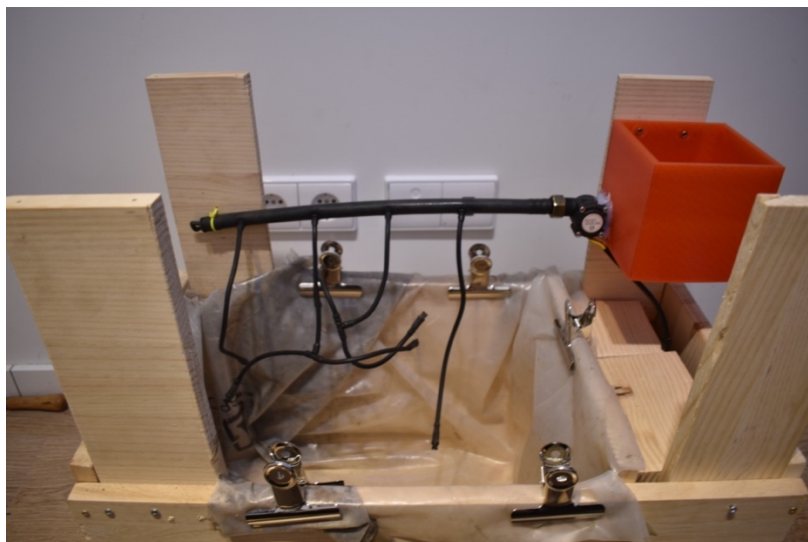
IoT aplicado al ámbito de la agricultura inteligente en el hogar.



**Figura 44: Tanque de agua y sensor de flujo de agua. Fuente: Elaboración propia.**

En la salida del sensor de flujo de agua, se ha instalado un sistema de tuberías para repartir el riego por todo el huerto. El sistema de tubería funciona por goteo y al no tener mucha altura este tanque de agua, la presión es baja y el paso de agua se ralentiza. Pero la velocidad con la que la bomba de agua sube el agua del tanque principal, es rápida. Gracias a este tanque de agua hecho con la impresión 3D, se puede almacenar el agua mientras que el sistema de goteo funciona lento sin problema de que se desborde el agua.

El resultado final, una vez están instaladas las tuberías y el tanque de agua acoplado al huerto es el siguiente.



**Figura 45: Tanque de agua y tuberías conectadas al huerto. Fuente: Elaboración propia.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Como se puede comprobar en la figura anterior, se ha colocado un plástico en el huerto de tal forma que la tierra no esta en contacto directo con la madera. La madera de pino se deteriora con humedad, por lo que se ha decidido poner una capa aislante para alargar la duración del huerto. Además, en caso de que haya agua sobrante con el sistema de riego, el agua caería directamente a la madera, o podría salirse y encharcar todo si no se ha cerrado bien. Gracias al plástico utilizado, el agua sobrante se quedaría abajo sin salirse.

Después, se ha llenado de tierra y humus de lombriz el huerto, y se ha colocado finalmente el techo del huerto como se puede ver en la siguiente figura.



**Figura 46: Huerto con tierra y techo. Fuente: Elaboración propia.**

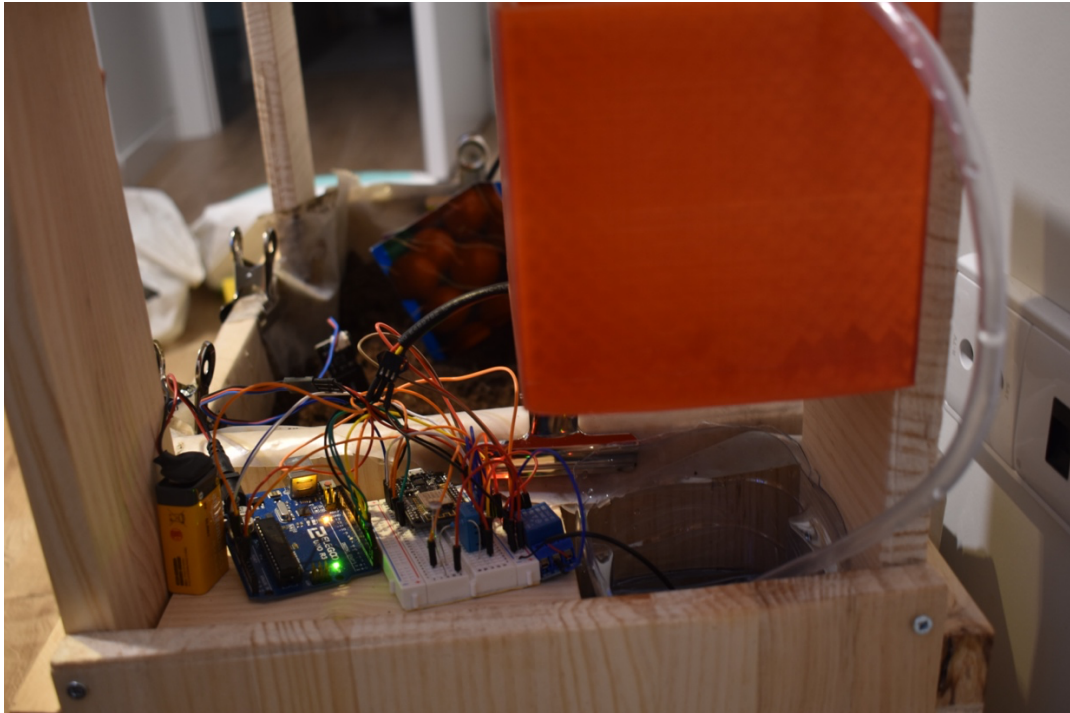
En su interior se encuentra las primeras semillas que se van a cultivar para realizar las pruebas. Son semillas de cebollino y de rábano, semillas las cuales tardan pocas semanas en cultivarse. En la parte de la derecha, debajo del tanque de agua, se encuentra el lugar donde se van a colocar los sensores y la placa de Arduino. Además, una tabla de madera que separarán los sensores del pequeño almacén que se ha dejado para guardar los materiales que se consideren necesarios.



**Figura 47: Almacén en el huerto. Fuente: Elaboración propia.**

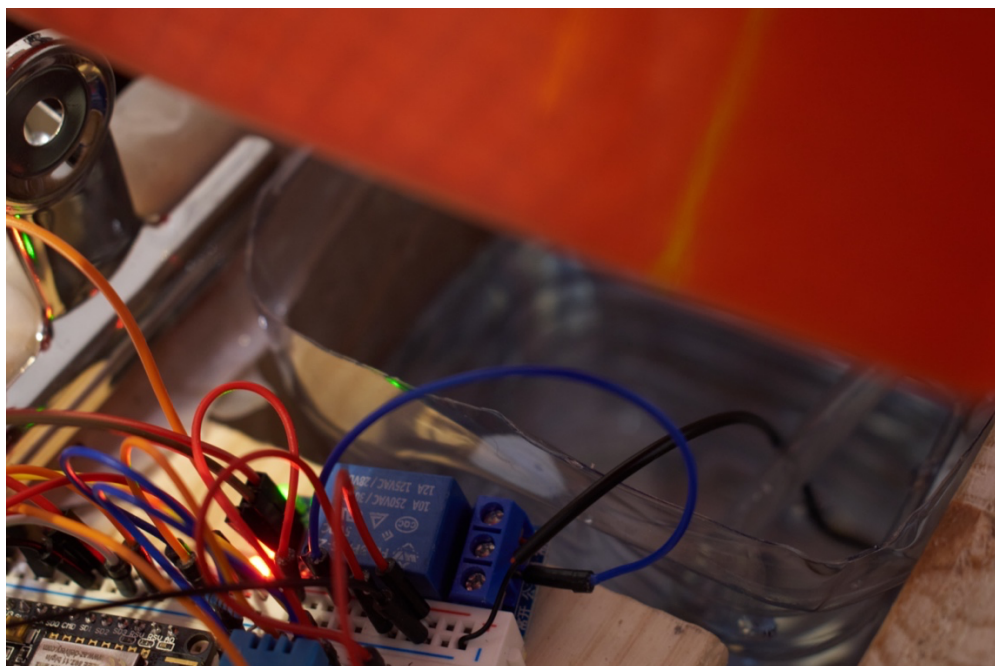
Por último, la placa de Arduino y la protoboard con los correspondientes sensores conectados.

El sistema de Arduino consta de los sensores introducidos en la prueba de concepto (temperatura, humedad, luminosidad y WIFI) más el sensor de humedad de tierra, sensor de flujo de agua, bomba de agua y el relé de la bomba de agua. El único elemento que se ha retirado respecto a la prueba de concepto es el led azul. El resultado final es el siguiente.



**Figura 48: Electrónica final huerto. Fuente: Elaboración propia.**

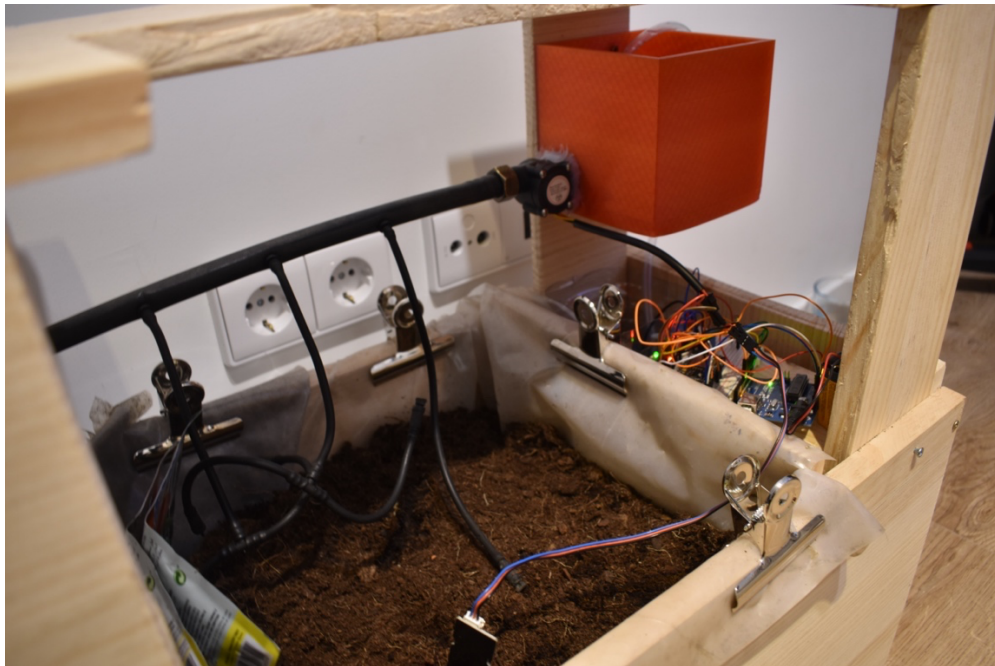
Como se puede comprobar en la anterior figura, ya se ha retirado la tabla de madera que estaba debajo del tanque hecho con impresión 3D, y se ha insertado un tanque de agua con una botella de agua. Dentro de dicha botella de agua se encuentra el motor que va conectado al relé para poder activarse cuando se necesite. La siguiente figura muestra con más detalle estos sensores.



**Figura 49: Relé y tanque de agua. Fuente: Elaboración propia.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

El huerto final con todos los elementos incorporados es el siguiente.



