

**UNIVERSIDAD ALFONSO X EL SABIO**

**ESCUELA POLITÉCNICA SUPERIOR**

**GRADUADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL**

**MÓDULO DE DOMÓTICA CON ARDUINO EN RED DE IOT**

**ALUMNO:** GOZALO DÍAZ DEL RÍO FONTÁN

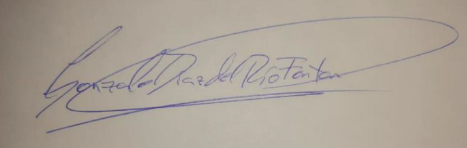
**N.P.:** 121291

**DIRECTOR DEL TRABAJO:** MARIA ANTONIA SIMÓN

**FECHA DE PRESENTACIÓN:** 8 DE SEPTIEMBRE DE 2020

**BREVE DESCRIPCIÓN:**

DESARROLLO DE UN MÓDULO DE CONTROL REMOTO AUTOMÁTICO Y VERSÁTIL PARA APLICACIONES DE DOMÓTICA, ASÍ COMO DEL SISTEMA DE CONTROL Y GESTIÓN PARA ESTOS MÓDULOS MEDIANTE EL USO DE ARDUINO Y RASPBERRY PI. APLICACIONES DE IOT EN LA DOMÓTICA Y LA INFORMÁTICA INDUSTRIAL.

**FIRMA DEL DIRECTOR FIRMA DEL TUTOR FIRMA DEL ALUMNO**

[1. Objetivos y alcance del Proyecto 2](#_Toc49419258)

[2. Introducción 3](#_Toc49419259)

[2.1 Estudio de concepto 3](#_Toc49419260)

[2.2 Internet de las Cosas 3](#_Toc49419261)

[2.3 Domótica 3](#_Toc49419262)

[2.4 Tecnologías y Lenguajes de Programación 5](#_Toc49419263)

[3. Memoria del Proyecto 8](#_Toc49419264)

[3.1 Materiales y programas Empleados 8](#_Toc49419265)

[3.2 Fases del Proyecto 11](#_Toc49419266)

[3.3 Programación 11](#_Toc49419267)

[3.4 Implementación del circuito 13](#_Toc49419268)

[4. Pliego de condiciones 14](#_Toc49419269)

[4.1 Conectividad 14](#_Toc49419270)

[4.2 Refrigeración 14](#_Toc49419271)

[4.3 Montaje 14](#_Toc49419272)

[5. Presupuesto 15](#_Toc49419273)

[6. Planos 16](#_Toc49419274)

[7. Anexos 17](#_Toc49419275)

[7.1 Programas 17](#_Toc49419276)

[7.2 Alternativa al Punto de Acceso en Raspberry Pi 21](#_Toc49419277)

[8. Bibliografía 23](#_Toc49419278)

[9. Conclusión 24](#_Toc49419279)

# Objetivos y alcance del Proyecto

En este proyecto se pretende diseñar un módulo controlador para ser empleado en ámbito de la domótica. Dicho módulo consistirá en una placa microcontroladora Arduino con un módulo de Wifi acoplado para permitir las comunicaciones inalámbricas. Estos módulos serán capaces de activar o desactivar actuadores en función de comandos originados en un ordenador central en la forma de una Raspberry Pi, así como de leer distintos sensores y comunicar esta lectura al ordenador.

En el consiguiente proyecto se realizará la programación y montaje de un módulo de Automatización mediante el uso de un Arduino Nano equipado con una antena de Wifi, así como la configuración de un Ordenador de Control.

El objetivo del proyecto es crear un módulo barato y accesible que permita controlar de forma inalámbrica cualquier dispositivo al que se conecte. Este control será posible realizarlo desde cualquier dispositivo capaz de comunicarse con una red Wifi.

El principal enfoque del proyecto es la versatilidad del producto. Se pretende que los módulos controladores puedan ser empleados para actuar sobre una gran variedad de dispositivos sin requerir alteraciones en su código. Para ello se pretende centralizar el control y la automatización en el ordenador central, minimizando la algoritmia contenida en los microcontroladores.

# Introducción

## Estudio de concepto

Nos encontramos en una época en la que la automatización en el ámbito de la domótica está en pleno auge. En todos los comercios de electrónica se venden y promocionan los dispositivos “Smart-home”, que prometen emular la existencia de un mayordomo virtual que encienda o apague luces, abra y cierre puertas, controle la temperatura, realice videovigilancia… Decenas de productos y funciones distintas se ponen a nuestra disposición para facilitarnos una interacción más automática con nuestro día a día.

Sin embargo, estas múltiples soluciones se encuentran siempre asociadas a una marca comercial, con Amazon Echo y Google Home a la cabeza del mercado actual, y requieren que todos los componentes del sistema pertenezcan a la misma familia. Con este proyecto se pretende crear un dispositivo que nos permita ejercer este nivel de control sobre aparatos y electrodomésticos “no inteligentes”, sirviendo así de puente para permitir la automatización domótica sin necesidad de actualizar todos los aparatos ya presentes en casa.

## Internet de las Cosas

El Internet de las Cosas (Internet of Things, IoT) es un sistema de dispositivos de computación, controladores mecánicos o digitales y otros aparatos interconectados entre sí con identificadores únicos. Esta red de dispositivos cuenta con la capacidad de transmitir datos entre sí y realizar diversas tareas sin necesidad de interacción humana. El Internet de las Cosas nace de los sistemas empotrados, la necesidad de análisis en tiempo real del machine learning.

Actualmente las aplicaciones del Internet de las Cosas están siendo constantemente expandidas y exploradas por la Industria para todo tipo de aplicaciones. Gracias al IoT, los sistemas cableados tradicionales y los complejos y delicados autómatas, omnipresentes en ámbitos industriales, están empezando a desaparecer para ser sustituidos por dispositivos más modernos con capacidades de IoT, mucho más versátiles y modulares.

## Domótica

La domótica (Unión del latín *domus*, casa, y *robótica)*, en su más básica definición, se trata de la implementación e integración de la tecnología en el diseño inteligente de un recinto cerrado. La domótica engloba una gran variedad de servicios y aplicaciones como la seguridad, la accesibilidad o el entretenimiento. El foco principal de la domótica es facilitar la vida doméstica proporcionando al usuario un nivel superior de interacción y control sobre su entorno mediante la instalación de dispositivos inteligentes que actúan de interfase entre humano y aplicación doméstica. Así, un frigorífico capaz de realizar pedidos al supermercado cuando un producto se acaba, un sistema de iluminación activado por voz o un termostato que se puede programar desde el teléfono móvil, independientemente de la distancia al hogar, todos entran bajo el concepto de la domótica.

### Historia de la Domótica

Los inicios de la Domótica se pueden atribuir a la distribución de energía a todos los hogares a principios del siglo 20. Este avance permitió la pronta aparición de elementos como máquinas lavavajillas, lavadoras automáticas de ropa o calentadores de agua. Estos electrodomésticos que ahora consideramos parte esencial de un hogar moderno fueron máquinas revolucionarias que permitieron automatizar tareas cotidianas y aminorar el esfuerzo humano necesario para llevarlas a cabo.

No fue hasta 1975 sin embargo que se desarrolló la primera tecnología de red de comunicación orientada hacia la domótica. Esta tecnología era el protocolo de comunicaciones X10, aún empleado en la industria moderna. Desarrollado en Escocia por Pico Electronics, X10 emplea las mismas líneas de corriente que la red eléctrica doméstica del interior del hogar para transmitir sus mensajes. Mediante picos de tensión en frecuencias electromagnéticas determinadas, este protocolo es capaz de transmitir instrucciones e información simple a una enorme variedad de dispositivos conectados a la red.

