



UNIVERSIDAD TECNOLÓGICA NACIONAL  
Facultad Regional Reconquista

Informe de Avance  
Autor: Ing. Talijancic Iván

**PID Tutorado N° 4270 - Regulador de Caudal Telegestionado**

---

---

---

# Índice

<b>1. Introducción:</b>	<b>1</b>
1.1. Objeto del Informe . . . . .	1
<b>2. Estado al inicio del ciclo de trabajo</b>	<b>2</b>
<b>3. Cambio de enfoque en la Solución</b>	<b>3</b>
3.1. Temáticas de investigación abordadas . . . . .	3
3.1.1. IoT . . . . .	3
3.1.2. OpenSource: Código Abierto . . . . .	4
3.1.3. Raspberry Pi . . . . .	4
3.1.4. MQTT . . . . .	4
3.1.5. Módulo Wi-Fi ESP8266 . . . . .	4
3.2. Alternativas Planteadas . . . . .	6
3.2.1. PIC-ModBus-Raspberry-MQTT-Cloud . . . . .	6
3.2.2. ESP-Raspberry(Gateway)-Server(Cloud) . . . . .	7
3.2.3. ESP-Raspberry . . . . .	8
<b>4. Publicaciones</b>	<b>9</b>
<b>5. Liquidación de Fondos</b>	<b>10</b>
<b>6. Conclusiones</b>	<b>11</b>

# **1. Introducción:**

## **1.1. Objeto del Informe**

El presente informe detalla a nivel descriptivo las tareas realizadas en el marco de elaboración del PID Tutorado N° 4270 - “Regulador de Caudal Telegestionado”, durante el período de trabajo 2017 - 2018.

## 2. Estado al inicio del ciclo de trabajo

Como se detalló en el anterior informe de avance, al iniciar este ciclo (año) de trabajo, a nivel electrónico el proyecto estaba en primera instancia resuelto.

Esta primera solución se basaba en una comunicación sobre radiofrecuencia en su capa física entre la compuerta (microcontrolador) y el servidor (PC).

Esta solución demostró ser funcional, a nivel experimental, mediante diversos ensayos realizados en el laboratorio de la facultad, en donde se logró corroborar que la alternativa planteada era factible, tanto a nivel de hardware, como de software embebido y software de PC.

En el siguiente esquema se puede visualizar de forma gráfica la topología de red planteada en esta primera solución y las tecnologías sobre las cuales la misma recaía:

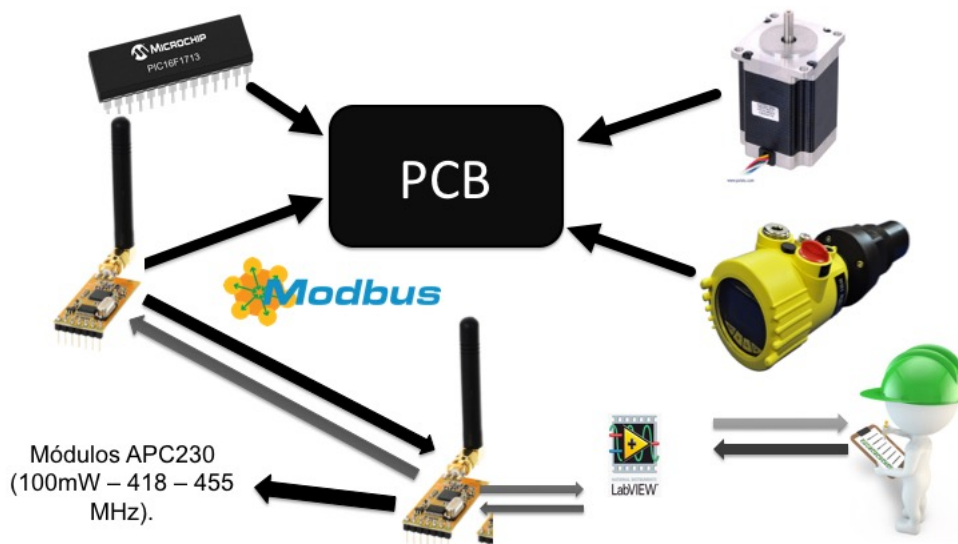


Figura 1: Esquema General, 1ra Solución

Como podemos ver en la imagen anterior, en la topología de red planteada para la 1er solución propuesta, se hizo uso de los siguientes elementos y tecnologías:

- PCB:** Una placa de circuito impreso (PCB), en la cual se aloja todo el hardware necesario para accionar la compuerta y captar los sensores que se precisen, en primera instancia un sensor de nivel. Así como contener al microcontrolador que viene a ser las veces de cerebro de cada compuerta.
- Módulos tranceptores ACP230**, que son los encargados de materializar la capa física de comunicación entre las compuertas y el servidor.
- Protocolo industrial de bus de campo ModBus**, utilizado para implementar la comunicación entre los micros y las PCs, servidores.
- SCADA programado íntegramente en LabVIEW** que se encarga de correr en background un OPC (Open Protocol Control) para gestionar la comunicación ModBus y al mismo tiempo sirve interfaz de usuario del sistema. Mediante la cual el operario puede interactuar en tiempo real con el sistema.

### 3. Cambio de enfoque en la Solución

Con el inminente advenimiento del Internet de las Cosas (IoT - Por sus siglas en inglés “Internet of Things”), resultó imperativo, adaptar nuestro enfoque de solución a la problemática abordada, pensando en que debíamos transformar a las compuertas telegestionadas para riego superficial en un dispositivo más dentro del “IoT”.

Con este nuevo enfoque en mente se trabajó, planteando diferentes alternativas, para lo cuál en primera instancia se realizó un trabajo de investigación en relación a las siguientes temáticas:

#### 3.1. Temáticas de investigación abordadas

##### 3.1.1. IoT

Si bien hoy por hoy, TODOS estamos familiarizados con el término IoT, a la hora de asociar un concepto a esta palabra muchos de nosotros tendremos dificultades. Es por ello que a continuación se describe en pocas palabras de que se trata el **Internet de las Cosas (IoT)**.

*El IoT está transformando la forma de interactuar entre individuos y entre ellos y su entorno. Se trata de un conjunto de elementos interconectados, tales como: actuadores, sensores y dispositivos a redes que habilitan el almacenamiento, intercambio y análisis de la información generada Ver: [4] y [3].*

*En palabras sencillas podemos pensar en el IoT como una red de dispositivos que pueden clasificarse, según su funcionalidad dentro de la red en:*

- **Cosas:** que son ni más ni menos que lo elementos finales en la red, más conocidos como *actuadores o sensores*.
- **Gateways:** Dispositivos concentradores de datos, a los cuales se comunican un conjunto de “*cosas*” y estos son los encargados luego, de enrutar estos datos a la nube.
- **Aplicaciones:** Son la etapa final dentro de la red y las encargadas de la interacción hombre-IoT.

En la Figura: 2 IoT - Arquitectura, podemos ver un esquema de lo anteriormente explicado.

