



**UNIVERSIDAD DEL AZUAY**  
**FACULTAD DE CIENCIA Y TECNOLOGÍA**  
**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**Diseño e Implementación de un Sistema SCADA para el  
Control de Riego Mediante un Dispositivo Móvil**

**Trabajo de graduación previo a la obtención del título de:  
INGENIERO ELECTRÓNICO**

**Autores:**  
**CHRISTIAN PATRICIO FERNÁNDEZ DE CÓRDOVA Á.**  
**DANIEL HUMBERTO VANEGAS V.**

**Director:**  
**ING. SIMÓN BOLÍVAR MÉNDEZ RENGEL**

**CUENCA - ECUADOR**  
**2018**

**DEDICATORIA:**

Dedico este trabajo de manera especial a mis Padres, ya que ellos fueron el principal cimiento para la construcción de mi vida profesional, los que me apoyaron con el sustento económico para poder estudiar y llegar a culminar una etapa más.

A mi esposa Silvia por todo su apoyo y por ser mi fuente de inspiración y motivación para poder superarme cada día más, por sus palabras de aliento y perseverancia para que cumpla con mis ideales.

**Christian Patricio Fernández de Córdoba Á.**

**DEDICATORIA:**

Este trabajo va dedicado a mi familia, muy especial a mis padres César y Marianela que han sido la fuerza y motivación para mí, gracias al apoyo de ellos en todo momento para que puede terminar mi carrera como profesional.

A mis hermanos q han sido un apoyo importante en mi estudio.

**Daniel Humberto Vanegas V.**

### **AGRADECIMIENTOS:**

Agradecemos primero a Dios por darnos la oportunidad de estudiar y llegar a ser unos profesionales. A la Universidad del Azuay y a sus docentes por los conocimientos compartidos para formarnos como profesionales, y de manera muy especial a nuestro director de tesis Ing. Bolívar Méndez Rengel por guiarnos durante el proceso de titulación.

## INDICE DE CONTENIDOS:

<b>DEDICATORIA:</b> .....	<b>II</b>
<b>AGRADECIMIENTOS:</b> .....	<b>IV</b>
<b>INDICE DE CONTENIDOS:</b> .....	<b>V</b>
<b>INDICE DE FIGURAS:</b> .....	<b>VIII</b>
<b>INDICE DE TABLAS:</b> .....	<b>XI</b>
<b>RESUMEN</b> .....	<b>XII</b>
<b>ABSTRACT</b> .....	<b>XIII</b>
<b>INTRODUCCION</b> .....	<b>1</b>
<b>CAPITULO 1</b> .....	<b>3</b>
<b>1.SISTEMAS DE RIEGO Y AUTOMATIZACIÓN</b> .....	<b>3</b>
1.1    Introducción .....	3
1.2    Sistema de Riego.....	3
1.2.1    Riego por Superficie .....	3
1.2.2    Presurizados .....	6
1.3    Automatización y Aplicaciones. ....	10
1.3.1    Sistema de Riego Instalado por la Compañía Waterman.....	10
1.3.2    Sistema de Control Remoto de Riego por Telefonía Móvil.....	12
1.3.3    Sistema de control basado en el sistema MyScada e Information Server	12
12	
<b>CAPITULO 2</b> .....	<b>14</b>
<b>2 SISTEMAS SCADA Y COMUNICACIONES INDUSTRIALES</b> .....	<b>14</b>
2.1    Introducción .....	14
2.2    SCADA .....	14
2.2.1    Objetivo de un sistema SCADA .....	14
2.2.2    Prestaciones.....	14
2.2.3    Ventajas.....	15
2.2.4    El Entorno .....	16
2.3    Criterios de selección y diseño.....	17
2.3.1    Disponibilidad.....	18
2.3.2    Robustez.....	19
2.3.3    Seguridad .....	19
2.3.4    Prestaciones.....	20