En la actualidad la domótica se está desviando hacia las comunicaciones inalámbricas, ya sean mediante infrarrojos, Bluetooth o wifi, y la aparición de nuevas tecnologías solo facilita el crecimiento del sector. Las estadísticas muestran que en Estados Unidos, a finales de 2012 1.5 millones de hogares tenían alguna forma de domótica instalada mientras que para el final de 2018, el número había aumentado hasta los 45 millones de viviendas. Este crecimiento exponencial sigue en pleno auge en todo el planeta y cada día surgen nuevas tecnologías para facilitar cada vez más las tareas del día a día.

### Aplicaciones de la Domótica

Dentro de las innumerables aplicaciones que existen de la domótica, se discutirán las más presentes y habituales:

#### Calefacción y aire acondicionado (Control de temperatura)

Estos sistemas controlan la calefacción, refrigeración y ventilación del hogar. Ya sea a través de termostatos que han de ser configurados manualmente o de sistemas inteligentes completamente conectados a la red, estos sistemas se encuentran presentes en la gran mayoría de los hogares modernos. Los modelos más nuevos permiten monitorizar y controlar la climatización individual de cada habitación del hogar desde cualquier lugar empleando un dispositivo móvil.

#### Vigilancia y seguridad

Puede ser empleando cámaras, micrófonos, sensores de CO2 o cualquier otra cantidad de distintas tecnologías, pero la domótica está muy presente en los sistemas de seguridad del hogar. Desde cámaras de circuito cerrado que almacenan sus grabaciones en discos duros hasta sistemas de detección de movimiento y ocupación que transmiten sus datos en directo a la nube para ser accedidos desde un móvil, la domótica permite aumentar la seguridad del hogar.

#### Control de dispositivos para eficiencia energética

Esta aplicación trata de regular el consumo del hogar de manera que se reduzcan al máximo el coste y la contaminación generados. Un hogar equipado con un módulo fotovoltaico, por ejemplo, puede disponer de un sistema domotizado que restrinja y automatice el uso de electrodomésticos de gran consumo como una lavadora o una secadora a aquellas horas en las que el Sol genere más energía. De la misma manera, un automatismo puede contener programadas aquellas franjas horarias en los que la compañía eléctrica cobra menos por potencia consumida y aprovecharlos para activar funciones o aparatos que consumen más energía, manteniéndolos en reposo durante las franjas de mayor consumo. Sistemas como estos pueden llegar a reducir de forma sorprendente los costes energéticos del hogar y son enormemente compatibles con otros sistemas inteligentes de control del hogar.

#### Accesibilidad para personas de avanzada edad y personas discapacitadas

Una aplicación que está creciendo de forma muy acelerada cuanto más avanza la tecnología es la de aumentar la accesibilidad al hogar de las personas con movilidad reducida, problemas de visión u otras discapacidades. Estos dispositivos permiten a una persona con movilidad reducida subir o bajar escaleras sin dificultad, abrir o cerrar puertas y tantas otras tareas cotidianas. Un asistente del hogar con interfaz de voz puede facilitar las tareas diarias a una persona con discapacidad visual.

## Tecnologías y Lenguajes de Programación

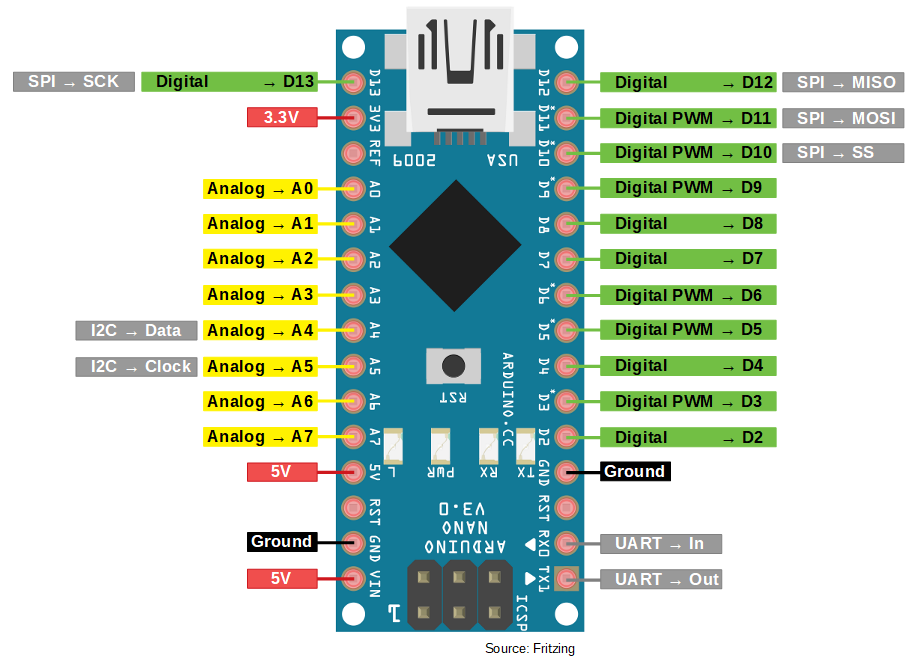
Para la realización de este proyecto se han empleado una variedad de tecnologías y lenguajes de programación, que se detallan a continuación:

### Arduino

Las placas de desarrollo Arduino son una colección de microcontroladores con una memoria flash y varios pines de entrada/salida asociados recogidos bajo una licencia de hardware libre. Su principal uso es en el ámbito de la robótica y la automatización y son programables a través de un software de licencia libre empleando el lenguaje C++. Una placa de Arduino puede leer y modificar el estado de tensión de sus pines tanto de forma discreta digital, con dos niveles lógicos, como de forma analógica, permitiendo leer y emitir valores continuos.

Se trata de una pieza de hardware barata, resistente, sencilla de programar, muy personalizable y enormemente versátil. Existen numerosos modelos con diferencias en el número de pines, la potencia del microcontrolador, el tamaño de la memoria y el número de pines. Para este proyecto se empleará el modelo Nano debido a su pequeño tamaño, pese a que el funcionamiento es idéntico si se desea emplear el modelo UNO o MEGA.

#### Arduino Nano

Entre las múltiples variedades de placas de desarrollo de Arduino, el Nano es la versión más reducida de tamaño que sin embargo mantiene todas las funcionalidades del modelo UNO, el más tradicional. El Nano cuenta con un controlador ATmega328P de 16MHz, con 1KB de EEPROM, 32 KB de memoria Flash y 2KB de SRAM. Para funcionar requiere alimentación entre 7 y 12 Voltios y opera sus entradas y salidas a 5V. Cuenta con 14 pines que pueden ejercer de entradas / salidas digitales (6 que permiten salidas PWM) y 8 analógicos. Puede emitir hasta 40mA de corriente por cada pin. Puede conectarse por USB a un ordenador para su programación y alimentación o puede ser alimentado a través de uno de sus pines. A través de otro de sus pines puede alimentar dispositivos a 3.3 V.

### Raspberry Pi 4 Model B – PimoroniRaspberry Pi

Las Raspberry Pi son una serie de pequeños ordenadores de una sola placa. Sus aplicaciones son prácticamente ilimitadas y se encuentran presentes en aplicaciones industriales, educativas, investigadoras y en robótica, destacando gracias a su sorprendente potencia, versatilidad y portabilidad. Existen múltiples generaciones del producto, cada una con capacidades mejoradas. Para este proyecto se empleará una Raspberry Pi 4 Modelo B de 4 Gb de memoria, pese a que el proyecto se puede replicar a la perfección con cualquier modelo a partir de la Raspberry Pi 3.