**Figura 50: Resultado final. Fuente: Elaboración propia.**

En cuanto a la plataforma IoT Blynk, se han añadido dos paneles más de texto, para mostrar la humedad de la tierra y el caudal del agua, en los canales V7 y V8 respectivamente.



**Figura 51: Datos Blynk final. Fuente: Elaboración propia mediante Blynk.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Los scripts modificados con el código final se encuentran en el Anexo A. El script para la placa de Arduino en el 9, y el script para la placa NodeMCU ESP8266 en el 10.

#### 5.1.2.3. Tutorial para la fabricación de un huerto inteligente

En el siguiente tutorial se van a especificar los pasos a seguir para la realización de un huerto inteligente automatizado y autónomo. El tutorial que se ha seguido puede ser más económico, por lo que en el desarrollo de este se especificarán cuales componentes son necesarios y cuáles opcionales. En primer lugar, los componentes que se necesitan para realizar este huerto son:

**Tabla 1: Elementos necesarios huerto. Fuente: Elaboración propia.**

Nombre del componente	Función	Obligatorio	Precio
Kit básico Arduino	Placa de Arduino, protoboard, cable de conexión usb a pc, cables para conectar sensores a protoboard, variedad de resistencias y adaptador pila a Arduino	X	12€
Kit básico riego	Sensor de humedad a tierra, mini bomba de agua, canal 5V relé módulo, tubería de agua	X	1,77€
Sensor DHT11	Sensor encargado de medir la temperatura y la humedad	X	0,79€
Fotorresistencia LDR	Sensor encargado de medir la luz		0,05€
Sensor de flujo de agua	Sensor encargado de medir el caudal de agua		4€
PLA impresión 3D	Material utilizado para la impresión 3D		9€
NodeMCU ESP8266	Placa para la conexión WIFI a internet	X	2,13€

Botella de agua	Utilizada como tanque para el agua	X	0,20€
Tablas de madera	Utilizadas para formar la estructura del huerto	X	30€
Pila de 9V	Utilizada para suministrar la corriente a la placa de Arduino y a la placa NodeMCU ESP8266	X	0,24€
Plataforma IoT Blynk	Plataforma utilizada para el envío de información de los sensores y para la interacción con la bomba de agua	X	0€
Otros	Plástico para cubrir la madera, tierra, semillas y clips para sujetar el plástico	X	10€

Los links de compra de los materiales se encuentran en el Anexo B.

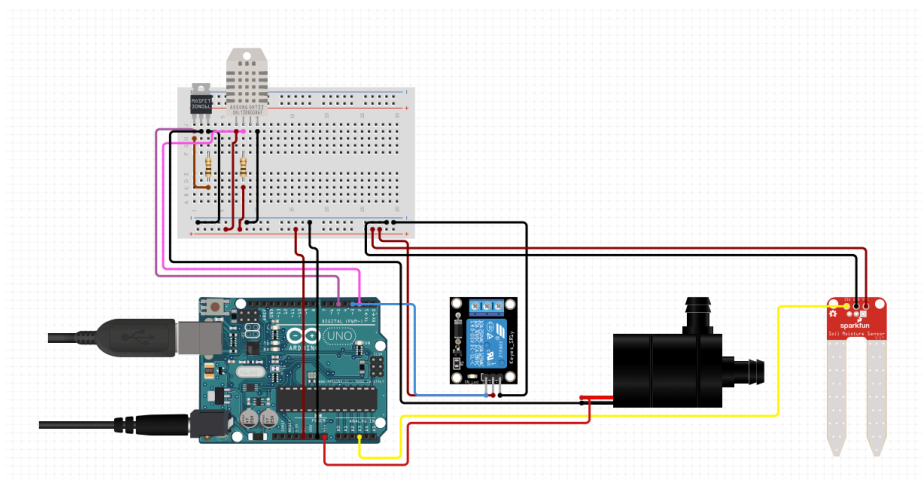
El coste total de los elementos obligatorios para el desarrollo del huerto es de:

$12+1,77+0,79+2,13+0,2+30+0,24+10 = 57,13€$ .

Una vez que se tengan todos los elementos necesarios, se comienza con el desarrollo de los componentes.

En primer lugar, la construcción del sistema electrónico con las diferentes placas y sensores.

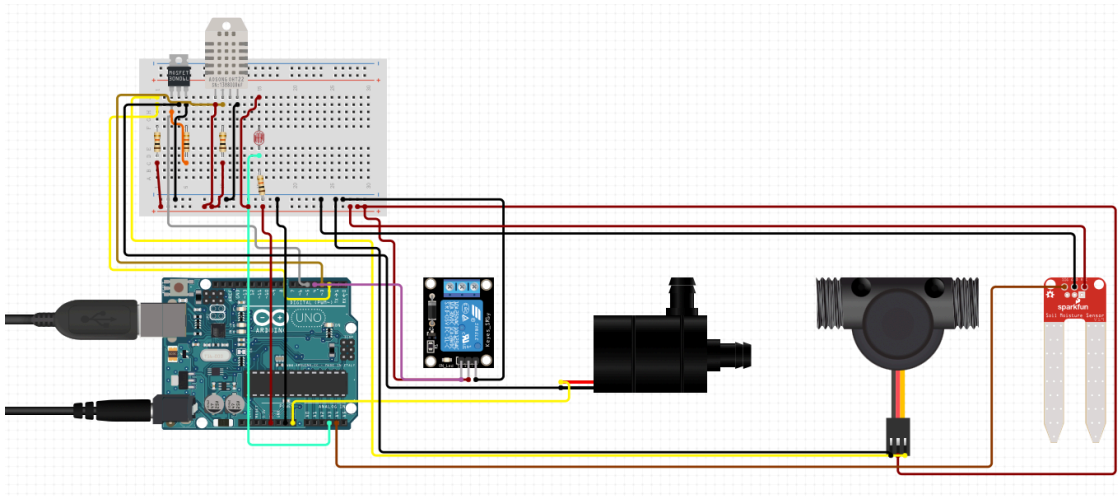
Los sensores conectados a la placa de Arduino básicos se distribuyen de la siguiente manera.



**Figura 52: Estructura sensores básicos. Fuente: Elaboración propia mediante circuito.io.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

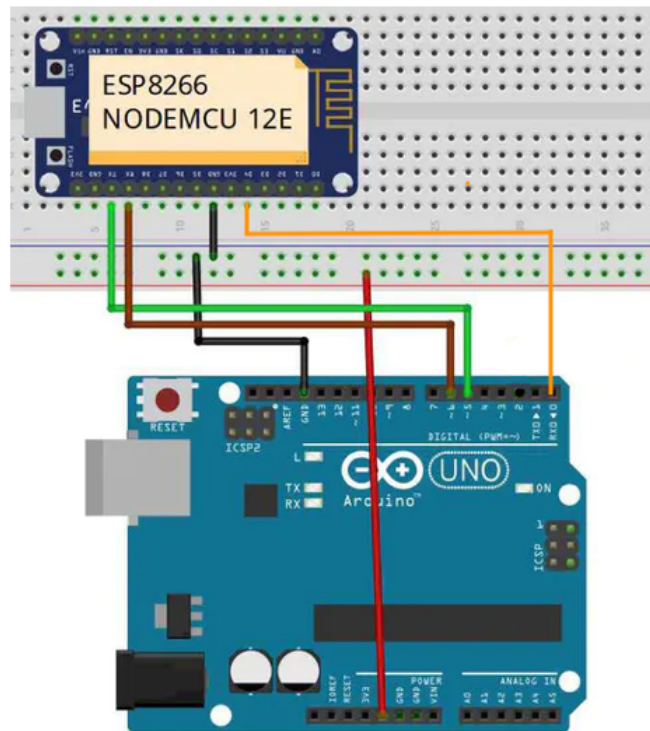
En el caso de utilizar todos los sensores mencionados, incluyendo los que no son obligatorios, el esquema sería el siguiente.



**Figura 53: Estructura sensores utilizados. Fuente: Elaboración propia mediante circuito.io.**

En la anterior figura se puede comprobar como se han añadido el sensor de fotorresistencia LDR y el sensor de flujo de agua. Estos sensores aportarán más información en la toma de decisiones, pero no son estrictamente necesarios.

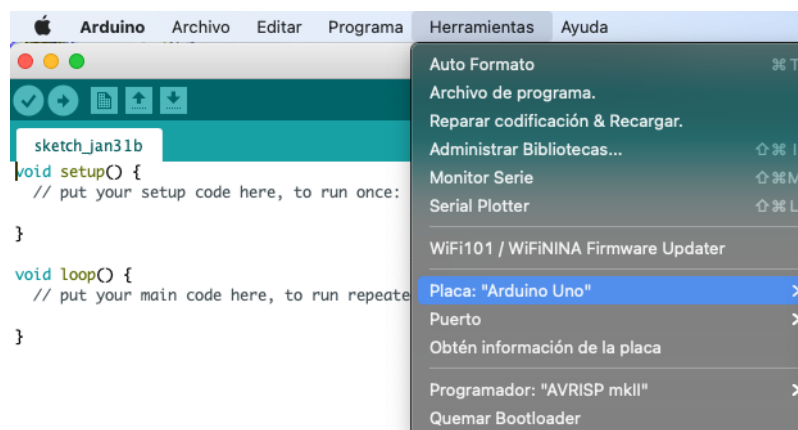
Una vez que los sensores estén conectados, se pasa a la configuración entre la placa de Arduino y la placa WIFI NodeMCU ESP8266. El esquema de conexión de cables es el siguiente.



**Figura 54: Conexión Arduino a NodeMCU8266Fuente: Elaboración propia a partir de (DIY, 2020).**

Una vez se hayan conectado todos los sensores en la placa de Arduino, como la conexión entre NodeMCU ESP8266 y Arduino, se pasa a la configuración por código de las placas. Para esto, hay que descargar el IDE de Arduino desde su página oficial: <https://www.Arduino.cc/en/software>

Una vez descargado el IDE de Arduino, se pasa a la configuración del código. En primer lugar, se realizará el código de Arduino, para esto hay que conectar la placa de Arduino con el cable al usb del ordenador. Una vez conectado, en el IDE de Arduino, en el apartado Herramientas se selecciona la placa Arduino Uno como se ve en la siguiente imagen.