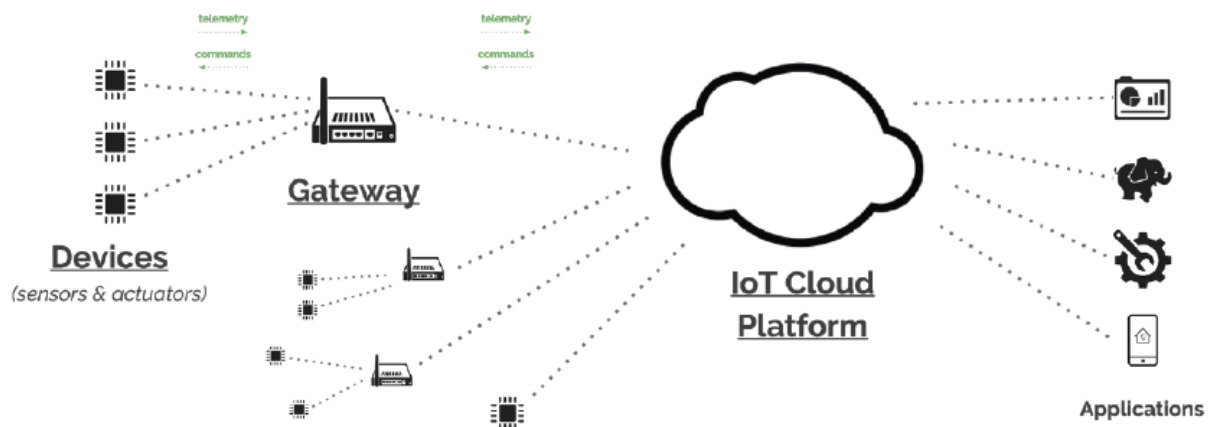


Figura 2: IoT - Arquitectura

### 3.1.2. OpenSource: Código Abierto

La rápida evolución de la tecnología implicada el IoT, se ve impulsada por la industria del software OpenSource (o, también conocido como software de código abierto) que se posiciona como la principal proveedora de software para éstos fines.

Es por ello que resultó evidente la necesidad de investigar al respecto del mundo del OpenSource, para saber que herramientas tenemos a disposición a la hora de materializar soluciones IoT. El propósito en estas investigaciones, fue interiorizarnos acerca de los requerimientos y arquitecturas demandadas por las soluciones de IoT. De modo similar al Stack LAMP (Linux/Apache HTTP Server /MySQL/PHP) que se encuentra ampliamente difundido y domina las infraestructuras web existentes.

### 3.1.3. Raspberry Pi

Raspberry Pi es un computador de placa reducida, computador de placa única o computador de placa simple (SBC - Simple Board Computer) de bajo costo desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación.

Esta fue la plataforma elegida en el proyecto, para hacer investigaciones y aplicaciones al respecto del mundo OpenSource. Así como para el diseño del gateway que necesitamos en la solución IoT a la que se pretende arribar.

El uso de las mismas nos permitió formarnos en el manejo de linux embebido, así como en la implementación de servicios de código abierto que corren sobre linux y nos permiten gestionar un sin fin de aplicaciones útiles, tales como: servidores web, dispositivos de almacenamiento en red, brokers de comunicación, Servidores OPCs, entre otros.

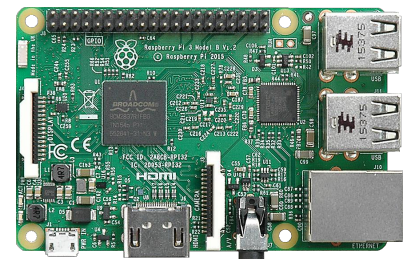


Figura 3: Raspberry Pi 3

### 3.1.4. MQTT

MQTT (por sus siglas en inglés “Message Queuing Telemetry Transport”) es un protocolo de comunicación basado en la publicación-suscripción. Trabaja en la capa superior del protocolo TCP/IP . Esta diseñado para conexiones remotas donde el ancho de banda es limitado, y para correr sobre dispositivos con escasos recursos de hardware. La topología de red propia del protocolo, basada en la suscripción-publicación de mensajes, requiere de un broker para funcionar. Este último vendría a ser el servidor en la comunicación, es el encargado de gestionar los usuarios, suscriptores y tópicos publicados.

En 2013 IBM lazo MQTT v3.1 y desde entonces, contando con el aval de IBM y toda su estructura, se plantea y posiciona como el protocolo standard para aplicaciones del IoT.

Su aplicación en diversas plataformas se encuentra hoy por hoy impulsada por el echo de que existen gran cantidad de librerías de código abierto que lo hacen portable a diferentes arquitecturas de electrónica embebida, como ser ARM, MIPs y otras, con implementaciones en diferentes lenguajes de programación, como ser: C, C++, Java.