Para su funcionamiento, como con cualquier ordenador, la Raspberry Pi requiere un Sistema Operativo. Este sistema se instala en una tarjeta SD que luego se inserta en la placa y actúa como disco duro del sistema. Hay numerosas imágenes distintas para Raspberry, dependiendo de la funcionalidad que se desee de ellas. Por su versatilidad y lo extendido de su uso, emplearemos Raspbian, una distribución de Debian adaptada a la Raspberry. Esto significa que, con muy pocas modificaciones, este proyecto puede funcionar con cualquier ordenador operando con una distribución de Linux, como por ejemplo Ubuntu.

### node-red-dashboard (node) - Node-REDRunning Node-RED on IBM i: Installation and first flow – Build Smart. Build Secure. IBM DeveloperNode-Red

Node-Red es una herramienta de programación diseñada por IBM para conectar dispositivos de hardware, APIs y servicios online dentro del ámbito del IoT. Consiste en una interfaz web alojada en un dispositivo que ejerce de anfitrión. En esta interfaz se pueden programar flujos de nodos que intercambian datos en forma de mensajes en formato JSON (JavaScript Object Notation). Al tratarse de una herramienta Open Source, cualquier usuario puede contribuir al proyecto sus propios nodos y funciones, por lo que el proyecto crece exponencialmente con su comunidad y es muy sencillo encontrar nodos y funciones personalizados para la aplicación que se desea. También cuanta con un nodo que permite programar funciones propias en lenguaje Javascript. Al haber sido desarrollado con IBM, su implementación junto con nodos de MQTT lo convierte en la más versátil herramienta para el IoT.

### MQTT

MQTT (Message Queing Telemetry Transport) es un protocolo de comunicaciones de red muy ligero basado en mecánicas de publicación-subscripción que transmite mensajes entre dispositivos, generalmente haciendo uso de conexiones TCP/IP (comunicaciones de red de Internet). Sin embargo, también está soportado por cualquier protocolo que ofrezca comunicaciones bidireccionales sin pérdidas. Su utilidad destaca en aplicaciones que requieran un código muy ligero o en sistemas con ancho de banda limitado.

El funcionamiento de MQTT es muy simple: los elementos que participan en la comunicación son un gestor de mensajes, el broker, y en número indeterminado de clientes. Los componentes de un mensaje MQTT son un tema y un contenido. Un cliente interactúa con el tema o temas deseados de dos maneras: puede **publicar** un mensaje en un tema concreto y puede **suscribirse** a uno o más temas. El gestor de mensajes se encarga de registrar los mensajes publicados en cada tema y distribuirlos de forma inmediata a todos los clientes suscritos a dicho tema. Existen funcionalidades que permiten asegurar distintas Calidades de Servicio (Quality of Service, QoS) para la comunicación:

* QoS 0: En modo QoS 0, el gestor envía los mensajes a los clientes suscritos en cuanto los recibe y después borra el mensaje. También se conoce como modo “Fire and Forget”.
* QoS 1: En modo QoS 1, el gestor al recibir un mensaje, lo reenviará a los clientes suscritos y esperará una confirmación de mensaje recibido antes de borrarlo. De no recibir la confirmación, volverá a enviar el mensaje de forma periódica hasta recibir la confirmación. Este modo se conoce también como “At least Once”, Al menos una vez.
* QoS 2: En modo QoS 2 el gestor se asegura de que el mensaje le llega al cliente esperando una confirmación por parte de este y luego vuelve a consultar con el cliente si la comunicación ha tenido éxito antes de borrar el mensaje. Este modo se conoce como “Once and Only Once”, Una vez y sólo una vez. QoS 2 también permite emplear la persistencia de mensajes, un sistema mediante el cual el gestor siempre almacena un registro de todos los mensajes enviados a cada tema para que si un nuevo cliente se suscribe a ese tema reciba todos los mensajes que se ha “perdido” antes de su suscripción.

Para la aplicación que nos ocupa emplearemos QoS 2, ya que la información que transmitiremos no es de naturaleza crítica.

### PuTTY, FileZilla

PuTTY es un cliente de SSH que permite acceder a la consola de comandos de una máquina de Ubuntu (En este caso el sistema Raspbian de la Raspberry Pi) y manejarla a través de una red a la que esté conectada. FileZilla permite la transferencia de archivos entre dos equipos empleando SSH. Ambos programas son necesarios para operar la Raspberry Pi.

# Memoria del Proyecto

## Materiales y programas Empleados

### Hardware

#### Raspberry Pi 4 Modelo B (4Gb).

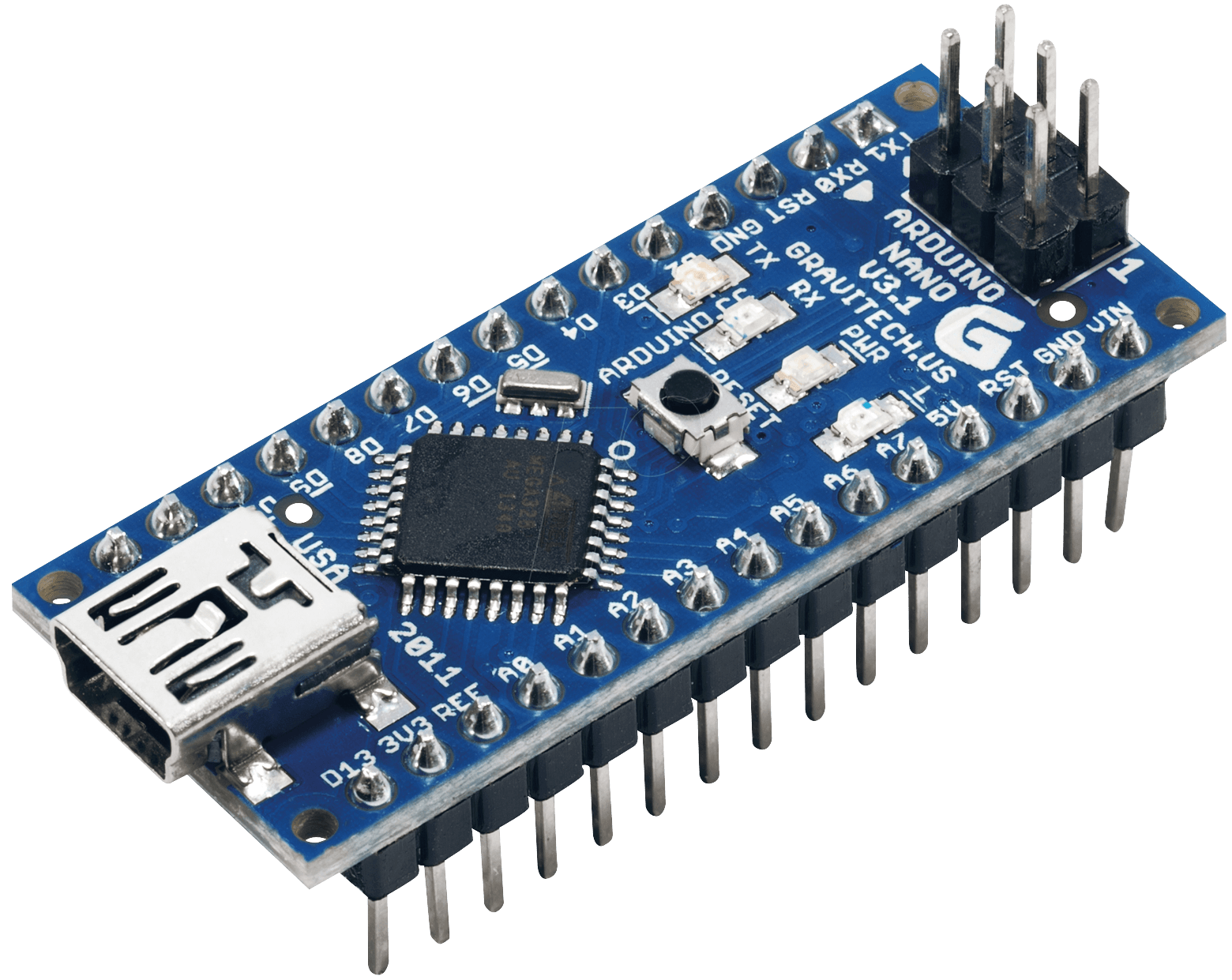
Puede ser sustituida por cualquier modelo igual o posterior a la Raspberry Pi 3. Para este proyecto se ha empleado la Pi 4 modelo B de 4 Gb de memoria RAM. Si se va a emplear el modelo 4 es recomendable acompañarlo de una carcasa con radiadores y ventilador incorporado, ya que es necesario facilitar la disipación de calor