**Figura 55: Seleccionar placa Arduino. Fuente: Elaboración propia mediante IDE de Arduino.**

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

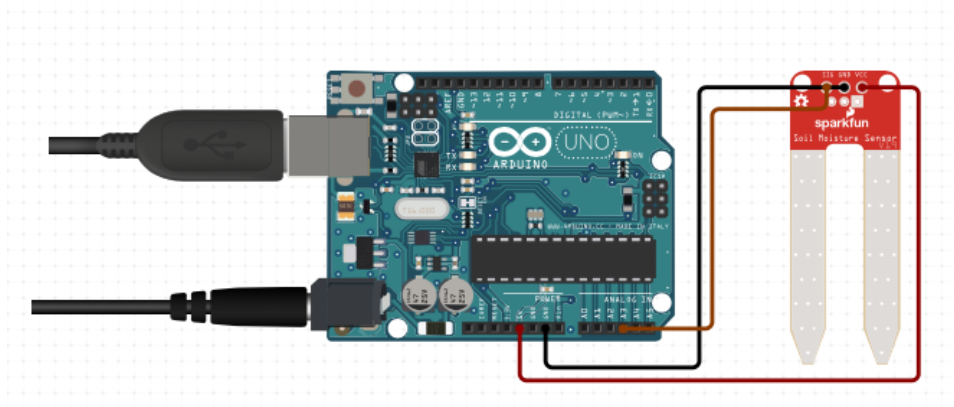
En el apartado Puerto se selecciona el puerto usb donde este conectado el usb de Arduino (suele ser COM3, COM4 o COM7, y entre paréntesis poner Arduino uno). Tras esto, en el mismo apartado de Herramientas, seleccionar la opción Administrar Bibliotecas... Ya que se necesitan instalar algunas para el funcionamiento del código. Estas bibliotecas son

- DHT sensor library / by Adafruit
- Blynk / by Volodymyr Shymanskyi

Se busca el nombre de las librerías en la barra de buscador, y se selecciona la que tenga el mismo autor que en las anteriores. Además, se tiene que introducir la siguiente frase en Arduino/Preferencias/Gestor de URLs Adicionales de Tarjetas:

[http://Arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://Arduino.esp8266.com/stable/package_esp8266com_index.json)

Después, se pulsa el botón instalar y se espera a que se termine la instalación. Una vez que se han instalado las librerías, se pasa a la configuración del sensor de humedad de tierra. Este sensor necesita de unos valores máximos y mínimos, que se consiguen sometiendo a este sensor a condiciones donde no hay humedad, y donde la humedad esta al máximo (es decir, dentro de un vaso de agua). Para comprobar estos valores máximos y mínimos, se tiene que conectar el sensor de humedad de tierra a la placa de Arduino por separado. Se pueden desconectar los cables que se habían configurado a la protoboard anteriormente, y se colocan los cables de la siguiente manera.

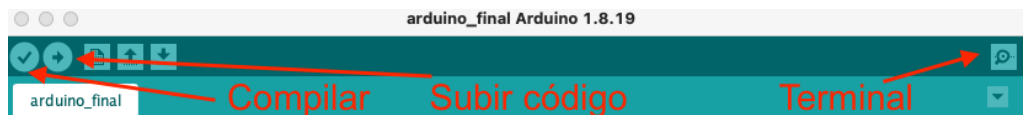


**Figura 56: Arduino y sensor de humedad de tierra. Fuente: Elaboración propia mediante circuito.io.**

Habiendo conectado este sensor, se copia y se pega el código que aparece en el Anexo A – 11 en el IDE de Arduino.

Una vez que este copiado el código este se compila para comprobar que no haya errores.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.




**Figura 57: Botones en IDE Arduino. Fuente: Elaboración propia mediante IDE de Arduino.**

Cuando se haya terminado de compilar el código, se pulsa el botón para subir código, y una vez hay terminado, se abre la terminal. En la terminal se verán los datos que manda el sensor. En primer lugar, se deja el sensor sin estar en contacto con nada. Se observan los valores que se muestran en la terminal y se apunta el valor más bajo, ya que este será la referencia que se tenga para determinar que hay un 0% de humedad. Después, se sumerge el sensor en un vaso de agua, y después de esperar un rato, se recoge el valor más alto que haya salido, este será la referencia de un 100% de humedad.

Tras esto, se vuelven a conectar los cables tal y como estaban antes, y se copia y pega el código del Anexo A – 12.

En las líneas 12 y 13 se pueden ver las variables `AireValue` y `WaterValue`, el valor que hay apuntado es uno de referencia, pero estos datos se tienen que modificar con los que se han obtenido anteriormente. Para esto, el número de `AireValue` se modifica por el que se había obtenido con teniendo el sensor sin contacto con nada, y en `WaterValue`, se modifica por el que se había obtenido al sumergir el sensor en un vaso de agua.

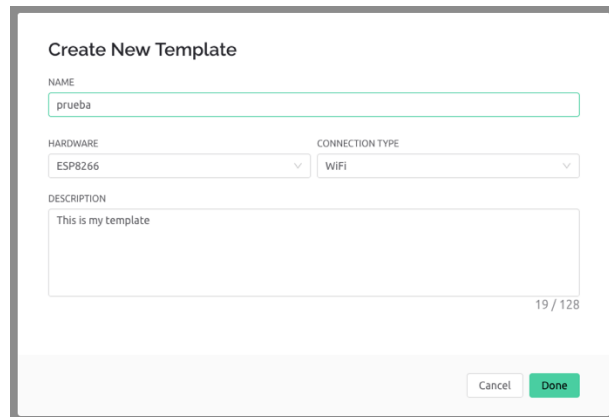
En el caso de que se estén utilizando solo los sensores que son obligatorios, hay que eliminar las siguientes líneas del anterior script: 5, 6, 16 - 26, 30, 32, 34, 49, 52, 55, 76 – 82. Estas líneas son las que hacen mención a los sensores que no son obligatorios. Cuando hay un – entre dos números, es que hay que eliminar todas las líneas que hay entre medias, incluyendo ambos números.

Tras esto, se compila de nuevo el archivo con el botón  y cuando termine, se vuelve a subir el proyecto. Cuando se haya terminado de compilar, antes de subir el código, se tiene que desconectar el pin que conecta con la placa NodeMCU ESP8266, y volverse a conectar una vez que el código haya terminado de subirse. El cable es el que se encuentra situado en el pin 0 de Arduino (Rx) y en el pin 4 de la placa WIFI. Cuando se haya terminado de subir el código, se vuelve a conectar el cable y se abre la terminal. Se verá en pantalla los valores que mandan los sensores conectados a Arduino.

Una vez que se ha terminado la configuración de la placa de Arduino, hay que registrarse en la plataforma IoT Blynk. Para esto se accede al siguiente enlace: <https://blynk.cloud/dashboard/register>

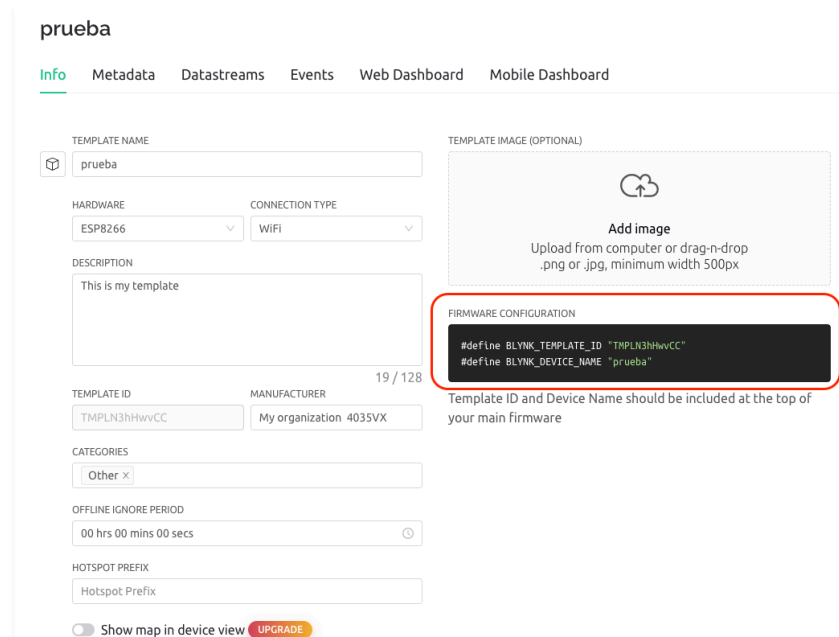
IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Una vez terminado el registro, se crea un nuevo template con el hardware ESP8266 y el tipo de conexión mediante WIFI.



**Figura 58: Nuevo template en Blynk. Fuente: Elaboración propia mediante Blynk.**

Dentro del template, en el apartado info, hay un código que necesitaremos en el siguiente paso.

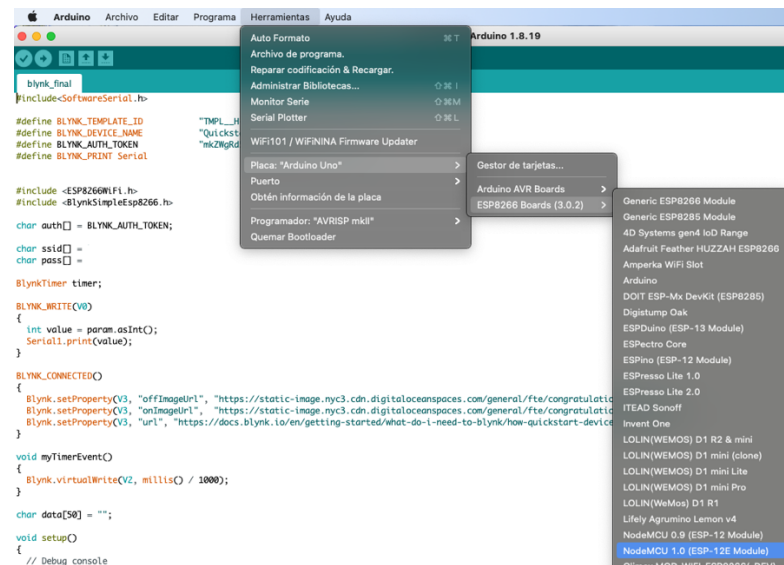


**Figura 59: Código para conectar NodeMCU ESP8266 con Blynk. Fuente: Elaboración propia mediante Blynk.**

Para la configuración de la placa WIFI se necesita un cable micro-sub para conectarlo al ordenador. Para configurar esta placa no es necesario desconectar la placa de Arduino. Una vez conectada la placa WIFI al ordenador mediante el cable, en el apartado Herramientas/Placas se busca la placa ESP8266 NodeMCU y en Herramientas/Puerto habrá aparecido un nuevo puerto, se selecciona el nuevo. La ventana del terminal se pondrá blanca

IoT aplicado al ámbito de la agricultura inteligente en el hogar.


y ya no se mostrarán los datos de la placa de Arduino, ya que ahora se está mostrando la salida de la placa WIFI pero todavía no esta configurada.



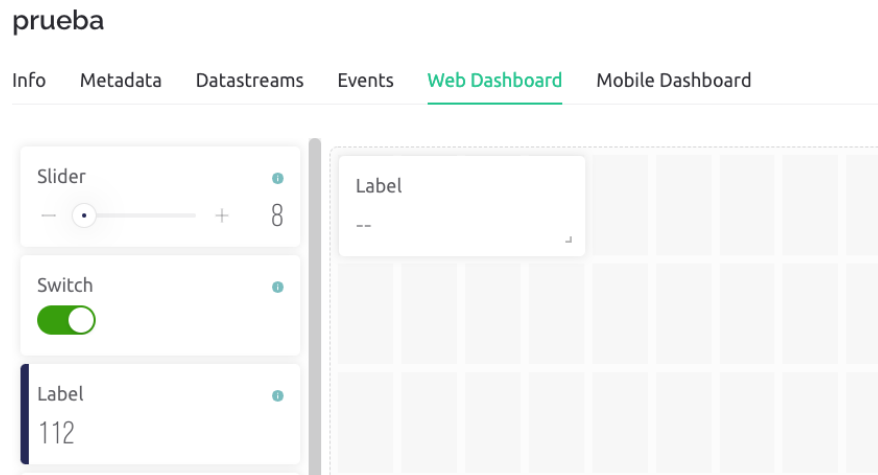
**Figura 60: Placa NodeMCU ESP8266 en herramientas. Fuente: Elaboración propia mediante IDE de Arduino.**

Se abre una nueva pestaña de Arduino pulsando en Archivo/Nuevo y se escribe el código del anexo A – 13.