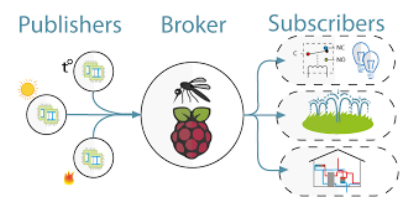


Figura 4: Esquema Red MQTT

### 3.1.5. Módulo Wi-Fi ESP8266

El módulo ESP8266 es básicamente un modulo Wi-Fi que integra en un solo chip un microcontrolador de 32bits de propósito general y altas prestaciones, así como el software y hardware

necesario para proveer conectividad Wi-Fi. En los últimos años este módulo a tenido una gran difusión por su uso en aplicaciones, tanto industrial, como a nivel de hobbistas y entusiastas de la electrónica programable y los sistemas embebidos.

Su aplico uso y reducido coste, así como su altas prestaciones lo posicionan como la alter-nativa ideal de mercado a la hora de encarar un proyecto de electrónica embebida que requiera conectividad inalámbrica y donde las redes de Wi-Fi satisfagan las necesidades del proyecto.

En la siguiente imagen podemos ver el módulo del que estamos hablando:

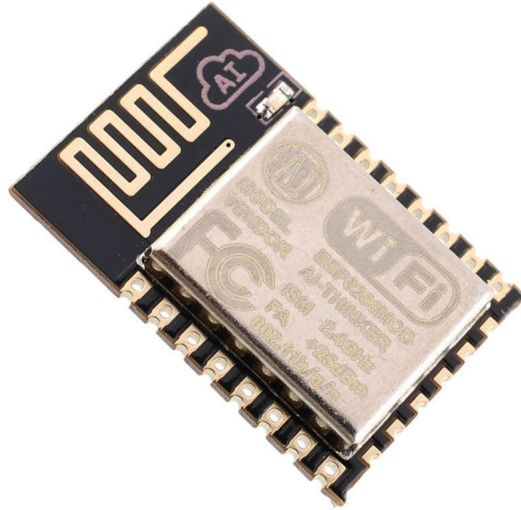


Figura 5: Módulo ESP

Este módulo puede utilizarse en forma conjunta (acoplado) a un microcontrolador, que se comunique con este mediante un puerto serie y comandos AT. O puede optarse por programar el microcontrolador interno con el que mismo cuenta para realizar diferentes tareas de propósito general.

## 3.2. Alternativas Planteadas

### 3.2.1. PIC-ModBus-Raspberry-MQTT-Cloud

Fue la 1er alternativa que se contempló y la misma consistía en una red, constituida por los siguientes elementos:

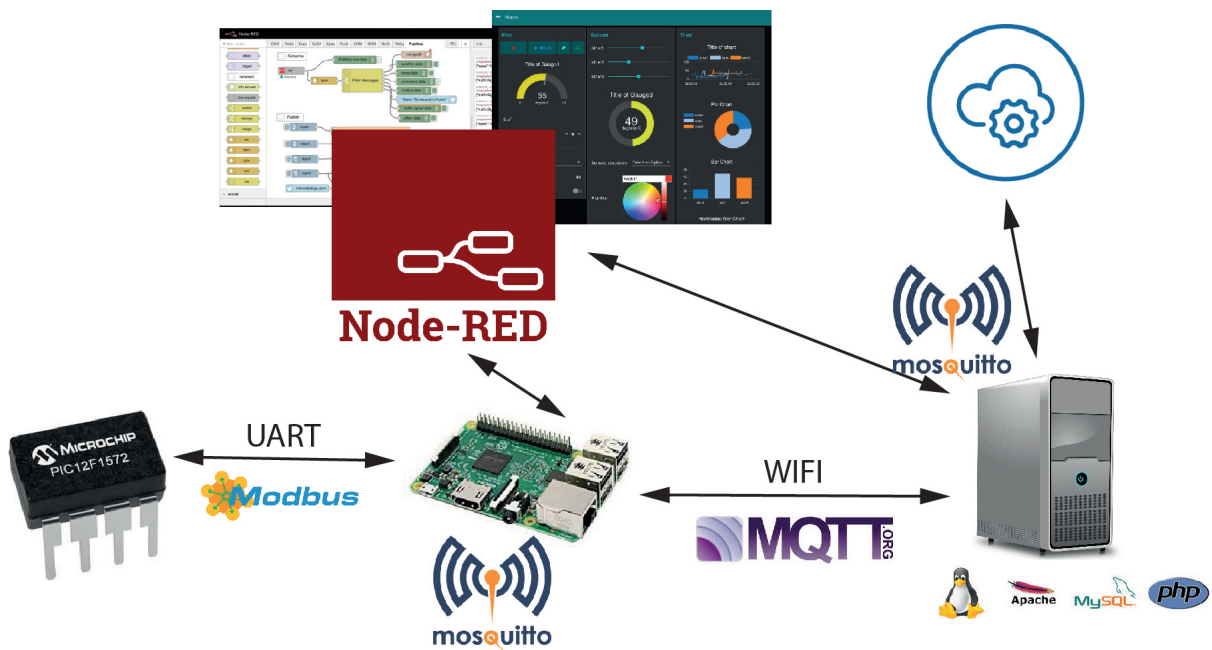


Figura 6: PIC-Raspberry-ServerPC

En este primer planteo, se utilizó un microcontrolador de la firma Microchip más precisamente el modelo PIC12F1572 (Ver: [7], para más información), para hacer la lectura de los sensores y posterior comunicación con las Raspberry, esto se decidió así por una cuestión de practicidad y disminución del tiempo de desarrollo para hacer pruebas. Debido a que ya se venía trabajando con estos micros y gran parte del software necesario estaba ya desarrollado. Es necesario aclarar que la Raspberry por sí sola podía realizar estas tareas, pero nuestra falta de experiencia en su uso y programación nos hizo en primera instancia decantar por el uso de un micro al que se le asignaron las realización de estas tareas “básicas”.

Para la implementación de este esquema se trabajó en el diseño y desarrollo de los siguientes elementos:

- **Software Embebido:** Desarrollo de librería para la implementación del protocolo ModBus serie RTU, en microcontroladores PIC de 8 bits. Y desarrollo de librerías para lectura de sensores varios, tanto analógicos como digitales.
- **Desarrollo de un servidor ModBus serie**, implementado sobre la Raspberry, corriendo sobre linux embebido y con la particularidad de poder servir datos en tiempo real mediante una interfaz web. Para este desarrollo se hizo uso de la plataforma de código abierto, desarrollada por IBM para aplicaciones del IoT y que lleva el nombre de **Node-Red**. La misma es una plataforma de desarrollo que provee una interfaz gráfica de programación, muy intuitiva (similar al LabVIEW) y con la ventaja que nos permite servir una interfaz de usuario a través de un simple navegador web.
- **Implementación de cliente MQTT** en la Raspberry, lo cual se realizó haciendo uso de la librería OpenSource mosquitto.
- **Implementación de un broker MQTT en servidor virtual**, para poder servir los datos a la nube.



Las pruebas funcionales de esta alternativa se hicieron leyendo datos de diferentes sensores de variables atmosféricas (temperatura, humedad relativa, presión, etc.), y cargando los datos en la nube para su acceso mediante diferentes medios. El objeto de las pruebas realizadas, fue probar la estabilidad de las diferentes comunicaciones que en la red intervienen.

### 3.2.2. ESP-Raspberry(Gateway)-Server(Cloud)

Con la inclusión del módulo Wi-Fi ESP8266 (Ver: [5] y [6]) al diseño de nuestra solución se simplifico bastante el panorama, ya que el mismo nos posibilitaba la conectividad inalámbrica entre el elemento final ( “cosa”, como ser sensor o actuador) y el gateway.