#### Pantalla táctil LCD para Raspberry Pi (Opcional)

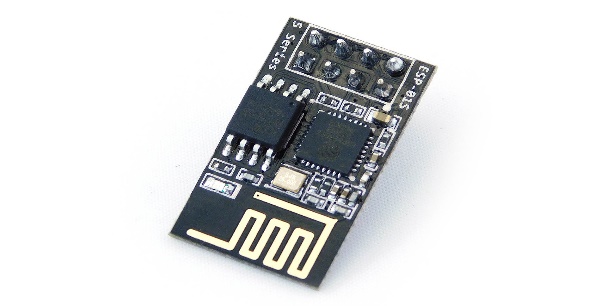
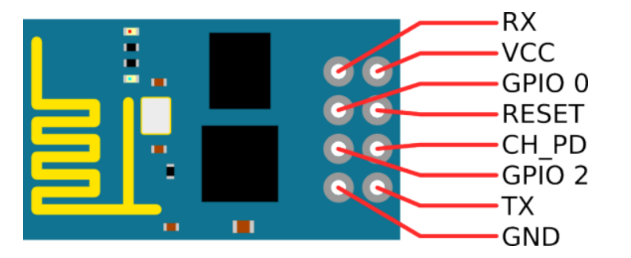
No es necesario disponer de pantalla LCD para la Raspberry, pero puede servir de otra interfaz más para interactuar con el sistema



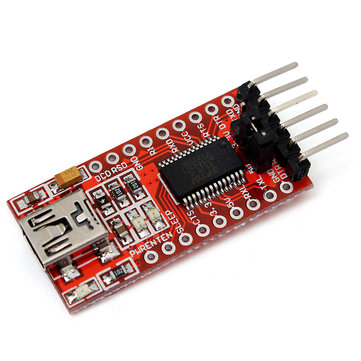
#### Arduino Nano

El modelo empleado de Arduino nano es el V3.0, que emplea un chip FTDI para comunicación serie con el ordenador. No debería haber ningún inconveniente usando un clon de Arduino (ELEGOO, por ejemplo), simplemente será necesario seleccionar el complemento correspondiente en la IDE de Arduino al escribir el programa.

#### Módulo de Wifi ESP8266 ES01

Existen varias versiones de este producto, pero parecen ser todas compatibles. Se puede emplear otro módulo de wifi compatible con Arduino si se desea, siempre que sea programable en C++ empleando el IDE de Arduino. Debido a que posee dos pines para comunicación serie, es necesario un adaptador para poder programarlo desde el ordenador. Algunas fuentes dicen que se puede emplear el propio Arduino para ello, pero no parece una técnica fiable o fácilmente replicable.

#### Conversor FTDI Mini USB a TTL Serie

Se trata simplemente de un adaptador que permite programar el ESP8266 desde el ordenador conectándolo con un cable USB.

#### Amazon.com: GFORTUN 10Pcs 4cm x 6cm Double Sided 2.54mm Protoboard ...Placas PCB perforadas para soldadura con estaño

#### Soldador, estaño, cables

#### Transformador de tensión 220V AC – 12 V DC

#### Convertidores de tensión 12V – 5V y 5V – 3.3V

#### Transistores y actuadores varios para demostración (Ventiladores, bombillas LED, etc.)

### Software

#### Raspberry Pi

Para la Raspberry Pi, se empleará la versión más actualizada de **Raspbian**, una distribución gratuita basada en Debian (Ubuntu) y optimizada para la Raspberry Pi.

Ya incluido en esta distribución se encuentra **Node-Red**, pese a que, para mayor seguridad acerca de la compatibilidad de todas las partes del proyecto se reinstalará la versión más moderna.

También es necesario un bróker de MQTT. Se empleará mosquitto por su facilidad de instalación y gestión.

Si se desea emplear una pantalla táctil LCD para Raspberry Pi, también será necesario instalar los controladores de dicha pantalla, que permiten emplear los puertos de entrada / salida genéricos de la Raspberry (GPIO) para transmitir imagen a la pantalla y leer las señales introducidas con el puntero táctil.

#### ESP8266

Para poder programar y operar correctamente con el ESP8266, es necesario descargar las bibliotecas *ESP8266WiFi.h* y *PubSubClient.h* de C++ en el ordenador que se vaya a emplear para la programación. Esto puede hacerse muy fácilmente desde el gestor de librerías de la IDE de Arduino.

#### PC

La principal herramienta que se necesita en el PC es la IDE de Arduino. Las siglas IDE, en inglés significan “Integrated Development Enviroment” (Entorno integrado de Desarrollo) y se refieren a un software que permita programar código en uno o varios lenguajes de programación distintos. La IDE de Arduino es un entorno gratuito proporcionado por la propia empresa Arduino Software a través de su página web, y permite programar código en C y C++ para luego grabarlo en microcontroladores de Arduino o similares. Esta herramienta permite ampliar su administrador de placas para escribir programas en más microcontroladores que los predeterminados, por lo que también hace posible programar el módulo ESP8266.

Otras dos herramientas necesarias para el PC son programas de comunicación SSH (Secure Shell) con la Raspberry. SSH es un protocolo de comunicación criptográfico permite acceder a la consola de comandos del sistema Raspbian en la raspberry, así como a su estructura de carpetas permitiendo al usuario tanto enviar órdenes y comandos como transferir archivos, todo desde el PC. Los dos programas de SSH a emplear son PuTTY y FileZilla, el primero permite la conexión con la consola de comandos mientras que el segundo permite realizar transferencias de archivos.

## Fases del Proyecto

A continuación, se detallarán las fases del proyecto para su replicación y los requisitos para el desarrollo de las mismas. Cada fase va precedida de un código que indica a que parte del proyecto pertenece:

RP – Raspberry Pi, TH – Desarrollo teórico, ESP – Módulo Wifi esp8266, AN – Arduino NANO, POT – Potencia

## Programación

### Raspberry Pi

#### RP1 – Raspbian

Requisito: Ninguno

En esta fase se instala el sistema operativo Raspbian en la Raspberry Pi.