Este código necesita varias modificaciones, la primera es el código para conectarse con Blynk. Para esto, se copia el código que se había mencionado anteriormente y se sustituye por el que aparece en este código en las líneas 2 y 3. La segunda modificación es el nombre y contraseña del router al que se va a conectar, que se tiene que indicar en las líneas 8 y 9. La última modificación es para eliminar las líneas que hacen referencia a los sensores que no son obligatorios, en el caso de que no se hayan utilizado. Si no se han usado los sensores de fotorresistencia LDR y el sensor de flujo de agua, hay que modificar la línea 50 por: `humGND = String(data).substring(16,19);` Y además, eliminar las siguientes líneas: 37, 39, 49, 51, 54, 56, 59, 61.

Después de las modificaciones, al igual que en el caso de Arduino, se compila el código con el botón . Una vez compilado se desconectan los cables que se comunican con la placa de Arduino (Rx y Tx), que son los cables que están en los pines 5 y 6 de la placa de Arduino, y se sube el código. Una vez que el código se ha subido, se vuelven a conectar los cables y se abre la terminal. Se esperan unos segundos a que la conexión se realice y se comprueba como se reciben los datos mandados por la placa de Arduino. Solo falta configurar el dashboard en la plataforma Blynk. Para esto, en el template que se ha creado en Blynk, se accede en el apartado Web Dashboard y se crean varios widgets de texto (label).

IoT aplicado al ámbito de la agricultura inteligente en el hogar.



**Figura 61: Configurar labels en Blynk. Fuente: Elaboración propia mediante Blynk.**

Se arrastra el widget que se quiere crear a la parte de la derecha. Una vez creado un widget, se pone el ratón encima y se pulsa la rueda de configuración. Se tiene que rellenar el título y el DataStream. El DataStream hay que configurar uno nuevo para cada widget y siempre es un pin virtual. Se crean los siguientes widgets con los siguientes datos: Title / Canal Pin Virtual /Data Type

- Temperatura / V4 / String
- Humedad / V5 / String
- Luminosidad / V6 / String
- Humedad de tierra / V7 / String
- Flujo de agua / V8 / String


Un ejemplo de como se crearía el de temperatura

**Label Settings**

TITLE  
Temperatura

Datastream

Virtual Pin Datastream

NAME	ALIAS
 Temperatura	Temperatura

PIN	DATA TYPE
V4	String

DEFAULT VALUE  
Default Value


+ ADVANCED SETTINGS

Cancel Create

**Figura 62: Widget en Blynk. Fuente: Elaboración propia mediante Blynk.**

Tras esto, se configura un widget Switch, con el título Riego, en el pin virtual V0 y el tipo de dato Integer. El resultado final debería de ser algo parecido a la siguiente imagen.

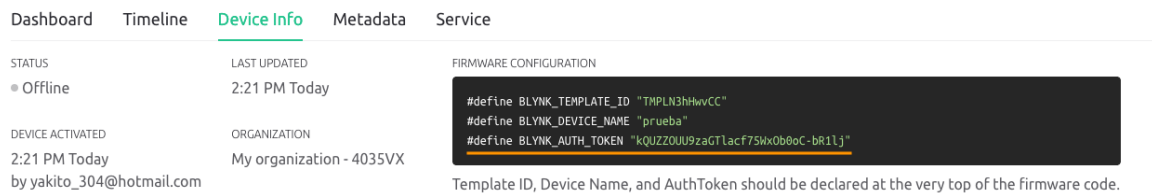
Events **Web Dashboard** Mobile Dashboard

temperatura (V4) String	humedad (V5) String	luminosidad (V6) String
Humedad de tierra (V7) String	Flujo de agua (V8) String	Riego (V0)  OFF


**Figura 63: Dashboard final Blynk. Fuente: Elaboración propia mediante Blynk.**

Tras esto, se pulsa el botón de guardar arriba a la derecha, se selecciona la primera opción en la ventana que sale (Update 1 device), se pulsa continuar, y en el menú de opciones de la izquierda, se selecciona la lupa y se pulsa el botón New Device o Nuevo Dispositivo. Este se va a configurar desde el Template, el cuál se acaba de crear. Se selecciona el template que se ha creado y se le da un nombre al dispositivo. Una vez creado, aparecerá el dashboard que se ha configurado antes con los valores de los sensores. En caso de que no se muestren los datos pasados unos segundos, en el apartado Device Info aparece el token de autorización que habría que añadir al código del script. En

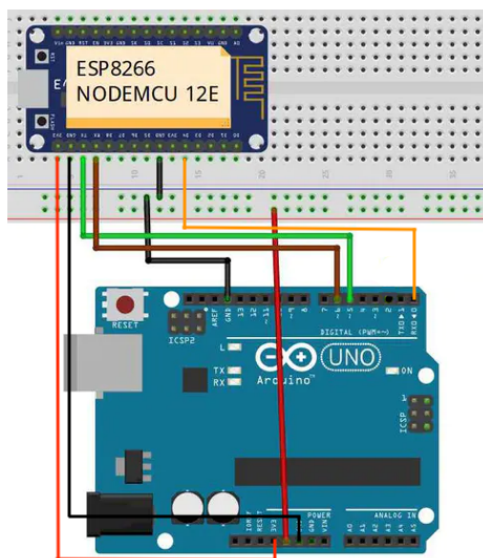
IoT aplicado al ámbito de la agricultura inteligente en el hogar.



**Figura 64: Token de autorización de Blynk. Fuente: Elaboración propia mediante Blynk.**

Se copia y se pega este token en el script del código del IDE de Arduino, justo en la línea siguiente a donde se había puesto anteriormente el template id y el device name (líneas 2 y 3, incluir el token en la línea 4). Después de pegarlo, volver a compilar el código con el botón , desconectar el cable de los pines 5 y 6 de Arduino, subir el código, y una vez subido, volver a conectar los cables.

Tras esto, se desconectan los dos cables usbs de las placas de Arduino y de NodeMCU ESP8266, y se conectan dos últimos cables entre la placa de Arduino y NodeMCU.



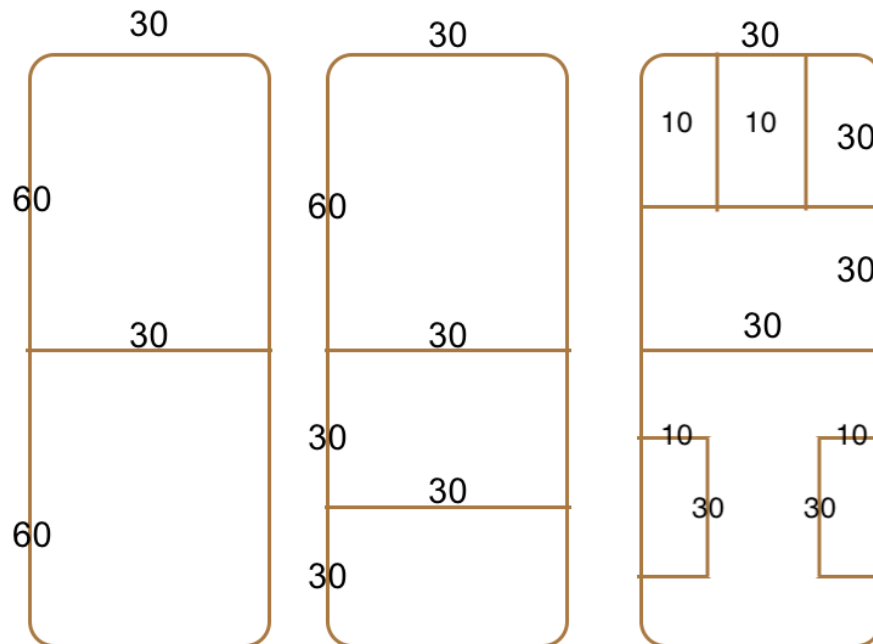
**Figura 65: Esquema final. Fuente: Elaboración propia a partir de (DIY, 2020).**

Estos cables dan corriente desde la placa de Arduino hasta la placa de NodeMCU ESP8266. Muy importante que la corriente sea de 3,3V, si no la placa WIFI se podría quemar. Tras esto, ya se puede conectar la pila de 9V al adaptador de Jack, y conectarlo a la placa de Arduino. El sistema se conecta automáticamente al router y a la plataforma IoT.

La plataforma IoT Blynk también dispone de una aplicación para dispositivos móviles si se quiere controlar desde ahí. Para ello, descargar la app de la store e iniciar sesión con los datos con los que se había registrado en la plataforma.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Tras tener el sistema electrónico configurado, solo falta la fabricación del huerto. Para esto, las dimensiones del huerto son variables según lo que se quiere cultivar. En este caso de prueba se van a cultivar cebollino y rabano que necesitan una profundidad de 20cm, por lo que el huerto no será muy profundo.



**Figura 66: Cortes de las tablas de madera. Fuente: Elaboración propia.**

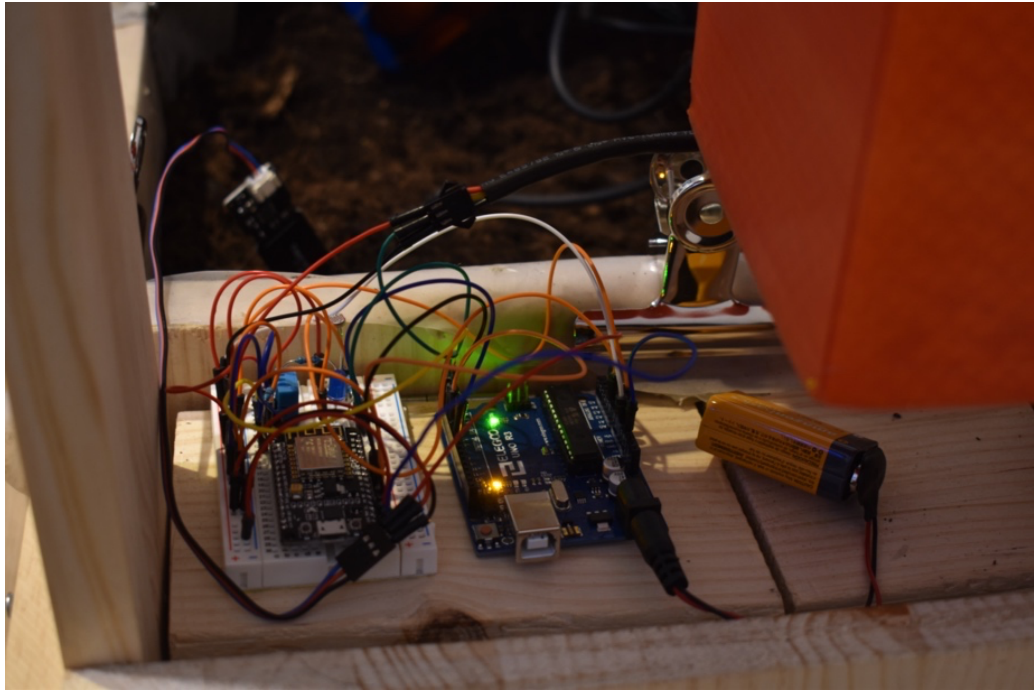
En este caso, la madera se ha cortado según la figura anterior. La primera tabla se ha utilizado para hacer los dos lados de 60X30 cms. La segunda tabla, se ha utilizado la mitad para la base de 60X30 cms, y la otra mitad partida en dos para los dos lados cortos de 30X30 cms. La tercera tabla se ha utilizado para el techo con forma de H, 5 tablillas de 30X10 cms y una tabla de 30X30 cms. Esta última tabla se usa para separar el huerto de una zona de almacenaje, ubicación del tanque de agua, y la zona donde estarán los circuitos eléctricos. Las tablillas de 30X10 cms, se van a utilizar 4 de ellas para las patas con las que sujetar el techo, y la que sobra, para hacer una pequeña puerta a la zona de almacenaje. Una vez cortada y atornillada la madera, se ha puesto un plástico en la zona del huerto, y encima se ha echado la tierra necesaria. Además, del tanque de agua se ha realizado una salida mediante unas tuberías para distribuir el agua por el huerto. El resultado final, a falta de colocar el sistema eléctrico es el siguiente.