Esta nueva funcionalidad/posibilidad dentro de la topología de la red nos permitió diseñar, desarrollar y probar diferentes configuraciones, dentro de las cuales, la primera en estudio, desarrollo y prueba fue la que podemos ver en la siguiente imagen:

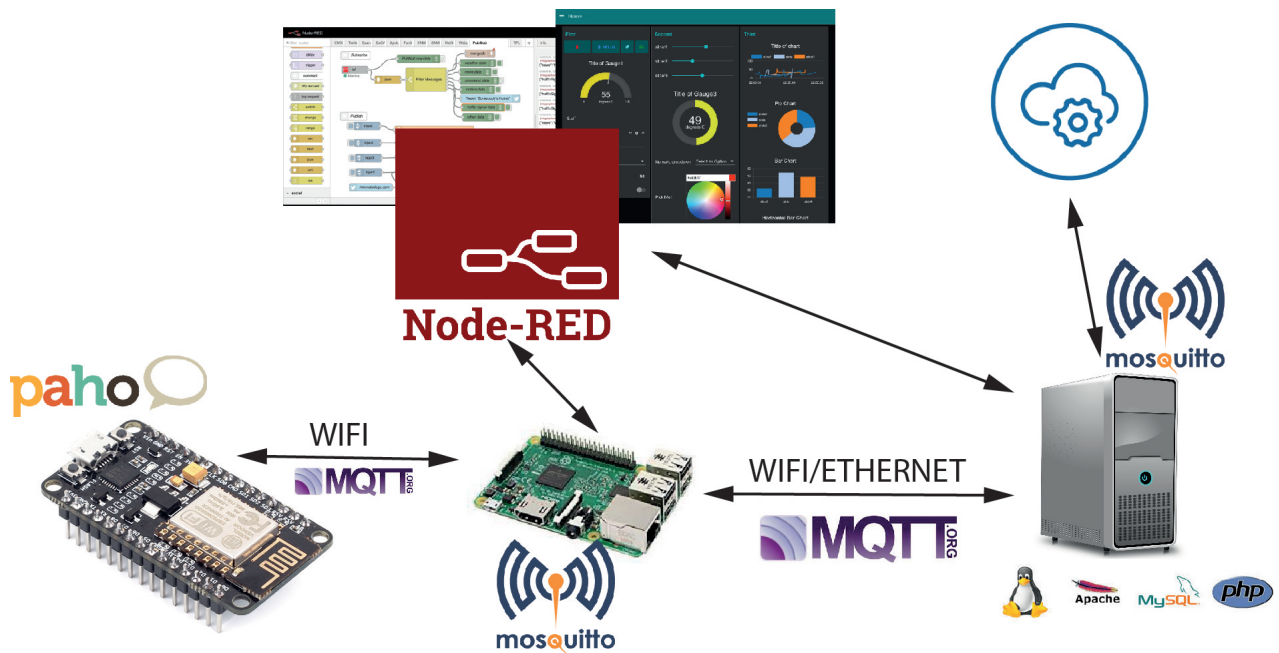


Figura 7: ESP-Raspberry-ServerPC

Para el diseño y desarrollo de esta alternativa se trabajo sobre:

- Estudio e investigación sobre el módulo wi-fi ESP8266. Funcionamiento, modos, programación, plataformas y lenguajes. Metodología de trabajo.
- Sistemas Operativos en Tiempo Real (RTOS - Por sus siglas en inglés de Real Time Operation System), el mismo se aplico en el desarrollo del software embebido que corre en el ESP8266, ya que la complejidad en la cantidad de tareas que el mismo debe realizar hacia prácticamente imposible su desarrollo sin valernos de las grandes ventajas que un sistema operativo en tiempo real nos brinda al respecto cuando de sistemas multi-tasking se trata. Con respecto a este apartado el RTOS seleccionado, dentro de la oferta opensource que existe fue FreeRTOS, ya que el mismo es gratuito y de gran difusión por lo cual es muy fácil encontrar información y ejemplos de uso. (Ver: [2] y [1])
- Con respecto a las plataforma para servicios de protocolos de comunicación, así como interfaces de usuario, se uso lo mismo que en la anterior alternativa.

Es interesante destacar que como podemos ver, tanto la Raspberry como el servidor PC, son capaces de servir interfaces de usuario, mediante un navegador web, la diferencia es el

alcance que tienen estas GUI (Grafic User Interface). Mientras la Raspberry sirve un SCADA del alcance en la red local, el servidor PC es accesible desde cualquier red externa.