Del sitio web de Raspberry se descarga el software Raspberry Pi Imager. Una vez descargado y habiendo insertado una tarjeta SD formateada en el ordenador, se ejecuta el programa y se elige grabar en la tarjeta el sistema NOOBS (New Out Of the Box Software), que permite instalar una variedad de imágenes.

Una vez grabado el sistema NOOBS, se debe conectar la Raspberry a un monitor y un teclado, que emplearemos para navegar por el menú de NOOBS hasta instalar Raspbian lite. Una vez instalado debemos acceder a las opciones de configuración del sistema mediante el comando sudo raspi-config y activar la comunicación SSH en las opciones de Interfaz.

#### RP2 – Wifi

Requisito: RP1

En esta fase la Raspberry Pi se configura como punto de acceso Wifi, generando la red “RaspiHouse”.

Para facilitar este proceso, se ha escrito un programa que contiene los comandos bash necesarios para realizar la configuración. Solo es necesario transferir mediante FileZilla el archivo wifiAPconfig.sh al directorio de usuario de la Raspberry y ejecutar desde ese directorio el comando sudo bash wifiAPconfig.sh.

Si se desea configurar el SSID y la contraseña de la red wifi, se puede editar el ejecutable con sudo nano wifiAPconfig.sh y cambiarlos en las líneas:

sudo echo -e "ssid=**RaspiHouse**" >> /etc/hostapd/hostapd.conf

sudo echo -e "wpa\_passphrase=**MyIoTProject**" >> /etc/hostapd/hostapd.conf

#### RP3 – MQTT

Requisito: RP1

En esta fase se instala y configura mosquitto, el software gestor de mensajes de MQTT que emplearemos. Simplemente es necesario ejecutar los siguientes comandos con la Raspberry conectada a internet:

sudo apt-get update

sudo apt-get install mosquitto

#### RP4 – Node-Red

Requisito: RP3

En esta fase se instala Node-Red junto con los servicios en segundo plano que permiten añadir nodos nuevos y gestionar el funcionamiento del servicio. Simplemente es necesario ejecutar los siguientes comandos con la raspberry conectada a internet:

sudo bash < (curl -sL <https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered>)

sudo systemctl enable nodered.service

sudo npm install node-red-dashboard

Por último es necesario transferir mediante FileZilla el archivo flows.json al directorio /home/pi/.nodered de la raspberry.

En cuanto se haya completado esta fase, ya se puede acceder a Node-Red a través de cualquier dispositivo conectado a la red Wifi de la Raspberry, accediendo al puerto 1880 de la dirección IP de la Raspberry (https://192.168.4.1:1880) en la barra de navegación del navegador). Para acceder a la aplicación se emplea la dirección https://192.168.4.1:1880/ui

### Módulo ESP8266

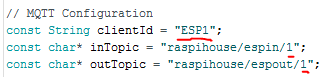
#### ESP1 – Programación

Requisito: RP4

Se programará el módulo ESP8266 empleando el IDE de Arduino. Para ello se debe montar el siguiente circuito con el ESP8266 y el adaptador de FTDI a TTL.

Una vez montado, desde la IDE de Arduino se debe configurar el módulo. Para ello se accede al menú Archivo -> Preferencias y en el campo de gestor de otras placas se añade la línea:

http://arduino.esp8266.com/stable/package\_esp8266com\_index.json

Reiniciamos el IDE de Arduino y ahora en “Herramientas” seleccionamos como tarjeta el módulo genérico de ESP8266. Conectamos el cable al adaptador de FTDI y escribimos el código contenido en el archivo espCode.ino. En este momento se determina el identificador único de este módulo ESP: en el código, bajo “MQTT Configuration” se debe especificar el identificador que se le asignará al módulo (Ver imagen).

Se observa en la imagen que el nombre del cliente que se ha proporcionado es ESP1 y que para la comunicación MQTT, se suscribirá al tema *raspihouse/espin/1* y publicará en *raspihouse/espout/2*. Estos valores pueden ser alterados como se desee.

### Arduino Nano

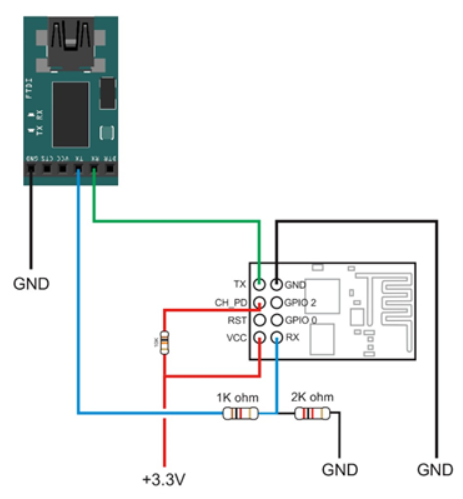
#### AN1 – Programación

Requisito: ESP1

Se programará el Arduino Nano haciendo uso de la IDE de Arduino. Simplemente es necesario conectarlo al PC, seleccionar la placa Nano en el selector de placas y escribir en él el código arduinoCode.ino

## Implementación del circuito

### Circuito de programación del módulo ESP826-01

El primer circuito que es necesario montar se trata del circuito para programar el módulo ESP8266. Este es un circuito provisional, ya que solo se emplea en la fase ESP1.

Circuito de programación de ESP8266-01 con adaptador FTDI a SSL

Según algunas fuentes, es posible que sea necesario conectar el pin GPIO 02 del ESP8266 a GND para activar el modo de programación. Esto no fue necesario en el prototipo de este proyecto, pero puede ser relevante para modelos ligeramente distintos del módulo ESP.

# Pliego de condiciones

Pese a que el proyecto está enfocado a la versatilidad y su aplicación pretende no estar condicionada por ningún factor externo, siendo posible desplegarlo en cualquier entorno sin dificultades, existen unas mínimas condiciones necesarias para su implementación. En este apartado se especificarán esas condiciones, así como se harán notar las precauciones especiales que deben ser tomadas a la hora de montarlo.

## Conectividad

Para el correcto funcionamiento del sistema, es necesario configurar los diferentes componentes del sistema para que se comuniquen entre sí a través de Wifi. En este Proyecto se emplea una red de Wifi generada por la Raspberry Pi para efectuar las comunicaciones, esto requiere que todos los componentes del proyecto se encuentren dentro del alcance de esta red. Dado que para ello es casi indispensable que todos se encuentren en la misma habitación, no se recomienda la implementación del proyecto a distancias mayores.

En el Anexo 3 se puede encontrar una forma de solventar esta limitación: en vez de emplear una red generada por la Raspberry, los módulos individuales y la Raspberry emplean una red Wifi preexistente (una red doméstica que alcance las habitaciones del hogar).

A la hora de cambiar la configuración de redes del sistema Raspbian, se recomienda tener un mínimo conocimiento de su funcionamiento. Esto es porque existe el riesgo de bloquear ciertas comunicaciones que no nos permitieran volver a acceder al sistema, requiriendo un formateo y nueva instalación de Raspbian.

## Refrigeración

La Raspberry Pi empleada es la Pi 4. Este modelo ofrece muy altas prestaciones para su tamaño y coste, pero también presenta problemas de recalentamiento. Si se pretende emplear el modelo 4, es recomendable añadir radiadores y hasta un pequeño ventilador a la Raspberry. La mayoría de carcasas (incluida la que ejerce de soporte para la pantalla LCD) vienen de serie con este ventilador y radiadores para ser instalados.