2.3.5	Mantenimiento .....	20
2.3.6	Escalabilidad .....	20
2.4	Arquitectura de un sistema SCADA .....	20
2.4.2	Software .....	24
2.5	Comunicaciones Industriales .....	25
2.5.1	Sistema de transporte de señal .....	26
2.5.2	Sistema de transmisión de señal.....	27
2.6	Protocolos de comunicación .....	28
2.6.1	Protocolo OSI.....	29
2.6.2	Protocolo TCP/IP .....	30
2.7	Tipos de red según forma (topológica) .....	32
2.7.1	Anillo .....	33
2.7.2	Estrella.....	34
2.7.3	Bus .....	34
2.7.4	Árbol .....	35
2.7.5	Red .....	36
2.8	Pirámide de automatización .....	36
2.8.1	Nivel de Gestión.....	37
2.8.2	Nivel de célula.....	38
2.8.3	Nivel de campo .....	38
2.8.4	Nivel actuador/sensor.....	38
<b>CAPITULO 3.....</b>		<b>39</b>
<b>3. DISEÑO DEL SISTEMA SCADA PARA EL CONTROL DE RIEGO.....</b>		<b>39</b>
3.1	SOFTWARE .....	39
3.1.1	Blynk.....	39
3.1.2	Arduino .....	40
3.2	Hardware .....	42
3.2.1	Arduino .....	42
3.3	Sensores .....	48
3.3.1	Sensor de Humedad FC-28 .....	49
3.3.2	Sensor de temperatura DS18b20.....	50
3.4	Actuador.....	51
3.4.1	Electroválvula .....	51
3.4.2	Aspersor .....	53

3.5	Diseño del proyecto.....	54
3.5.1	Servidor local .....	54
3.5.2	Servidor Blynk Internet.....	63
3.5.3	Programación de control de riego. ....	64
<b>CAPITULO 4.....</b>		<b>70</b>
<b>4. IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA DE RIEGO. ....</b>		<b>70</b>
<b>INTRODUCCIÓN: .....</b>		<b>70</b>
4.1	Implementación.....	70
4.1.1	Sistema de riego por aspersión.....	72
4.1.2	Sistema de riego por goteo .....	76
4.1.3	Instalación de Sensores .....	77
4.2	Pruebas del Sistema.....	79
4.2.1	Pruebas de Accionamiento.....	79
4.2.2	Pruebas de lecturas de Sensores.....	81
4.3	Operaciones del funcionamiento del sistema.....	85
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>87</b>
CONCLUSIONES. ....		87
RECOMENDACIONES. ....		87
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>89</b>
<b>ANEXOS .....</b>		<b>91</b>

**INDICE DE FIGURAS:**

Figura 1.1: Riego por surco.....	4
Figura 1.2: Riego por manto .....	5
Figura 1.3: Riego por tuberías.....	5
Figura 1.4: Riego por aspersión fijo.....	7
Figura 1.5: Riego por aspersión móvil.....	8
Figura 1.6: Riego por micro aspersión.....	9
Figura 1.7: Riego por goteo .....	10
Figura 1.8: Visualización del riego en el Parque Joseph Cooper.....	11
Figura 2.1: Embotelladora de Coca Cola en Filipinas .....	15
Figura 2.2: Presenta una estructura corporativa dividida en tres niveles que engloban funciones principales.....	17
Figura 2.3: Principio de Redundancia.....	19
Figura 2.4: Estructura básica de un sistema de supervisión y mando.....	21
Figura 2.5: Una aplicación HMI integrado con los programas de LabVIEW 8.5 DSC Módulo themultisegmented.....	22
Figura 2.6: Esquema de conexión MTU .....	22
Figura 2.7: RTU .....	23
Figura 2.8: Control Centralizado .....	25
Figura 2.9: Control Distribuido.....	26
Figura 2.10: Bucle analógico de corriente .....	27
Figura 2.11: Componentes de un enlace de datos .....	28
Figura 2.12: Capas y unidades de intercambio del modelo OSI.....	30
Figura 2.13: Comparación del Modelos OSI con TCP/IP .....	31
Figura 2.14: Topología de anillo.....	33
Figura 2.15: Topología Estrella.....	34
Figura 2.16: Topología tipo Bus .....	35
Figura 2.17: Topología tipo Árbol .....	35
Figura 2.18: Topología tipo Red .....	36
Figura 2.19: Pirámide (CIM) .....	37
Figura 3.1: Logo Blynk.....	39
Figura 3.2: Logo Arduino .....	40
Figura 3.3: Entorno Arduino.....	41