**Figura 67: Estructura final del huerto. Fuente: Elaboración propia.**

El sistema de tuberías es opcional, ya que con la manguera que tiene el kit básico de riego para Arduino, se le pueden hacer unos agujeros pequeños para que salga el agua a su paso, y con eso es suficiente para regar la tierra.

El sistema eléctrico se tiene que colocar en un lugar que tenga las mismas condiciones que el huerto, es decir misma luz, temperatura, humedad, etc. Lo mejor es dejarlo al lado, en el caso de este huerto se va a colocar en el lado derecho, al lado del tanque de agua. El único sensor que hay que tener en cuenta es el de humedad de tierra, que tiene que ir dentro de la tierra.



**Figura 68: Sistema eléctrico conectado. Fuente: Elaboración propia.**

Hay que estar pendiente de rellenar el tanque de agua cuando se agote, y procurar poner el huerto cerca de alguna ventana para que le de la luz necesaria para crecer. En caso de que se quiere regar de forma manual el huerto, se hace mediante la plataforma Blynk pulsando el botón de riego, si no, se regara de forma automática.

Posibles errores que puedan surgir sobre la elaboración del huerto y como solucionarlos

- Abrir Arduino y que se quede la pantalla gris y sin poder escribir
  - Abrir preferencias de Arduino, y desactivar la opción “Usar editor externo”.
- No se puede conectar a la red WIFI
  - Esto puede ocurrir debido a que alguna cadena de texto no la reconozca como texto. Por ejemplo, el símbolo \ es un símbolo reservado, y lo que vaya a continuación no lo reconoce como texto, de tal forma que si la contraseña es “Prue\ba123” no reconocerá la contraseña porque después de la barra invertida esta esperando algún carácter reservado. En este caso, habría que escribir la contraseña de la siguiente manera “Prue\\ba123”. Al poner la doble barra se entiende que no hay elemento reservado, y aunque haya doble barra invertida, se entiende como si fuese una.
- Al compilar el código sale un error por pantalla
  - Comprobar que todos los cables estén correctamente conectados.
  - En el caso de utilizar solo los sensores obligatorios, comprobar que se han eliminado y modificado las líneas correspondientes en los scripts

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

- Se han copiado las dobles comillas como curvadas en vez de comillas dobles rectas. “” estás dobles comillas dan error, "" estas no.
- Al subir el código sale un error por la pantalla
  - Comprobar si se han desconectado los cables para la comunicación serial antes de subir el código. Pines 5 y 6 en la placa de Arduino cuando se suba el código a la placa WIFI, y el pin 0 de la placa de Arduino cuando se suba el código a la placa de Arduino.

#### 5.1.2.4. Problemas encontrados y como solucionarlos

Durante el desarrollo del proyecto se han encontrado diferentes problemas que no se habían pensado durante la fase de investigación y que se ha tenido que reaccionar de diversas formas.

- La pila de petaca que se quería utilizar era una pila de 4,5V. Al conectar la placa de Arduino con cables desde la pila de petaca se veía un led de color naranja que parpadeaba. Esto ocurre porque el voltaje que se esta recibiendo es bajo, y puede dejar de funcionar. Además, la conexión con cables era bastante inestable. Por esto se decidió buscar otra solución donde la pila tuviese más voltaje y la conexión de la alimentación a la placa fuese más estable.
- Con la placa solar ha ocurrido el mismo problema que con la pila de 4,5V. La salida de corriente que daba la placa solar es de 5V, lo cual esta por debajo de lo recomendado que es entre 7V y 12V.
- Pantalla RGB. No se ha considerado necesaria teniendo una aplicación en el ordenador y en el móvil. Al tener un componente más conectado haría que la batería se acabase antes y además elevaría el presupuesto del desarrollo para una función que ya realiza la plataforma IoT.
- Para poder hacer el tutorial se ha utilizado la misma protoboard que viene en el kit de Arduino, aunque se tuviese una más grande. El espacio es muy escaso y los elementos muy compactos, sobre todo por la placa NodeMCU ESP8266 que ocupa mucho.
- El sensor de flujo de agua se pensaba que también servía como una válvula que abriese o cerrase el paso del agua. Debido a esto el tanque de agua se ha puesto en altura, para que el agua bajase solo por la presión de estar en alto. Al configurar el sensor y comprobar que solo mostraba el valor del caudal del agua, se han tenido que adquirir 2 sensores más, la bomba de agua y el relé para activarla. El sensor del caudal

no es necesario, pero aporta la información de la cantidad de agua que ha salido, lo cual se puede utilizar para saber cuanto se quiere regar.

#### **5.1.3. SPRINT 4: Pruebas**

El sprint final consiste en realizar pruebas finales para comprobar que todo funciona correctamente.

En la figura 48, adjuntada en el apartado 4.1.3.2 del desarrollo del sprint 3 se puede comprobar el resultado final del huerto inteligente y en la figura 51 del mismo apartado, la conexión con la plataforma IoT. Las pruebas se realizaron durante el proceso de desarrollo, una vez terminado, no se han encontrado más problemas.

## 6. Conclusiones y trabajos futuros

Este trabajo tenía como objetivo principal diseñar e implementar un huerto inteligente autónomo de bajo coste conectado a una aplicación IoT. Gracias al desarrollo de la solución llevada a cabo durante el sprint 3, redactada en el apartado 4.1.3 ese objetivo queda resuelto.

Las principales contribuciones obtenidos tras la realización de este proyecto son las siguientes

- Diseño e implementación. En el apartado 4.1.3.2
- Bajo coste. Presupuesto inicial de 70€ (apartado 3.3) y coste final de 57,13€ (apartado 4.1.3.3)
- Conexión a la aplicación IoT. Conexión con Blynk en el apartado 4.1.2.2.4

Teniendo en cuenta estos resultados, queda demostrado en el desarrollo del TFM que han sido alcanzados los objetivos específicos planteados ya que:

1. Estudiar y analizar las diferentes tecnologías relativas a la Industria 4.0. El análisis de estas tecnologías se ha llevado a cabo en el apartado 2.3, y en el 2.4 se ha descrito la conclusión de dicho estudio.
2. Diseñar un prototipo de huerto inteligente a bajo coste. En el apartado 4.1.3.3 se ha demostrado que el coste final del producto es de 57,13 €, en comparación con huertos inteligentes comerciales como ClickAndGrow que tienen un coste 230€, se considera que se ha complicado el objetivo de diseñar un prototipo de bajo coste.
3. Fabricar un producto final, útil y tangible. Este objetivo se considera alcanzado ya que el huerto inteligente final (figuras 46 y 50) es tangible y puede ser utilizado para el consumo propio, por lo que es útil.
4. Realizar un tutorial para la implementación del huerto. El tutorial se ha realizado en el apartado 4.1.3.3 en el cuál se hace un resumen de los pasos a seguir, mostrando los esquemas finales de conexión de sensores, y el código final que se ha utilizado en cada placa.
5. Aprender sobre el desarrollo de un sistema con varios habilitadores digitales conectados. Gracias a este trabajo se han aprendido como utilizar e implementar diferentes habilitadores digitales de la Industria 4.0 y como se conectan entre sí. Los habilitadores digitales han sido: la robótica por parte de los microcontroladores de Arduino y NodeMCU, los sensores de temperatura, humedad de tierra, flujo de agua, luminosidad y humedad, la impresión 3D gracias a la construcción del tanque de agua y la una plataforma IoT gracias a Blynk.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Por consiguiente, se puede concluir que el objetivo principal, diseñar e implementar un huerto inteligente autónomo de bajo coste conectado a una aplicación IoT, ha sido alcanzado tras la realización de este TFM.

#### **6.1.1. “Líneas de trabajo futuras”**

Los dos habilitadores digitales que han faltado por añadir al proyecto del huerto inteligente han sido el big data y la ciberseguridad. Ambos habilitadores pueden agregar un gran potencial al proyecto.

##### **6.1.1.1. Big data**

A través de una plataforma Cloud como Azure se podría crear una base de datos donde almacenar los valores que mandan los sensores, y cuando se activa el sistema de riego. En el momento que haya una gran cantidad de datos, se podría hacer un análisis de cuando se activa para sacar conclusiones e incluso se podría agregar algún mecanismo de inteligencia artificial con machine learning para añadir esas optimizaciones de forma automática a los microcontroladores, y que el sistema de riego se active en función de dichos análisis.

El almacenamiento de grandes datos no solo sería importante para la toma de decisiones en el sistema de riego, si no que también se podría almacenar en la base de datos la planta que se esta cultivando y hacer otros tipos de análisis. Como cuales son las plantas que más se cultivan, cuanto riego necesitan, cuanto tiempo tardan en salir con este sistema, etc.

##### **6.1.1.2. Ciberseguridad**

El envío de los valores de los sensores en este proyecto se ha transformado varias veces por lo que en algún caso se puede mostrar algún dato que no es real, por no decir que todos los valores están redondeados. La “blockchain” es un sistema seguro e inmutable, por lo que aporta una capa extra de seguridad a este sistema, pudiendo leer los valores originales sin ningún tipo de modificación. IOTA es el ecosistema que propone el uso de la “blockchain” para los ecosistemas del internet de las cosas. El token utilizado es MIOTA, pero no es necesario su uso ya la “blockchain” de IOTA funciona diferente al resto de blockchains. Las blockchains comunes, como Bitcoin o Ethereum, utilizan esta cadena de bloques para mantener publico el libro mayor distribuido, en cambio en IOTA funciona con DAG (Grafo acíclico dirigido) que en lugar de unir bloque a bloque en una “línea infinita”, verifica el estado del libro mayor mediante un grafo.

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

Al publicar los datos que se han obtenido de los sensores en el DAG de IOTA, se podrían observar los valores de los sensores desde cualquier parte del mundo, sin necesidad de estar conectado a la misma red local, y estando seguro de que esos valores no han sido modificados.