Esto se puso a prueba generando un servidor virtual en el servidor de la facultad. Dicho servidor se encuentra aún corriendo y esto puede corroborarse visitando el siguiente link: [190.114.205.143:1880/ui](http://190.114.205.143:1880/ui) en un navegador web cualquiera, ya sea desde una PC o un smartpho-  
ne.

### 3.2.3. ESP-Raspberry

Esta fue la ultima de las configuraciones estudiadas y ensayadas. La misma presenta menos elementos en el diseño de la red IoT y es por ende interesante para aplicaciones sencillas, en donde no tengamos una gran cantidad de usuario en simultáneos. Solo esta constituida por las cosas, manifestadas en los módulos ESP, que pueden hacer las veces de sensores o actuadores y él/los gateway/servidor al mismo tiempo materializados en la Raspberry. Es el potencial de estas pequeñas computadoras los que nos permite diseñar configuraciones de este estilo.

Debemos prestar especial atención en este apartado a la genialidad que se esconde detrás de la Raspberry, una computadora de simple placa y reducido costo (\$2.000 a Marzo de 2018) que nos brinda la posibilidad de armarnos nuestro propio servidor de manera sencilla, y recurriendo a herramientas de código abierto.

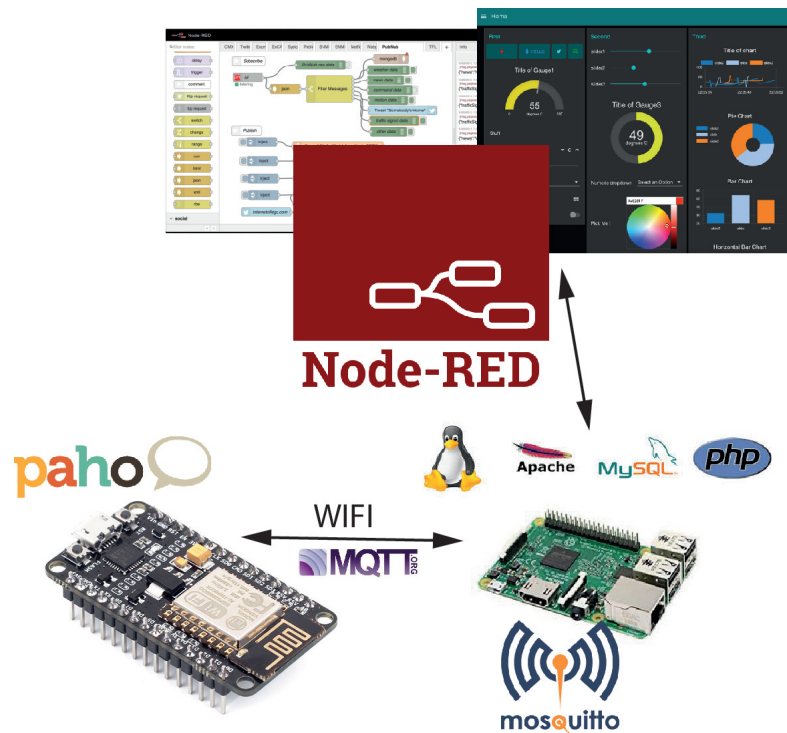


Figura 8: ESP-Raspberry

En este caso particular sobre lo que mayoritariamente se trabajó fue en la investigación, diseño y desarrollo del servidor LAMP (Linux/Apache/MySQL/PHP) que corre sobre la Raspberry y hace las veces de broker MQTT al mismo tiempo que logea información en base de datos y nos permite acceder a los mismo a través de la nube, mediante una interfaz de usuario desarrollada íntegramente en Node-Red.