Figura 3.4: Tarjeta Arduino Mega 2560 R3.....	44
Figura 3.5: Tarjeta Arduino Ethernet Shield.....	45
Figura 3.6: Modulo Relé de 4 canales.....	46
Figura 3.7: Esquema del módulo relé 4 canales.....	47
Figura 3.8: Sensor de Humedad FC-28.....	49
Figura 3.9: Sensor de Temperatura Ds1820.....	51
Figura 3.10: Electroválvula.....	52
Figura 3.11: Aspersor de Agua .....	53
Figura 3.12: Tarjeta Raspberry pi2 .....	55
Figura 3.13: Pagina para descargar sistema operativo Rasbian. ....	57
Figura 3.14: Configuración PuTTY .....	58
Figura 3.15: Ingreso al terminal de Raspberry Pi .....	58
Figura 3.16: Versión java.....	58
Figura 3.17: Configuración de la IP fija en la tarjeta Raspberry PI2.....	59
Figura 3.18: Configuración dispositivo móvil para comunicación con servidor local .....	60
Figura 3.19: Ejemplo Arduino .....	61
Figura 3.20: Configuración de Arduino con el servidor Blynk .....	61
Figura 3.21: Router Wireless D-link Dir-610 150mbps 4 Puertos Wifi Ipv6.....	62
Figura 3.22: Muestra de reserva de IP para la tarjeta Raspberry pi 2 .....	62
Figura 3.23: Puertos conectados al Router.....	63
Figura 3.24: Servidor Blynk Nube .....	64
Figura 3.25: Diagrama de flujo .....	65
Figura 3.26: Creación de proyecto .....	66
Figura 3.27: Pasos de Blynk .....	67
Figura 3.28: Configuración widget botón .....	68
Figura 3.29: Clonar Proyecto .....	69
Figura 4.1: Plano del lugar donde se implementó el sistema de riego.....	71
Figura 4.2: Terreno para riego por aspersión .....	72
Figura 4.3: Electroválvula.....	73
Figura 4.4: Aspersor.....	73
Figura 4.5: Caja de Mando.....	74
Figura 4.6: Diagrama de la conexión de la tarjeta ARDUINO .....	75
Figura 4.7: Jardinera.....	76

Figura 4.8: Funcionamiento de riego por goteo .....	77
Figura 4.9: Aditamentos para riego por goteo .....	77
Figura 4.10: Sensor de humedad “1” patio .....	78
Figura 4.11: Sensor de humedad “2”Jardinera.....	78
Figura 4.12: Sensor de temperatura .....	79
Figura 4.13: Configuración de dos botones de aspersion y goteo .....	79
Figura 4.14: Cambio de estado aspensor a Regando .....	80
Figura 4.15: Cambio de estado Goteo a Regando.....	81
Figura 4.16: Monitor Serie Lectura de Valores de los sensores .....	82
Figura 4.17: Prueba de Sensores .....	83
Figura 4.18: Prueba de Sensores de Humedad y temperatura.....	84
Figura 4.19: Visualización final de la aplicación.....	85
Figura 4.20: Proceso de funcionamiento.....	86

**INDICE DE TABLAS:**

Tabla 3.1: Características de algunas tarjetas de Arduino .....	43
Tabla 3.2: Características Arduino Mega.....	44
Tabla 3.3: Características de la Tarjeta Arduino Ethernet Shield.....	45
Tabla 3.4: Características de Raspberry Pi2 .....	56

## DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA EL CONTROL DE RIEGO MEDIANTE UN DISPOSITIVO MÓVIL

### RESUMEN

En este trabajo de investigación se monitorea y controla los sistemas de riego, mediante un dispositivo móvil. Para comandar los actuadores y adquirir las señales de los sensores se utilizó una tarjeta Arduino, para la comunicación y funcionamiento se configuro un servidor con una tarjeta Raspberry, para el monitoreo y mando se utilizó la plataforma de Blynk.