## Montaje

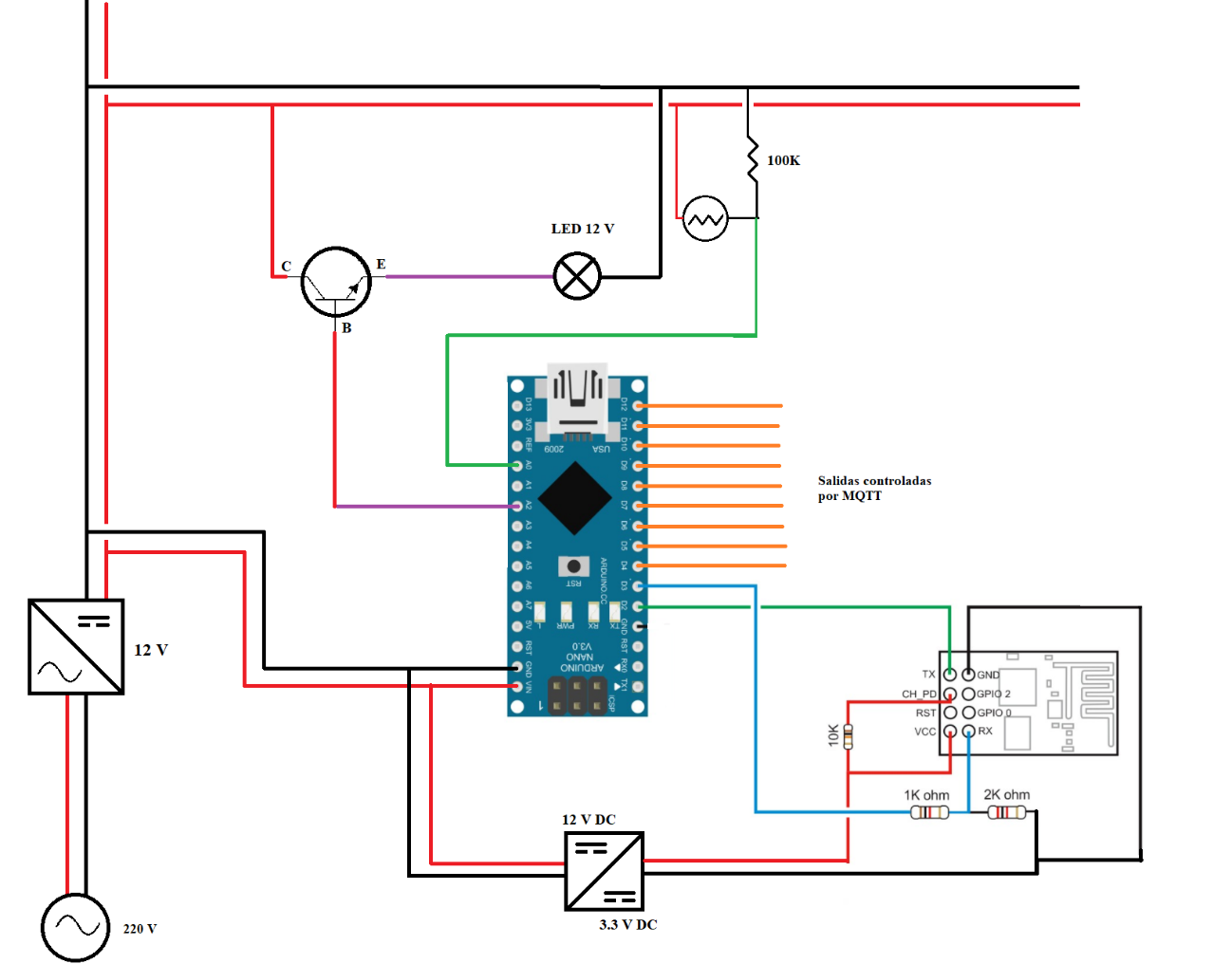
El montaje de los componentes del proyecto requiere que se realice la soldadura de los componentes a las PCBs. Esta soldadura, si no se realiza con extremo cuidado y atención puede llegar a dañar los componentes o a crear cortocircuitos. Especialmente el Arduino y el módulo ESP8266 pueden dañarse si la soldadura entra en contacto con la propia placa que contiene el microcontrolador

# Presupuesto

* Arduino Nano 10.99 € / 3 uds
* Raspberry Pi 4 4Gb con carcasa, tarjeta SD y accesorios 76.49 €
* Solamente la Raspberry Pi sin accesorios (52.00 €)
* Módulo ESP8266-01 10.81 € / 5uds
* Pantalla táctil LCD para Raspberry Pi (incluye carcasa + ventilación) 28.99 €
* Placas PCB (distintos tamaños) 11.49 €/ 30 uds
* Soldador, puntas, estaño y cables 11.99 €
* Transistores 12.00 € / 8 uds
* Transformador 220 AC – 12 DC 14.99 €
* Adaptador FTDI / SSL por USB 7.60 € / 3 uds
* Convertidores buck 12 V - 3.3V 3.72 € / 10 uds
* Convertidores buck 12V – 5V 5.23 € / 5 uds

# Planos

El montaje del proyecto es muy simple: solamente es necesario conectar el Arduino con el módulo ESP8266 de la siguiente manera:



# Anexos

## Programas

### Módulo ESP

Aquí se muestra el código programado en el módulo ESP8266-01. En este ejemplo, los temas de MQTT son raspihouse/espin/2 para la subscripción (identificando este módulo como el módulo 2 dentro de la red) y raspihouse/espout/2 para la publicación de mensajes.

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

const bool DEBUG = false;

// Wifi configuration

const char\* ssid = "RaspiHouse";

const char\* password = "MyIoTProject";

const char\* mqtt\_server = "192.168.4.1";

// MQTT Configuration

const String clientId = "ESP2";

const char\* inTopic = "raspihouse/espin/2";

const char\* outTopic = "raspihouse/espout/2";

// Client Objects

WiFiClient espClient;

PubSubClient client(espClient);

//Message buffers

#define MSG\_BUFFER\_SIZE (50)

char msg[MSG\_BUFFER\_SIZE];

int value = 0;

void setup\_wifi() {

delay(10);

WiFi.mode(WIFI\_STA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

}}

void callback(char\* topic, byte\* payload, unsigned int length) {

if(DEBUG){

Serial.print("Message arrived [");

Serial.print(topic);

Serial.print("] ");

}

payload[length] = '\0';

Serial.println((char\*)payload);

payload = '\0';}

void reconnect() {

// Loop until we're reconnected

while (!client.connected()) {

// Attempt to connect

if (client.connect(clientId.c\_str())) {

client.publish(outTopic, "Connected");

if(DEBUG){

Serial.println("Connected");

}

client.subscribe(inTopic);

}

else {

delay(5000);

}}}

void setup() {

pinMode(LED\_BUILTIN, OUTPUT);

Serial.begin(9600);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

if(DEBUG){

Serial.print("Setup coplete");

}

}

unsigned long lastBlink = millis();

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

if (Serial.available()){

String str = Serial.readString();

int len = str.length();

char payload[len];

str.toCharArray(payload, len);

client.publish(outTopic, payload);

}

// Blink the LED every 3 seconds

if (millis() - lastBlink > 1500) {

lastBlink = millis();

if(DEBUG){

Serial.print("...");

}

digitalWrite(LED\_BUILTIN, !digitalRead(LED\_BUILTIN));

}}

### Arduino Nano

En el siguiente código se ha programado el Arduino Nano para configurar sus salidas digitales como controladas mediante MQTT a través del módulo ESP8266, y además, empleando dos de sus puertos analógicos, se han conectado una fotorresistencia y una lámpara LED. Cuando la fotorresistencia detecte valores bajos de luz ambiente, el LED se encenderá hasta que la fotorresistencia vuelva a detectar luz durante más de 5 segundos. Cada vez que el LED se enciende o se apaga, el cambio se notifica por MQTT para poder reflejarlo en Node-Red.