## Referencias bibliográficas

- Acosta, A., Aguilar, A., & Pinzón Trejos, C. (29 de Julio de 2015). *Automatización de bajo costo utilizada en la producción agrícola en invernaderos y huertos caseros*. Panamá: Universidad Tecnológica de Panamá.
- Ata Jaber, A. (14 de Julio de 2017). *Diseño de un huerto inteligente destinado al autoconsumo*. Cataluña, España: Universitat Politècnica de Catalunya.
- Biohort GmbH. (2021). *Biohort*. Obtenido el 15 de Octubre de 2021, de Huerto urbano: [https://www.biohort.com/shop/es/huerto-urbano.html?product\\_id=840&noredirect=1&gclid=CjwKCAiA4veMBhAMEiwAU4XRrm8X48rOOC7ggxgceqvCUccV5JsPtVGECCvqYPVK29FOVKYeH0iqhoCSesQAvD\\_BwE](https://www.biohort.com/shop/es/huerto-urbano.html?product_id=840&noredirect=1&gclid=CjwKCAiA4veMBhAMEiwAU4XRrm8X48rOOC7ggxgceqvCUccV5JsPtVGECCvqYPVK29FOVKYeH0iqhoCSesQAvD_BwE)
- blumfeldt. (2021). *Urban Bamboo Jardín hidropónico*. Obtenido el 14 de Octubre de 2021, de [https://www.blumfeldt.es/Accesorios-de-jardin/Invernaderos-pequenos/Urban-Bamboo-Jardin-hidroponico-12-plantas-24W-LED-7-litros-Bambu.html?gclid=CjwKCAiA4veMBhAMEiwAU4XRrxjPS1QGqsu2pHY2cKyO4TmLgMaYinFlvL3HeOoGtNZO74I9wDqidRoCOXwQAvD\\_BwE](https://www.blumfeldt.es/Accesorios-de-jardin/Invernaderos-pequenos/Urban-Bamboo-Jardin-hidroponico-12-plantas-24W-LED-7-litros-Bambu.html?gclid=CjwKCAiA4veMBhAMEiwAU4XRrxjPS1QGqsu2pHY2cKyO4TmLgMaYinFlvL3HeOoGtNZO74I9wDqidRoCOXwQAvD_BwE)
- Click & Grow. (2021). *Click & Grow*. Obtenido el 14 de Octubre de 2021, de El Smart Garden 9: <https://eu.clickandgrow.com/products/the-smart-garden-9>
- Click & Grow. (2021). *Click & Grow*. Obtenido el 14 de Octubre de 2021, de The Wall Farm Indoor Vertical Garden: <https://eu.clickandgrow.com/products/wall-farm-indoor-vertical-garden>
- Cortina, M. (2019). *Arduino*. Obtenido de TerraSmart - La agricultura del futuro: <https://www.arduino.cc/pro/case-studies/terrasmart>
- Crespo, E. (12 de 09 de 2017). *Aprendiendo Arduino*. Obtenido de ESP8266 y Arduino: <https://aprendiendoarduino.wordpress.com/2017/09/12/esp8266-y-arduino/>
- Descubre arduino. (03 de 07 de 2021). *Descubre arduino*. Obtenido de ESP32 vs ESP8266 ¿Cuales son las diferencias entre ambos módulos?: <https://descubrearduino.com/esp32-vs-esp8266/>

- DIY, R. (05 de 05 de 2020). *Hackster*. Obtenido de Send Data From Arduino to NodeMCU and NodeMCU to Arduino: <https://www.hackster.io/RoboticaDIY/send-data-from-arduino-to-nodemcu-and-nodemcu-to-arduino-17d47a>
- elifhocarobotik. (17 de Junio de 2021). *hackster.io*. Obtenido de ESP8266 ESP-01 Smart Door / Smart Intercom: <https://www.hackster.io/elifhocarobotik/esp8266-esp-01-smart-door-smart-intercom-05abb8>
- Finio, B. (s.f.). *instructables.com*. Obtenido de Raspberry Pi Controlled Irrigation System: <https://www.instructables.com/Raspberry-Pi-Controlled-Irrigation-System/>
- gradyh. (31 de Agosto de 2015). GradyHillhouseGarduino.
- hackster.io. (21 de Mayo de 2017). *hackster*. Obtenido de Smart Garden: <https://www.hackster.io/patelDeep/smart-garden-cda82f>
- Hydroenv. (s.f.). *Hydroenv*. Obtenido de Guía para el cultivo de rábano: [https://www.hydroenv.com.mx/catalogo/index.php?main\\_page=page&id=406](https://www.hydroenv.com.mx/catalogo/index.php?main_page=page&id=406)
- Icaza Guamba, C. A. (2021). Análisis, diseño y desarrollo de un sistema inteligente y automatizado de monitoreo y control de cultivos con IoT. Caso de estudio: huertos urbanos. Quito: UCE.
- Majofesa maderas. (24 de Febrero de 2020). *Majofesa*. Obtenido de Maderas económicas: cuáles son y dónde encontrarlas: <https://www.majofesa.com/maderas-economicas-cuales-son-y-donde-encontrar-las-mas-baratas/>
- Mundo huerto. (03 de 07 de 2021). *Mundo huerto*. Obtenido de Cuántas fresas da una planta: <https://www.mundohuerto.com/cultivos/cultivo-fresa/cuantas-da-una-planta>
- Potter, C. (27 de Diciembre de 2015). *hackster.io*. Obtenido de The hydroMazing Smart Garden System: <https://www.hackster.io/bitsandbots/the-hydrumazing-smart-garden-system-a802e5>
- Prácticas con Arduino. (s.f.). *Prácticas con Arduino*. Obtenido de Alimentación eléctrica de Arduino: [http://www.practicasonarduino.com/manualrapido/alimentacin\\_elctrica\\_de\\_arduino.html](http://www.practicasonarduino.com/manualrapido/alimentacin_elctrica_de_arduino.html)
- Raihan, A. (20 de Noviembre de 2016). *create.arduino.cc*. Obtenido de Smart Garden: <https://create.arduino.cc/projecthub/smart-boys/smart-garden-2dd7b0>

Runcci-yun. (21 de 11 de 2019). *Amazon*. Obtenido de Sistema de riego automático DIY Kit Arduino: <https://www.amazon.es/RUNCCI-YUN-Sistema-Automático-Arduino-Humedad/dp/B0814HXWVV>

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Arduino: <https://es.wikipedia.org/wiki/Arduino>

## Anexo A. Código implementado

1. Código implementado para el sensor DHT11 en la prueba de concepto.

```
#include <DHT.h>

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
    Serial.begin(9600);
    dht.begin();
}

void loop() {
    delay(5000);

    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Error obteniendo los datos del sensor DHT11");
        return;
    }

    float hif = dht.computeHeatIndex(f, h);
    float hic = dht.computeHeatIndex(t, h, false);

    Serial.print("Humedad: ");
    Serial.print(h);
    Serial.print(" %\t");

    Serial.print("Temperatura: ");
    Serial.print(t);
    Serial.print(" *C ");

    Serial.print(f);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Serial.print(" *F\t");  
Serial.print("Índice de calor: ");  
Serial.print(hic);  
Serial.print(" *C ");  
Serial.print(hif);  
Serial.println(" *F");  
}
```

## 2. Código implementado para el sensor DHT11 y LDR en la prueba de concepto.

```
#include <DHT.h>  
  
#define DHTPIN 2  
#define LDRPin A3  
  
#define DHTTYPE DHT11  
  
DHT dht(DHTPIN, DHTTYPE);  
  
void setup() {  
  Serial.begin(9600);  
  pinMode(LDRPin, INPUT);  
  dht.begin();  
  
}  
  
void loop() {  
  delay(5000);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
float h = dht.readHumidity();  
float t = dht.readTemperature();  
float f = dht.readTemperature(true);  
  
if (isnan(h) || isnan(t) || isnan(f)) {  
    Serial.println("Error obteniendo los datos del sensor DHT11");  
    return;  
}  
  
float hif = dht.computeHeatIndex(f, h);  
float hic = dht.computeHeatIndex(t, h, false);  
  
int LDRvalue = analogRead(LDRPin);  
  
Serial.print("Humedad: ");  
Serial.print(h);  
Serial.print(" %\t");  
Serial.print("Temperatura: ");  
Serial.print(t);  
Serial.print(" *C ");  
Serial.print("Índice de calor: ");  
Serial.print(hic);  
Serial.print(" *C ");  
Serial.print("\t");  
Serial.print("Luminosidad: ");  
Serial.print(LDRvalue);  
Serial.print("\n");
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
}
```

### 3. Código implementado para la placa NodeMCU ESP8266 en la prueba de concepto.

```
#include <ESP8266WiFi.h>

#define WLAN_SSID      "...your SSID..."
#define WLAN_PASS      "...your password..."

WiFiClient client;

void setup() {
  Serial.begin(115200);
  delay(10);

  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);

  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}

void loop() {
}
```

### 4. Código implementado para la placa Arduino para la comunicación en la prueba de concepto

```
#include<SoftwareSerial.h>

SoftwareSerial espSerial(5,6);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
#include <DHT.h>

#define DHTPIN 2

#define LDRPin A0


#define DHTTYPE DHT11


DHT dht(DHTPIN, DHTTYPE);


void setup() {
    Serial.begin(115200);
    espSerial.begin(115200);
    pinMode(LDRPin, INPUT);
    dht.begin();
    delay(2000);
}


void loop() {
    delay(5000);
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float f = dht.readTemperature(true);
    if (isnan(h) || isnan(t) || isnan(f)) {
        Serial.println("Error obteniendo los datos del sensor DHT11");
        return;
    }

    float hif = dht.computeHeatIndex(f, h);
    float hic = dht.computeHeatIndex(t, h, false);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
int LDRvalue = analogRead(LDRPin);

Serial.println("H= " + String(h));
Serial.println("T= " + String(t));
Serial.println("L= " + String(LDRvalue));

espSerial.print("H= " + String(h));
espSerial.print("T= " + String(t));
espSerial.print("L= " + String(LDRvalue));
delay(5000);
}
```

5. Código implementado para la placa NodeMCU ESP8266 para la comunicación en la prueba de concepto

```
#include <ESP8266WiFi.h>

#define WLAN_SSID      "...your SSID..."
#define WLAN_PASS      "...your password..."

WiFiClient client;

void setup() {
  Serial.begin(115200);
  delay(10);

  Serial.println(); Serial.println();
  Serial.print("Connecting to ");
  Serial.println(WLAN_SSID);

  WiFi.begin(WLAN_SSID, WLAN_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();

  Serial.println("WiFi connected");
}
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Serial.println("IP address: "); Serial.println(WiFi.localIP());  
}  
  
void loop() {  
    if (Serial.available()) {  
        Serial.write(Serial.read());  
    }  
}
```

6. Código implementado para la placa Arduino para la comunicación en la prueba de concepto

```
#define BLYNK_TEMPLATE_ID          "TMPL__H-Qc2T"  
#define BLYNK_DEVICE_NAME         "Quickstart Device"  
#define BLYNK_AUTH_TOKEN          "mkZWgRd5fV7LKBBGUUwK8w9JXiNkuFRo"  
  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "ssid";  
char pass[] = "pass";  
  
BlynkTimer timer;
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
// This function is called every time the Virtual Pin 0 state changes

BLYNK_WRITE(V0)

{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();
}

BLYNK_WRITE(V1)

{
    // Set incoming value from pin V0 to a variable
    int value = param.asInt();
}

// This function is called every time the device is connected to the Blynk.Cloud
BLYNK_CONNECTED()

{
    // Change Web Link Button message to "Congratulations!"
    Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
    Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
};

    Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

// This function sends Arduino's uptime every second to Virtual Pin 2.
void myTimerEvent()

IoT aplicado al ámbito de la agricultura inteligente en el hogar.
```

```
{  
  // You can send any value at any time.  
  // Please don't send more that 10 values per second.  
  Blynk.virtualWrite(V2, millis() / 1000);  
}  
  
char data[50] = "";  
  
void setup()  
{  
  // Debug console  
  Serial.begin(115200);  
  Serial1.begin(115200);  
  
  Blynk.begin(auth, ssid, pass);  
  // You can also specify server:  
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);  
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);  
  
  // Setup a function to be called every second  
  timer.setInterval(1000L, myTimerEvent);  
}  
  
String temp;  
String hum;  
String lum;  
  
void loop()
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
{  
  Blynk.run();  
  timer.run();  
  
  if (Serial.available()) {  
    Serial.readBytesUntil('>', data, 24);  
    Serial.println();  
  
    temp = String(data).substring(2,7);  
    hum = String(data).substring(9,14);  
    lum = String(data).substring(16,19);  
  
    Serial.println(temp);  
    Serial.println(hum);  
    Serial.println(lum);  
  
    Blynk.virtualWrite(V4, temp);  
    Blynk.virtualWrite(V5, hum);  
    Blynk.virtualWrite(V6, lum);  
  }  
}
```