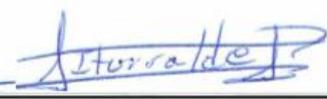
El dispositivo móvil y la plataforma Blynk son los encargados de la visualización y accionamiento, ya que en ellos se visualiza el resultado que se obtienen de los sensores, y se puede comandar los actuadores de acuerdo a la necesidad del sistema.

**Palabras claves:** Arduino, Raspberry, Blynk, Dispositivo Móvil, SCADA.



Ing. Bolívar Méndez Rengel. Msc

**Director del Trabajo de Titulación**

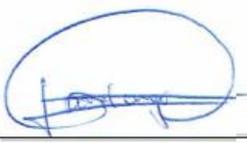


Ing. Daniel E. Iturralde Piedra Ph.D.

**Coordinador de Escuela**



Christian Patricio Fernández de Córdova A.



Daniel Humberto Vanegas V.

**Autores**

## DESIGN AND IMPLEMENTATION OF A SCADA SYSTEM FOR THE IRRIGATION CONTROL THROUGH A MOBILE DEVICE

### ABSTRACT

In this research work, irrigation systems were monitored and controlled by a mobile device. An Arduino card was used to control the actuators and acquire the sensor signals. A server with a Raspberry card was configured for operations and communications. Finally, the Blynk platform was used for monitoring and control.

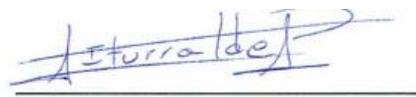
The mobile device and the Blynk platform were responsible for displaying the status and operation of the sensors. The actuators could be commanded according to the need of the system.

**Keywords:** Arduino, Raspberry, Blynk, mobile device, SCADA.



Ing. Bolívar Méndez Rengel. Msc

**Degree Work Director**

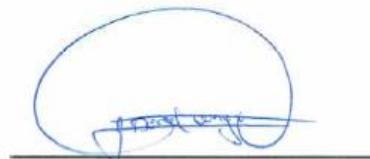


Ing. Daniel E. Iturralde Piedra Ph.D.

**Faculty Coordinator**



Christian Patricio Fernández de Córdova A.



Daniel Humberto Vanegas V.

**Authors**



Translated by

Ing. Paul Arpi

Córdova Álvarez Christian Patricio

Vanegas Verdugo Daniel Humberto

Trabajo de titulación

Ing. Bolívar Méndez Rengel.

Abril, 2018

## **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA PARA EL CONTROL DE RIEGO MEDIANTE UN DISPOSITIVO MÓVIL**

### **INTRODUCCION**

Como los seres vivos, las plantas necesitan agua para vivir. Si el suelo no tiene la suficiente cantidad de agua por medios naturales como lluvia o manantiales, es necesario un riego artificial adecuado según el tipo de plantas y suelo que se pueda tener.

Los sistemas de riego automatizados que existen actualmente en nuestro mercado son muy costosos, por lo cual se ve muy complicado que se pueda adquirir con facilidad. Por lo cual se ve la necesidad de crear un sistema de riego menos costoso e interactivo con el usuario, donde se pueda tener datos técnicos como temperatura y humedad en tiempo real y a su vez poder actuar sobre el sistema si el caso lo amerita.

Hoy con los avances tecnológicos hay grandes industrias automatizados con un sistema SCADA, con lo cual se ve la motivación de poder controlar el sistema de riego inalámbricamente con un dispositivo móvil.

En el proyecto se empezó analizando diferentes tipos de riego, como sus ventajas y desventajas, concluyendo que lo más viable para las jardineras es la implementación de riego por goteo y para la cancha que tiene una mayor área de terreno, se aplicó riego por aspersión.

Luego se analizó y se estudió todo lo concerniente de un sistema SCADA con lo cual podamos aplicar al proyecto. Teniendo claro los que buscamos y necesitamos, se

empezó a indagar sobre diferentes tipos de tarjetas de adquisición de datos como también una aplicación que vaya de acuerdo a nuestras necesidades.

Lo más conveniente para el proyecto fue utilizar la tarjeta de Arduino para adquisición de datos y una tarjeta de Raspberry pi 2 que funcione como servidor para nuestro sistema y la aplicación Blynk para poder programar y comunicarse con el servidor y así interactuar con el sistema en tiempo real.