#include <SoftwareSerial.h>

SoftwareSerial esp(2, 3); // RX, TX

unsigned long lastBlink = 0;

String state;

int lamp\_timer;

void setup() {

digitalWrite(LED\_BUILTIN, HIGH);

for(int i = 4; i < 12; i++){

pinMode(i, OUTPUT);

}

Serial.begin(9600);

esp.begin(9600);

digitalWrite(LED\_BUILTIN, LOW);

}

void loop (){

if (esp.available()){

int pin = esp.parseInt();

if(pin < 13 && pin > 1){

digitalWrite(pin, !digitalRead(pin));

state = digitalRead(pin) == HIGH ? " ON" : " OFF";

Serial.println(String(pin) + state);

esp.println(String(pin) + state);

}

}

// Blink the LED every 3 seconds

if (millis() - lastBlink > 1500) {

lastBlink = millis();

digitalWrite(LED\_BUILTIN, !digitalRead(LED\_BUILTIN));

}

// Insert custon code below

// For example, make synchronous calls to read a sensor

// Photorresistor connected to pin A0 triggers a lamp controlled by pin A1

if(digitalRead(A0) < 30 && digitalRead(A2) == LOW){

lamp\_timer = millis();

digitalWrite(A2, HIGH);

Serial.println("Lamp ON");

esp.println("Lamp ON");

}

if(millis() - lamp\_timer > 5000 && digitalRead(A2) >30 && digitalRead(A0) == HIGH){

digitalWrite(A2, LOW);

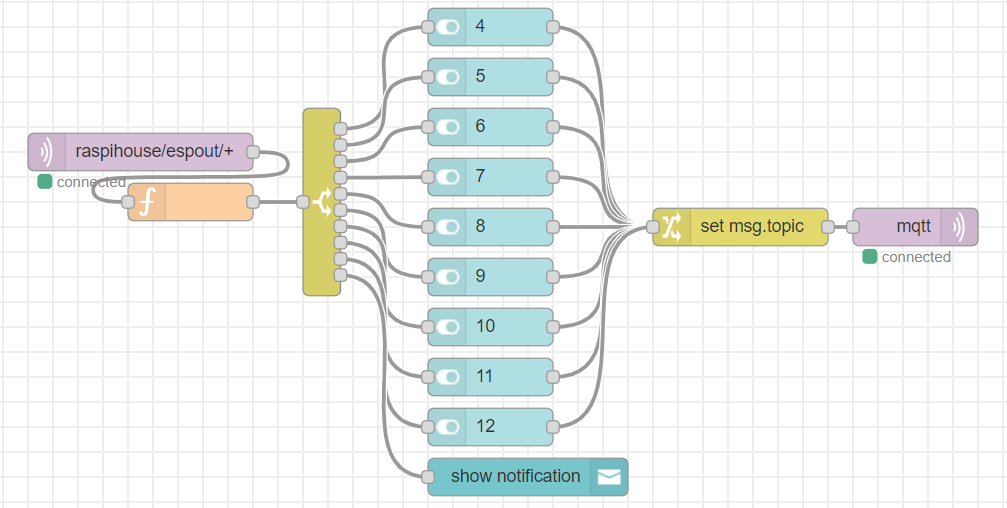
Serial.println("Lamp OFF");

esp.println("Lamp OFF");

}

}

### Node-Red



El código de Node-Red se almacena en format JSON (JavaScript Object Notation). Se incluye a continuación:

[{"id":"dc679b91.be9ab8","type":"tab","label":"Flow 1","disabled":false,"info":""},{"id":"85f9a98d.4abfa8","type":"mqtt out","z":"dc679b91.be9ab8","name":"","topic":"","qos":"","retain":"","broker":"ba42d1a.b19673","x":750,"y":300,"wires":[]},{"id":"8530674d.835968","type":"mqtt in","z":"dc679b91.be9ab8","name":"","topic":"raspihouse/espout/+","qos":"2","datatype":"auto","broker":"ba42d1a.b19673","x":130,"y":240,"wires":[["66d47305.b5869c"]]},{"id":"8c6ffe13.f36bf","type":"change","z":"dc679b91.be9ab8","name":"","rules":[{"t":"set","p":"topic","pt":"msg","to":"raspihouse/espin/2","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":610,"y":300,"wires":[["85f9a98d.4abfa8"]]},{"id":"81704e22.e1187","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"4","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"4","onvalueType":"num","onicon":"","oncolor":"","offvalue":"4","offvalueType":"num","officon":"","offcolor":"","x":410,"y":140,"wires":[["8c6ffe13.f36bf"]]},{"id":"c592b061.d3c2f","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"5","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"5","onvalueType":"num","onicon":"","oncolor":"","offvalue":"5","offvalueType":"num","officon":"","offcolor":"","x":410,"y":180,"wires":[["8c6ffe13.f36bf"]]},{"id":"82196b2.7506698","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"6","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"6","onvalueType":"num","onicon":"","oncolor":"","offvalue":"6","offvalueType":"num","officon":"","offcolor":"","x":410,"y":220,"wires":[["8c6ffe13.f36bf"]]},{"id":"a337cedd.65e5d","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"7","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"7","onvalueType":"num","onicon":"","oncolor":"","offvalue":"7","offvalueType":"num","officon":"","offcolor":"","x":410,"y":260,"wires":[["8c6ffe13.f36bf"]]},{"id":"3eeb1d2c.bb36a2","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"8","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"8","onvalueType":"num","onicon":"","oncolor":"","offvalue":"8","offvalueType":"num","officon":"","offcolor":"","x":410,"y":300,"wires":[["8c6ffe13.f36bf"]]},{"id":"3d0ea542.6761fa","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"9","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"9","onvalueType":"num","onicon":"","oncolor":"","offvalue":"9","offvalueType":"num","officon":"","offcolor":"","x":410,"y":340,"wires":[["8c6ffe13.f36bf"]]},{"id":"3d4a207e.21e9b","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"10","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"10","onvalueType":"num","onicon":"","oncolor":"","offvalue":"10","offvalueType":"num","officon":"","offcolor":"","x":410,"y":380,"wires":[["8c6ffe13.f36bf"]]},{"id":"6c1224d9.41316c","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"11","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"11","onvalueType":"num","onicon":"","oncolor":"","offvalue":"11","offvalueType":"num","officon":"","offcolor":"","x":410,"y":420,"wires":[["8c6ffe13.f36bf"]]},{"id":"58509412.b91f6c","type":"ui\_switch","z":"dc679b91.be9ab8","name":"","label":"12","tooltip":"","group":"c5fd532c.51f91","order":3,"width":0,"height":0,"passthru":false,"decouple":"false","topic":"","style":"","onvalue":"12","onvalueType":"num","onicon":"","oncolor":"","offvalue":"12","offvalueType":"num","officon":"","offcolor":"","x":410,"y":460,"wires":[["8c6ffe13.f36bf"]]},{"id":"66d47305.b5869c","type":"function","z":"dc679b91.be9ab8","name":"","func":"board = msg.topic.split('/')[2];\npayload = msg.payload.split('\\n');\n\nfor(var i in payload){\n noti = msg.payload.split(' ')\n msg.pin = noti[0];\n \n if(noti[1] == \"ON\"){msg.payload = true}\n else if(noti[1] == \"OFF\"){msg.payload = false}\n else{return null}\n \n \n if(noti[0] == 'Lamp'){\n msg.payload = \"Lamp is \" + noti [1];\n }\n node.send(msg);\n}","outputs":1,"noerr":0,"x":170,"y":280,"wires":[["355f27a3.3e1298"]]},{"id":"355f27a3.3e1298","type":"switch","z":"dc679b91.be9ab8","name":"","property":"pin","propertyType":"msg","rules":[{"t":"eq","v":"4","vt":"str"},{"t":"eq","v":"5","vt":"str"},{"t":"eq","v":"6","vt":"str"},{"t":"eq","v":"7","vt":"str"},{"t":"eq","v":"8","vt":"str"},{"t":"eq","v":"9","vt":"str"},{"t":"eq","v":"10","vt":"str"},{"t":"eq","v":"11","vt":"str"},{"t":"eq","v":"12","vt":"str"},{"t":"eq","v":"Lamp","vt":"str"}],"checkall":"true","repair":false,"outputs":10,"x":275,"y":280,"wires":[["81704e22.e1187"],["c592b061.d3c2f"],["82196b2.7506698"],["a337cedd.65e5d"],["3eeb1d2c.bb36a2"],["3d0ea542.6761fa"],["3d4a207e.21e9b"],["6c1224d9.41316c"],["58509412.b91f6c"],["1799bcd5.62f6d3"]],"l":false},{"id":"1799bcd5.62f6d3","type":"ui\_toast","z":"dc679b91.be9ab8","position":"bottom right","displayTime":"10","highlight":"","sendall":true,"outputs":0,"ok":"OK","cancel":"","raw":false,"topic":"","name":"","x":440,"y":500,"wires":[]},{"id":"ba42d1a.b19673","type":"mqtt-broker","z":"","name":"localhost","broker":"localhost","port":"1883","clientid":"nodered","usetls":false,"compatmode":false,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""},{"id":"c5fd532c.51f91","type":"ui\_group","z":"","name":"Default","tab":"404f7dbe.807134","order":1,"disp":true,"width":"6","collapse":false},{"id":"404f7dbe.807134","type":"ui\_tab","z":"","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]