## 4. Publicaciones

Resultados obtenidos de las investigaciones y trabajos realizados en el marco de realización de este proyecto PID fueron publicados en diferentes certámenes de divulgación científica como ser:

- **JIT 2017:** La cual se realizó en Reconquista y en donde se obtuvo premio al mejor poster en la categoría.
- **Congreso de Smart Cities:** Realizado en la ciudad de Reconquista, por iniciativa del municipio. El mismo fue de gran despliegue y se contó con la participación de empresas y personas muy influyentes en las temáticas referidas.

En el corriente año 2018 se esperan presentar nuevos resultados, en congresos a fines a la temática, y en el JIT 2018. También se hará la carga de todas las publicaciones derivadas de las investigaciones en el RIA (Repositorio Institucional Abierto).

## 5. Liquidación de Fondos

Con respecto a la liquidación de fondos con los que contaba el proyecto, se hicieron las siguientes compras:

Ítem N°	Cantidad	Descripción
1	2	Monitor LED 19" HD (Con conexión HDMI)
2	4	Kit teclado y mouse inalámbrico
3	2	Monitor LED 24" Full HD (Con conexión HDMI)
4	3	Tarjeta de memoria micro SD - Clase 10 32Gb
5	2	Kit Raspberry Pi 3
6	1	Monitor LED Curvo 27"
7	5	Bobinas de cable conductor de $0,75mm^2$
8	1	Fuente de alimentación de laboratorio 0-30 V 0-5A
9	1	Fuente de alimentación de laboratorio, regulada digital 30V 6A 3Channels
10	1	Multímetro de banco para laboratorio
11	3	Conversor VGA-HDMI

Tabla 1: Tabla de Compras Realizadas 2017

Este conjunto de elementos adquiridos suman en conjunto un total de \$74.240,74.

## 6. Conclusiones

Con respecto al trabajo realizado en el periodo 2017-2018, correspondiente a la elaboración del **PID Tutorado 1270** - *“Regulador de Caudal Telegestionado”*, se puede concluir:

- El cambio de enfoque, pensando en el regulador de caudal telegestionado como un dispositivo más dentro del mundo del IoT nos permitió ampliar nuestro rango de acción, obligándonos a encarar nuevas temáticas de investigación y desarrollar nuevas competencias en el manejo de herramientas informáticas así como de recursos.
- El inherente perfil multidisciplinario que el IoT presenta, es un gran desafío desde el punto de vista técnico e intelectual, por ende el trabajo en este área de aplicación del conocimiento agregó gran valor a los implicados en el proyecto.
- Plantear diferentes alternativas de solución, diseñarlas y elaborarlas para su posterior ensayo fue de gran importancia y agregan valor al trabajo realizado, ya que el mismo no consistió en una mera recopilación bibliográfica para su posterior investigación, sino que además de esto permitió corroborar en la práctica cuestiones conceptuales y de diseño, así como afianzar conocimientos y desarrollar competencias en campos nuevos de aplicación.
- Con fondos del proyecto se compraron gran cantidad de insumos, así como herramientas de laboratorio y recursos informáticos que permiten enriquecer las instalaciones de la facultad, así como brindar nuevas y variadas herramientas didácticas para otorgar una experiencia educativa de mayor calidad a nuestros estudiantes.

## Referencias

- [1] BARRY, R. *Mastering the FreeRTOS Real Time Kernel A Hands-On Tutorial Guide*, pre-release 161204 ed. Real Time Enginners Ltd., 2016.
- [2] BARRY, R. *The FreeRTOS Reference Manual API Functions and Configuration Options*, reference manual for freertos version 10.0.0 issue 1 ed. Real Time Engineers Ltd, 2017.
- [3] ECLIPSE IOT WORKING GROUP. Open source software for industry 4.0. *iot.eclipse.org* (2017).
- [4] ECLIPSE IOT WORKING GROUP. The three software stacks required for iot architectures. *iot.eclipse.org* (2017).
- [5] ESPRESSIF. <https://www.espressif.com/en/products/hardware/esp-wroom-02/overview>.
- [6] ESPRESSIF. *ESP-WROOM-02 Datasheet*, v2.5 ed. Espressif, 2017.
- [7] MICROCHIP. *PIC12F1572 Datasheet*. Microchip, 2018.