Se logró establecer la comunicación de un dispositivo móvil con los sensores y actuadores a través del servidor, con lo cual pudimos obtener datos de humedad y temperatura en tiempo real y a su vez accionar las electroválvulas para el control de riego.

## CAPITULO 1.

### 1. SISTEMAS DE RIEGO Y AUTOMATIZACIÓN

#### 1.1 Introducción

En los últimos años se viene implementando de manera práctica y a mayor escala, sistemas que permiten la realización de labores de riego de un modo programado y remoto para facilitar el mantenimiento y la conservación de espacios verdes. Además de tener un control adecuado sobre los recursos y la importación de una conciencia de regulación y aprovechamiento del agua.

En el momento en que se habla de regular y aprovechar el agua se empieza a pensar en sistemas de control que optimicen los recursos, el concepto de automatización de regadíos responde a la necesidad de lograr un aprovechamiento más razonable sobre los recursos hídricos, tendiendo a optimizar la gestión, mejorando la eficiencia en el uso del agua para diferentes tipos de aplicaciones.

#### 1.2 Sistema de Riego.

Un sistema de riego es el conjunto de estructuras, que permite dotar de agua a la tierra o a los cultivos, por medios artificiales. El uso del agua en un sistema de riego está condicionado por la forma de distribuir las dotaciones a las tomas de regante y por la forma de dar los riegos a los campos de cultivo. (Losada Villasante, 2005)

Hay varios sistemas de riego, unos son más eficientes que otros, como también no en todos se puede ahorrar agua.

Tipos de riego:

- Gravitacional o superficie
- Presurizados

##### 1.2.1 Riego por Superficie

El riego por superficie es el método más antiguo de riego, actualmente sigue entre los más utilizados. El agua se desplaza sobre una superficie, gracias a la fuerza de la gravedad fluye de un punto a otro, de diferente nivel, el caudal va disminuyendo de

acuerdo como avanza sobre la parcela regada, debido a una infiltración en el suelo. (Riegos a pie, 2012)

Tipos de riego por superficie:

- Por surco
- Por Melga o Manto
- Por tuberías

#### **1.2.1.1 Por surco.**

Consiste en llevar el agua de riego a través de pequeños canales llamados surcos (figura 1.1). El agua corre en la dirección de la pendiente, a lo largo de la hilera de los cultivos o terreno. (Hidalgo, 2012)



Figura 1.1: Riego por surco

Fuente: <https://auladeagricultura.wikispaces.com/4.+Sistemas+de+riego>

#### **1.2.1.2 Por melga o manto.**

El agua se hace escurrir a través de una faja de terreno, en la dirección de la pendiente, y está contenida por dos bordes (figura 1.2).



Figura 1.2: Riego por manto

Fuente: <https://auladeagricultura.wikispaces.com/4.+Sistemas+de+riego>

### 1.2.1.3 Por tuberías.

El agua llega a las zanjas por medio de tubos (figura1.3). Este sistema permite reducir las pérdidas por conducción o distribución.



Figura 1.3: Riego por tuberías

Fuente: <https://auladeagricultura.wikispaces.com/4.+Sistemas+de+riego>

### Ventajas

- Se aprovecha la pendiente natural de un terreno para el riego, haciendo uso de tecnología rudimentaria
- Abarata utilizando un cuerpo de agua disponible.

### Desventajas.

- Se pierde mucha agua por evaporación y por el drenaje del suelo
- Se requiere mayor cantidad de agua
- Promueve la erosión del terreno,

- Algunas partes recibirán riego excesivo o se inundarán y otras no recibirán suficiente.

### **1.2.2 Presurizados**

Requieren de una determinada presión para operar. El agua se obtiene por una diferencia de altura entre la fuente de agua y el sector a regar, o equipo de bombeo. (Armijos Torres & Pozo Morillo, 2006)

#### **Ventajas.**

- Se adaptan mejor a las aplicaciones frecuentes de escaso volumen a las que las plantas reaccionan mejor.
- Son más eficientes en el uso del agua.
- Manejo más económico al no requerir mucha mano de obra y al no humedecer todo el suelo.
- No precisan sistematización del terreno.