## Alternativa al Punto de Acceso en Raspberry Pi

La red Wifi generada por la Raspberry Pi no presenta un alcance muy grande. Si se desea desplegar este proyecto a distancias mayores que una habitación, es necesario configurar la raspberry para que se conecte a la red Wifi doméstica con una dirección IP estática. Para ello, la manera más sencilla es conectando la Raspberry a un monitor, ratón y teclado. Una vez encendida, en la parte superior del escritorio se encuentran los ajustes de red. Simplemente es necesario seleccionar la interfaz que se esté empleando para la conexión wifi (wlan0) o cableada (eth0) y en el campo de dirección IP introducir una IP válida y libre de la red 192.168.0.0/16 (Por ejemplo 192.168.1.123).

Hecho esto y habiendo comprobado que, tras un reinicio, la Raspberry efectúa correctamente la conexión a Internet, es necesario modificar el programa a introducir en el módulo ESP8266 de la forma que se indica a continuación:

PUNTO DE ACCESO RED WIFI DOMÉSTICA

// Wifi configuration // Wifi configuration

const char\* ssid = "RaspiHouse"; const char\* ssid = “<SSID>”;

const char\* password = "MyIoTProject"; const char\* password = “<Password>”;

const char\* mqtt\_server = "192.168.4.1"; const char\* mqtt\_server = "192.168.1.123";

# Bibliografía

Arduino Project Hub: https://create.arduino.cc/projecthub/

Things Board IoT Platform: https://thingsboard.io/docs/samples/arduino/

Arduino User Forum: https://arduino.stackexchange.com/

Conexión y Programación de ESP8266: https://iot-guider.com/esp8266-nodemcu/program-esp8266-arduino-using-ftdi/

Librería WifiESP para ESP8266: https://github.com/bportaluri/WiFiEsp

Librería PubSubClient para MQTT: https://github.com/knolleary/pubsubclient/

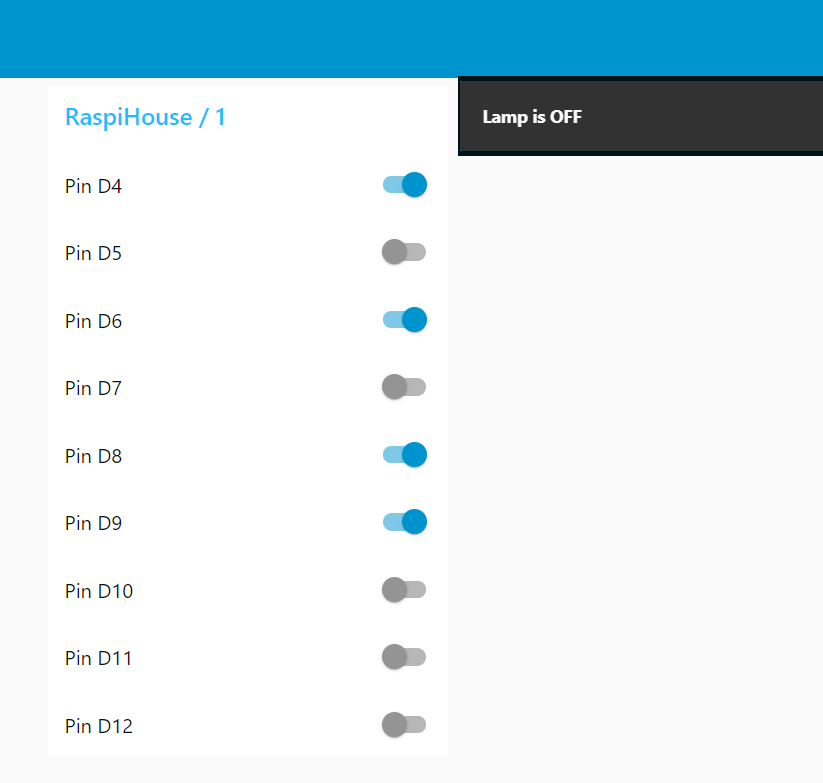
Manual módulo ESP8266-01: https://www.espressif.com/sites/default/files/documentation/4a-esp8266\_at\_instruction\_set\_en.pdf

Stack Overflow. Comunidad de programadores: https://stackoverflow.com/

Node-Red: https://nodered.org/docs

# Conclusión

Con este proyecto se ha logrado diseñar un módulo de domótica versátil y escalable, creando una red de IoT a la que se pueden conectar un número indefinido de dispositivos adicionales. Gracias a la utilidad de Node-Red, puede programarse lógica y funciones adicionales sin necesidad de reprogramar el Arduino o el módulo ESP.

La aplicación final toma la forma de uno o más módulos de domótica siendo controlados por la misma instancia de Node-Red desde cualquier dispositivo con conectividad Wifi (teléfono móvil, Tablet PC, Raspberry Pi…) y a través de una interfaz sencilla y personalizable.

El prototipo cumple con los objetivos de escalabilidad y versatilidad propuestos. Cada módulo cuenta con 9 pines de entrada/salida digitales completamente controlables desde la interfaz de aplicación, así como de 8 puertos de entrada/salida analógica que permiten añadir funcionalidades programadas en el Arduino. Esta asignación de puertos digitales para la aplicación y analógicos para funciones programadas permite que no haya interferencias o choques entre ambas.