## 7. Código implementado para la placa Arduino para la prueba de concepto final

```
#include<SoftwareSerial.h>  
  
SoftwareSerial espSerial(5,6);  
  
#include <DHT.h>
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
#define DHTPIN 2

#define LDRPin A3


#define DHTTYPE DHT11


DHT dht(DHTPIN, DHTTYPE);

String str;


const int ledPin = 5;


void setup() {
    Serial.begin(115200);
    espSerial.begin(115200);


    pinMode(LDRPin, INPUT);
    pinMode(ledPin, OUTPUT);


    dht.begin();
    delay(2000);

}


void loop() {
    delay(5000);

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    float f = dht.readTemperature(true);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
if (isnan(h) || isnan(t) || isnan(f)) {  
    Serial.println("Error obteniendo los datos del sensor DHT11");  
    return;  
}
```

```
float hif = dht.computeHeatIndex(f, h);  
float hic = dht.computeHeatIndex(t, h, false);
```

```
int LDRvalue = analogRead(LDRPin);
```

```
Serial.println("H:" + String(h));  
Serial.println("T:" + String(t));  
Serial.println("L:" + String(LDRvalue));
```

```
espSerial.print("H:" + String(h));  
espSerial.print("T:" + String(t));  
espSerial.print("L:" + String(LDRvalue));  
espSerial.println(">");  
delay(5000);
```

```
if(Serial.available()) {  
    String show;  
    show = String(Serial.read());  
    if(show == "48") {  
        digitalWrite(ledPin , LOW);  
    }  
    else if (show == "49") {  
        digitalWrite(ledPin , HIGH);  
    }  
}
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
    }  
  }  
  
}
```

#### 8. Código implementado para la placa Arduino para la comunicación en la prueba de concepto

```
#include<SoftwareSerial.h>  
  
#define BLYNK_TEMPLATE_ID      "TMPL__H-Qc2T"  
#define BLYNK_DEVICE_NAME      "Quickstart Device"  
#define BLYNK_AUTH_TOKEN       "mkZWgRd5fV7LKBBGUUwK8w9JXiNkuFRo"  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
  
char ssid[] = "ssid";  
char pass[] = "pass";  
  
BlynkTimer timer;  
  
BLYNK_WRITE(V0)  
{  
  int value = param.asInt();
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
    Serial1.print(value);
}

BLYNK_WRITE(V1)
{
    int value = param.asInt();
    Serial1.print(value);
}

BLYNK_CONNECTED()
{
    Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
    Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
    Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

void myTimerEvent()
{
    Blynk.virtualWrite(V2, millis() / 1000);
}

char data[50] = "";

void setup()
{
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
// Debug console

Serial.begin(115200);

Serial1.begin(115200);

Blynk.begin(auth, ssid, pass);


timer.setInterval(1000L, myTimerEvent);
}


String temp;
String hum;
String lum;


void loop()
{
  Blynk.run();
  timer.run();


  if (Serial.available()) {
    Serial.readBytesUntil('>', data, 24);
    Serial.println();


    temp = String(data).substring(2,7);
    hum = String(data).substring(9,14);
    lum = String(data).substring(16,19);


    Serial.println(temp);
    Serial.println(hum);
    Serial.println(lum);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Blynk.virtualWrite(V4, temp);  
Blynk.virtualWrite(V5, hum);  
Blynk.virtualWrite(V6, lum);  
}  
}
```

## 9. Código implementado final para la placa de Arduino

```
#include<SoftwareSerial.h>  
  
SoftwareSerial espSerial(5,6);  
  
#include <DHT.h>  
#define DHTPIN 2  
#define WaterPin 3  
#define LDRPin A0  
#define HumGNDPin A1  
  
#define DHTTYPE DHT11  
  
DHT dht(DHTPIN, DHTTYPE);  
String str;  
  
const int relePin = 8;  
  
// Sensor de humedad de tierra valores  
const int AireValue = 531;  
const int WaterValue = 272;
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
int soilMoistureValue = 0;
int soilmoisturepercent=0;

volatile long tActual= 0;
volatile long tAnterior = 0;
volatile long difTiempo = 0;
float factor_conversion=7.11;
int volumen = 0;
int frec = 0;

void interrupcion(){
  tActual = millis();
  difTiempo = tActual - tAnterior;
  tAnterior = tActual;
}

void setup() {
  Serial.begin(115200);
  espSerial.begin(115200);

  pinMode(LDRPin, INPUT);
  pinMode(HumGNDPin, INPUT);
  pinMode(WaterPin, INPUT);
  pinMode(relePin, OUTPUT);
```

```
attachInterrupt(digitalPinToInterrupt(WaterPin),interrupcion,CHANGE);//(Interrupción 0(Pin2),función,Flanco de subida)

    dht.begin();

    delay(2000);

}

void loop() {

    delay(5000);

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {

        Serial.println("Error obteniendo los datos del sensor DHT11");

        return;

    }

    int LDRvalue = analogRead(LDRPin);

    Serial.println("H:" + String(h));

    Serial.println("T:" + String(t));

    Serial.println("L:" + String(LDRvalue));

    espSerial.print("H:" + String(h));

    espSerial.print("T:" + String(t));

    espSerial.print("L:" + String(LDRvalue));
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
// Sensor humedad de tierra

soilMoistureValue = analogRead(HumGNDPin);

soilmoisturepercent = map(soilMoistureValue, AireValue, WaterValue, 0, 100);

if(soilmoisturepercent >= 100)
{
    Serial.println("HM: 100 %");
    espSerial.print("HM:" + String(100));
}
else if(soilmoisturepercent <=0)
{
    Serial.println("HM: 0 %");
    digitalWrite(relePin , HIGH);
    delay(2000);
    digitalWrite(relePin , LOW);
    espSerial.print("HM:" + String(0));
}
else if(soilmoisturepercent >0 && soilmoisturepercent < 100)
{
    Serial.print("HM: ");
    Serial.print(soilmoisturepercent);
    Serial.println("%");
    espSerial.print("HM:" + String(soilmoisturepercent));
}

// Sensor de flujo de agua

float frecuencia = 1000/difTiempo;

float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
volumen=volumen+(caudal_L_m/60)*(difTiempo/1000); //
volumen(L)=caudal(L/s)*tiempo(s)

Serial.print ("W: ");
Serial.println (String(volumen,2));
espSerial.print("W:" + String(volumen));

espSerial.println(">");
delay(5000);

// Pulsador de riego desde Blynk
if(Serial.available()) {
    String show;
    show = String(Serial.read());
    if (show == "49") {
        digitalWrite(relePin , HIGH);
        delay(2000);
        digitalWrite(relePin , LOW);
    }
}

// Automatización del riego
if (soilmoisturepercent <=30) {
    digitalWrite(relePin , HIGH);
    delay(2000);
    digitalWrite(relePin , LOW);
}

if (h <= 15 && t >= 35) {
```

```
while (volumen <= 1){  
    digitalWrite(relePin , HIGH);  
}  
digitalWrite(relePin , LOW);  
}  
  
}
```

#### 10. Código implementado final para la placa de NodeMCU ESP8266

```
#include<SoftwareSerial.h>  
  
#define BLYNK_TEMPLATE_ID          "TMPL__H-Qc2T"  
#define BLYNK_DEVICE_NAME          "Quickstart Device"  
#define BLYNK_AUTH_TOKEN           "auth_token"  
#define BLYNK_PRINT Serial  
  
#include <ESP8266WiFi.h>  
#include <BlynkSimpleEsp8266.h>  
  
char auth[] = BLYNK_AUTH_TOKEN;  
char ssid[] = "ssid";  
char pass[] = "pass";  
  
BlynkTimer timer;  
  
BLYNK_WRITE(V0)  
{  
    int value = param.asInt();  
    Serial.println(value);  
}
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Serial1.print(value);

// Pasados 2 segundos se pone en off automáticamente el sensor

delay(2000);

Blynk.virtualWrite(V0, 0);
}

BLYNK_CONNECTED()
{
    Blynk.setProperty(V3, "offImageUrl", "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");

    Blynk.setProperty(V3, "onImageUrl", "https://static-
image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png"
);

    Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-
do-i-need-to-blynk/how-quickstart-device-was-made");
}

void myTimerEvent()
{
    Blynk.virtualWrite(V2, millis() / 1000);
}

char data[50] = "";

void setup()
{
    // Debug console

    Serial.begin(115200);

    Serial1.begin(115200);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Blynk.begin(auth, ssid, pass);

timer.setInterval(1000L, myTimerEvent);
}

String temp;
String hum;
String lum;
String humGND;
String water;

void loop()
{
  Blynk.run();
  timer.run();

  if (Serial.available()) {
    Serial.readBytesUntil('>', data, 35);
    Serial.println();

    temp = String(data).substring(2,7);
    hum = String(data).substring(9,14);
    lum = String(data).substring(16,19);
    humGND = String(data).substring(22,25);
    water = String(data).substring(27,31);

    Serial.println(temp);
    Serial.println(hum);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
Serial.println(lum);

Serial.println(humGND);

Serial.println(water);


Blynk.virtualWrite(V4, temp);

Blynk.virtualWrite(V5, hum);

Blynk.virtualWrite(V6, lum);

Blynk.virtualWrite(V7, humGND);

Blynk.virtualWrite(V8, water);

}

}
```

#### 11. Código implementado para la configuración del sensor de humedad de tierra

```
Int SensorPin = A3;

void setup() {

  Serial.begin(115200);

  pinMode(SensorPin, INPUT);

}

void loop() {

  int humedad = analogRead(SensorPin);

  Serial.println(humedad);

  Delay(1000);

}
```