#### **Desventajas:**

- Radica en la mayor inversión que requiere, tanto en lo que a equipos de riego se refiere como a las infraestructuras.

#### **1.2.2.1 Por aspersión**

Simula de alguna manera el aporte de agua que realizan las lluvias. Consiste en distribuir el agua por tuberías a presión y aplicarla a través de aspersores en forma de lluvia. Se busca aplicar una lámina que sea capaz de infiltrarse en el suelo o terreno sin producir escurrimiento del agua. Si el equipo está bien diseñado respecto al tipo de suelo a regar se obtiene una lámina muy uniforme sin que se presente escurrimiento. Los diversos sistemas existentes van desde los equipos autopropulsados como los cañones regadores o los equipos de avance frontal, hasta equipos de diferentes dimensiones de alas móviles. (Armijos Torres & Pozo Morillo, 2006)

Tanto los sistemas de aspersión como los de goteo utilizan dispositivos de emisión o descarga en los que la presión disponible en las tuberías porta emisores (ramales, alas o laterales de riego) induce un caudal de salida. La diferencia entre ambos métodos

radica en la magnitud de la presión y en la geometría del emisor. (Tarjuelo Martín-Benito, 2005)

Hay dos formas de riego por aspersión, la fija (figura 1.4) y la móvil (figura 1.5)

#### **Ventajas:**

- La conducción fuera del cuadro de cultivo se hace por tuberías sin pérdidas
- Los equipos móviles se prestan para la aplicación de riegos complementarios debido a que son desplazables y no precisan sistematización de los terrenos.

#### **APLICACIONES:**

- Se usa en una diversa gama de cultivos que van desde hortalizas, pasturas, cereales, y en riegos complementarios de cultivos extensivos, patatas, hortalizas etc.
- El riego por aspersión se emplea, tanto en las regiones secas como en las húmedas. (García Lozano, 1967)
- Se le puede considerar como un sistema versátil, ya que se adapta a la mayoría de los cultivos, así como a las diversas condiciones climáticas. (Zimmermann, 1975)



Figura 1.4: Riego por aspersión fijo

Fuente: <http://ecologiaactual.com/como-ahorrar-agua-mediante-riego-por-aspersion/>



Figura 1.5: Riego por aspersión móvil

Fuente: <https://auladeagricultura.wikispaces.com/4.+Sistemas+de+riego>

#### **1.2.2.2 Por micro aspersión**

Similar al anterior, pero a escala muy reducida. Se disponen de una gran cantidad de mangueras de riego que recorren las líneas del cultivo con emisores individuales o para un grupo de plantas “micro aspersor” que con diferentes diseños moja una superficie relativamente pequeña (figura 1.6). (Armijos Torres & Pozo Morillo, 2006)

#### **Ventajas:**

- No moja la totalidad del suelo
- Permite el riego por debajo de las copas de las plantas sin mojarlas

**USOS:** Principalmente frutales.



Figura 1.6: Riego por micro aspersion

Fuente:<http://www.jardinerosenaccion.es/riego-automatico-por-microaspersion.php>

### 1.2.2.3 Por goteo

El agua se conduce a presión por tuberías y luego por mangueras de riego que recorren las hileras del cultivo (figura 1.7).

El emisor, externo o incorporado a la manguera de riego es un “gotero” de caudal y separación variable según el suelo y los cultivos aplica el agua en forma de gotas que se van infiltrando a medida que caen. (Armijos Torres & Pozo Morillo, 2006)

#### **Ventajas**

- No moja la totalidad del terreno.
- No moja las hojas por lo que no es tan exigente en calidad de agua.
- No tiene piezas móviles y es de fácil mantenimiento.
- Gran uniformidad

**USOS:** Cultivos hortícolas y frutales



Figura 1.7: Riego por goteo

Fuente: <https://auladeagricultura.wikispaces.com/4.+Sistemas+de+riego>

### 1.3 Automatización y Aplicaciones.

#### 1.3.1 Sistema de Riego Instalado por la Compañía Waterman.

En el 2011, la compañía australiana Waterman Riego, miembro del IA Group Pty Ltd. Instaló un nuevo sistema de riego en la Ciudad de Mandurah para controlar 18 parques. Mandurah está situado al sur de Perth, la capital de Australia Occidental. Tiene una población de 74.000 y tiene una superficie de 174 km<sup>2</sup>. El sistema de control consta de 18 controladores lógicos programables equipados con procesadores basados en Linux. Cada PLC se conecta a los módulos de E / S a través de Ethernet y, a través de los routers inalámbricos 3G al servidor central mediante conexiones VPN. El servidor está cargado con el software Reliance 4 Control Server. En Mandurah, cada parque tiene una bomba para el agua subterránea y un controlador de riego para distribuir el agua a varios aspersores. El Controlador de Riego Waterman desarrollado por Automatización Industrial tiene la capacidad de controlar hasta 100 electroválvulas por el parque y medir una variedad de sensores, como la presión, el flujo o las condiciones meteorológicas, se puede apreciar la (figura 1.8) como es el panel de mando.

Los procesos de riego son visualizados y controlados por los clientes ligeros del sistema Reliance SCADA / HMI (para ser específicos, Reliance 4 Mobile Client instalado en los teléfonos celulares del personal operativo). Las personas autorizadas

pueden así acceder al sistema 24/7 a través de sus teléfonos inteligentes para monitorear y controlar el riego en cada parque.

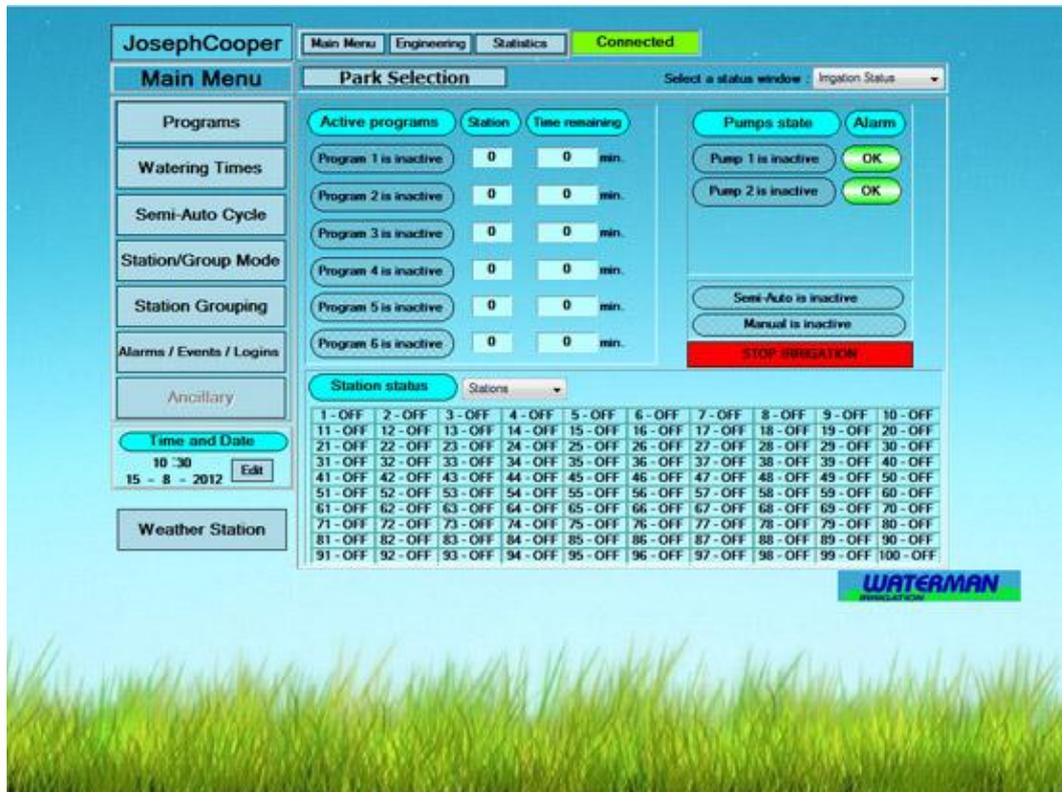


Figura 1.8: Visualización del riego en el Parque Joseph Cooper

Fuente: (Visualization and control of an irrigation control system for the Mandurah suburb in Perth, Western Australia, 2011)