#### 12. Código implementado para la configuración de la placa de Arduino en el tutorial

```
#include<SoftwareSerial.h>

SoftwareSerial espSerial(5,6);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
#include <DHT.h>

#define DHTPIN 2

//#define WaterPin 3

#define LDRPin A0

#define HumGNDPin A1

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

String str;

// Sensor de humedad de tierra valroes

const int AireValue = 531;

const int WaterValue = 272;

int soilMoistureValue = 0;

int soilmoisturepercent=0;

volatile long tActual= 0;

volatile long tAnterior = 0;

volatile long difTiempo = 0;

float factor_conversion=7.11;

int volumen = 0;

int frec = 0;

void interrupcion(){

    tActual = millis();

    difTiempo = tActual - tAnterior;

    tAnterior = tActual;

}

void setup() {

    Serial.begin(115200);

    espSerial.begin(115200);

    pinMode(LDRPin, INPUT);

    pinMode(HumGNDPin, INPUT);

    // pinMode(WaterPin, INPUT);

    pinMode(ledPin, OUTPUT);
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
//      attachInterrupt(digitalPinToInterrupt(WaterPin),interrupcion,CHANGE);//(Interrupción
0(Pin2),función,Flanco de subida)

    dht.begin();

    delay(2000);
}

void loop() {

    delay(5000);

    float h = dht.readHumidity();

    float t = dht.readTemperature();

    float f = dht.readTemperature(true);

    if (isnan(h) || isnan(t) || isnan(f)) {

        Serial.println("Error obteniendo los datos del sensor DHT11");

        return;

    }

    float hif = dht.computeHeatIndex(f, h);

    float hic = dht.computeHeatIndex(t, h, false);

    int LDRvalue = analogRead(LDRPin);

    Serial.println("H:" + String(h));

    Serial.println("T:" + String(t));

    Serial.println("L:" + String(LDRvalue));

    espSerial.print("H:" + String(h));

    espSerial.print("T:" + String(t));

    espSerial.print("L:" + String(LDRvalue));

    // Sensor humedad de tierra

    soilMoistureValue = analogRead(HumGNDPin);

    soilmoisturepercent = map(soilMoistureValue, AireValue, WaterValue, 0, 100);

    if(soilmoisturepercent >= 100)

    {

        Serial.println("HM: 100 %");

        espSerial.print("HM:" + String(100));

    }

    else if(soilmoisturepercent <=0)
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
{
  Serial.println("HM: 0 %");
  espSerial.print("HM:" + String(0));
}
else if(soilmoisturepercent >0 && soilmoisturepercent < 100)
{
  Serial.print("HM: ");
  Serial.print(soilmoisturepercent);
  Serial.println("%");
  espSerial.print("HM:" + String(soilmoisturepercent));
}

// Sensor de flujo de agua
// float frecuencia = 1000/difTiempo;
// float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m
// volumen=volumen+(caudal_L_m/60)*(difTiempo/1000); // volumen(L)=caudal(L/s)*tiempo(s)
// Serial.print ("W: ");
// Serial.println (String(volumen,2));
// espSerial.print("W:" + String(volumen));
espSerial.println(">");
delay(5000);
}
```

### 13. Código implementado para la configuración de la placa NodeMCU ESP8266 en el tutorial

```
#include<SoftwareSerial.h>

#define BLYNK_TEMPLATE_ID          "Template_ID"
#define BLYNK_DEVICE_NAME          "Device_Name"
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
char ssid[] = "ssid router";
char pass[] = "password router";

BlynkTimer timer;

BLYNK_WRITE(V0)
{
    int value = param.asInt();
    Serial1.print(value);
}

BLYNK_CONNECTED()
{
    Blynk.setProperty(V3, "offImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations.png");
    Blynk.setProperty(V3, "onImageUrl", "https://static-image.nyc3.cdn.digitaloceanspaces.com/general/fte/congratulations_pressed.png");
    Blynk.setProperty(V3, "url", "https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made");
}

void myTimerEvent()
{
    Blynk.virtualWrite(V2, millis() / 1000);
}

char data[50] = "";

void setup()
{
    // Debug console
    Serial.begin(115200);
    Serial1.begin(115200);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(1000L, myTimerEvent);
}

String temp;
```

IoT aplicado al ámbito de la agricultura inteligente en el hogar.

```
String hum;

String lum;

String humGND;

String water;

void loop()
{
  Blynk.run();
  timer.run();
  if (Serial.available()) {
    Serial.readBytesUntil('>', data, 35);
    Serial.println();
    temp = String(data).substring(2,7);
    hum = String(data).substring(9,14);
    lum = String(data).substring(16,19);
    humGND = String(data).substring(22,25);
    //   water = String(data).substring(27,31);
    Serial.println(temp);
    Serial.println(hum);
    Serial.println(lum);
    Serial.println(humGND);
    //   Serial.println(water);
    Blynk.virtualWrite(V4, temp);
    Blynk.virtualWrite(V5, hum);
    Blynk.virtualWrite(V6, lum);
    Blynk.virtualWrite(V7, humGND);
    //   Blynk.virtualWrite(V8, water);
  }
}
```

## Anexo B. Links de los materiales

1. Kit básico Arduino:

[https://es.aliexpress.com/item/1005003577722990.html?gatewayAdapt=glo2esp&spm=a2g0o.productlist.0.0.6d06282e2RrV8g&algo\\_pvid=166ad648-567f-4514-9c69-9f86da77cc54&algo\\_exp\\_id=166ad648-567f-4514-9c69-9f86da77cc54-26&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2212000026352296800%22%7D&pdp\\_pi=-1%3B9.44%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/1005003577722990.html?gatewayAdapt=glo2esp&spm=a2g0o.productlist.0.0.6d06282e2RrV8g&algo_pvid=166ad648-567f-4514-9c69-9f86da77cc54&algo_exp_id=166ad648-567f-4514-9c69-9f86da77cc54-26&pdp_ext_f=%7B%22sku_id%22%3A%2212000026352296800%22%7D&pdp_pi=-1%3B9.44%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch)

2. Kit básico riego:

[https://es.aliexpress.com/item/1005003601545735.html?spm=a2g0o.productlist.0.0.28e44b03vpcTCY&algo\\_pvid=fdee82d6-c34d-479b-a23f-1d2ee20a9f76&algo\\_exp\\_id=fdee82d6-c34d-479b-a23f-1d2ee20a9f76-1&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2212000026452954674%22%7D&pdp\\_pi=-1%3B1.77%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/1005003601545735.html?spm=a2g0o.productlist.0.0.28e44b03vpcTCY&algo_pvid=fdee82d6-c34d-479b-a23f-1d2ee20a9f76&algo_exp_id=fdee82d6-c34d-479b-a23f-1d2ee20a9f76-1&pdp_ext_f=%7B%22sku_id%22%3A%2212000026452954674%22%7D&pdp_pi=-1%3B1.77%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch)

3. Sensor DHT11:

[https://es.aliexpress.com/item/32642239022.html?spm=a2g0o.productlist.0.0.6a895f7613APD8&algo\\_pvid=013accd3-a416-4271-a284-4beb4609bbaf&algo\\_exp\\_id=013accd3-a416-4271-a284-4beb4609bbaf-1&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2259594384219%22%7D&pdp\\_pi=-1%3B0.79%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/32642239022.html?spm=a2g0o.productlist.0.0.6a895f7613APD8&algo_pvid=013accd3-a416-4271-a284-4beb4609bbaf&algo_exp_id=013accd3-a416-4271-a284-4beb4609bbaf-1&pdp_ext_f=%7B%22sku_id%22%3A%2259594384219%22%7D&pdp_pi=-1%3B0.79%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch)

4. Fotorresistencia LDR:

[https://es.aliexpress.com/item/1005003629058275.html?spm=a2g0o.productlist.0.0.1fc43dc0tuxTkL&algo\\_pvid=df2e52f2-975e-4072-986b-310fa6aa06b1&algo\\_exp\\_id=df2e52f2-975e-4072-986b-310fa6aa06b1-0&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%2212000026565486021%22%7D&pdp\\_pi=-1%3B0.94%3B-1%3B29%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/1005003629058275.html?spm=a2g0o.productlist.0.0.1fc43dc0tuxTkL&algo_pvid=df2e52f2-975e-4072-986b-310fa6aa06b1&algo_exp_id=df2e52f2-975e-4072-986b-310fa6aa06b1-0&pdp_ext_f=%7B%22sku_id%22%3A%2212000026565486021%22%7D&pdp_pi=-1%3B0.94%3B-1%3B29%40salePrice%3BEUR%3Bsearch-mainSearch)

5. Sensor de flujo de agua:

[https://es.aliexpress.com/item/1005001652070023.html?spm=a2g0o.productlist.0.0.2a09343fjX8bmJ&algo\\_pvid=e11e584e-885b-4e1b-a78f-f6da53983f3c&algo\\_exp\\_id=e11e584e-885b-4e1b-a78f-f6da53983f3c-0&pdp\\_ext\\_f=%7B%22sku\\_id%22%3A%22-1%22%7D&pdp\\_pi=-1%3B3.08%3B-1%3B-1%40unknownPrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/1005001652070023.html?spm=a2g0o.productlist.0.0.2a09343fjX8bmJ&algo_pvid=e11e584e-885b-4e1b-a78f-f6da53983f3c&algo_exp_id=e11e584e-885b-4e1b-a78f-f6da53983f3c-0&pdp_ext_f=%7B%22sku_id%22%3A%22-1%22%7D&pdp_pi=-1%3B3.08%3B-1%3B-1%40unknownPrice%3BEUR%3Bsearch-mainSearch)

6. PLA impresión 3D:

[https://es.aliexpress.com/item/1005002276992733.html?spm=a2g0o.productlist.0.0.45837bc6XfH4tE&algo\\_pvid=224b7996-d410-4b55-b218-0547d85a251d&algo\\_exp\\_id=224b7996-d410-4b55-b218-0547d85a251d-](https://es.aliexpress.com/item/1005002276992733.html?spm=a2g0o.productlist.0.0.45837bc6XfH4tE&algo_pvid=224b7996-d410-4b55-b218-0547d85a251d&algo_exp_id=224b7996-d410-4b55-b218-0547d85a251d-)

[5&pdp\\_ext f=%7B%22sku\\_id%22%3A%2212000019878355555%22%7D&pdp\\_pi=-1%3B17.87%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](#)

7. NodeMCU ESP8266:

[https://es.aliexpress.com/item/32520574539.html?spm=a2g0o.productlist.0.0.3e222561loUPYt&algo\\_pvid=cc5b7553-53dc-45ae-a271-8cb70aef194f&algo\\_exp\\_id=cc5b7553-53dc-45ae-a271-8cb70aef194f-0&pdp\\_ext f=%7B%22sku\\_id%22%3A%2259096362340%22%7D&pdp\\_pi=-1%3B2.13%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/32520574539.html?spm=a2g0o.productlist.0.0.3e222561loUPYt&algo_pvid=cc5b7553-53dc-45ae-a271-8cb70aef194f&algo_exp_id=cc5b7553-53dc-45ae-a271-8cb70aef194f-0&pdp_ext f=%7B%22sku_id%22%3A%2259096362340%22%7D&pdp_pi=-1%3B2.13%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch)

8. Tablas de madera: <https://www.leroymerlin.es/fp/10242190/tablero-macizo-de-pino-de-30x120x1-8-cm>

9. Pila 9V:

[https://es.aliexpress.com/item/1005003843793171.html?spm=a2g0o.productlist.0.0.10fb43dd8o9Tvn&algo\\_pvid=3f274204-18ec-4701-8c39-e021692550ae&algo\\_exp\\_id=3f274204-18ec-4701-8c39-e021692550ae-2&pdp\\_ext f=%7B%22sku\\_id%22%3A%2212000027319738772%22%7D&pdp\\_pi=-1%3B2.39%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch](https://es.aliexpress.com/item/1005003843793171.html?spm=a2g0o.productlist.0.0.10fb43dd8o9Tvn&algo_pvid=3f274204-18ec-4701-8c39-e021692550ae&algo_exp_id=3f274204-18ec-4701-8c39-e021692550ae-2&pdp_ext f=%7B%22sku_id%22%3A%2212000027319738772%22%7D&pdp_pi=-1%3B2.39%3B-1%3B-1%40salePrice%3BEUR%3Bsearch-mainSearch)