### 1.3.1.1 Beneficios para los usuarios

Un buen ejemplo del impacto y la importancia de este sistema se evidencia en lo ocurrido entre diciembre 2011-febrero 2012. Un ciudadano se contactó con el consejo durante la noche informando que la línea de tubería principal de riego en uno de los parques estaba rota y una cantidad masiva de agua inundaba la zona. La denuncia fue inmediatamente remitida al supervisor de riego el cual directamente desde su casa se conectó al sistema de Waterman Riego SCADA a través de su ordenador portátil / PDA, confirmó la lectura de los datos de presión y caudal de cada parque y suspendió el programa de riego con el fin de resolver completamente el problema. Esto se logró en menos de 10 minutos de la llamada. Sin el sistema SCADA, el supervisor hubiese tenido que salir de su casa en su vehículo hasta el parque y cambiar manualmente el

sistema. Esta operación habría requerido al menos dos horas durante la noche. (Visualization and control of an irrigation control system for the Mandurah suburb in Perth, Western Australia, 2011).

### **1.3.2 Sistema de Control Remoto de Riego por Telefonía Móvil**

El objetivo general de la aplicación fue llevar a cabo el sistema de riego y fertilización de una viña mediante un control remoto, mejorando en la medida de lo posible la calidad del fruto. El elemento de control es un PLC<sup>1</sup>, que se encuentra integrado en el cultivo. El acceso al PLC, se realiza mediante un módem. Que a través de él es posible acceder a la CPU<sup>2</sup> del PLC desde prácticamente cualquier lugar del mundo, mediante el teléfono móvil consiguieron actuar sobre el PLC, el cual envía mensajes de texto a un teléfono móvil cuando se activan una de las entradas y de igual manera con este teléfono móvil activar salidas, enviando un mensaje de texto a la tarjeta que está en el módem remoto.

Los dispositivos de control utilizados en su aplicación fueron tensiómetros, sensores de humedad, válvulas eléctricas, sensores de temperatura, manómetros, bomba, aspersores, bulbos para el goteo, PLC además de un modem.

Como conclusión uno de los resultados que obtuvieron es que el sistema disminuye considerablemente los costes por desplazamiento del personal del servicio técnico, además de tener un control remoto eficiente mediante un dispositivo móvil el cual mediante mensajes de texto se puede realizar un control sobre el sistema. (Vargas Reinales, Jiménez Macías, & Pérez de la Parte, 2007)

### **1.3.3 Sistema de control basado en el sistema MyScada e Information Server**

Francisco Xavier Palacios Silva y Francisco Javier Mera Clavijo diseñaron un Sistema SCADA que se controla de forma remota a través de un dispositivo móvil y monitoreado a través de la gran nube, el internet, para el cual primero diseñaron e implementaron una plataforma de pruebas la cual consta de un motor trifásico, un PLC y un variador de frecuencia, los cuales son controlados a través de una red wifi mediante un celular Iphone 4 utilizando el software MyScada, utilizaron también el

---

<sup>1</sup> **PLC:** (*Programmable Logic Controller*) Controlador lógico programable

<sup>2</sup> **CPU:** (*Central Processing Unit*) Unidad de Proceso Central

software Information Server de la compañía Wonderware para la creación de un portal web y monitoreo dicho proceso a través de una red intranet o internet.

MySCADA es un sistema de control con todas las funciones y aplicaciones para la Adquisición y manejo de Datos (Sistemas SCADA) con una Interfaz avanzada hombre-máquina o también llamada HMI. El software proporciona al usuario la capacidad de controlar, monitorear y mostrar el estado de los procesos de tipo:

- Tecnológico
- Infraestructura
- Sistemas Industriales
- Instalaciones

Como conclusión del artículo, ellos diseñaron un sistema SCADA para el control de un proceso de encendido de un motor en lazo abierto, permitiéndoles realizar pruebas y obtener resultados sobre el comportamiento de la misma y del software. A su vez consiguieron realizar con éxito la comunicación entre la plataforma de prueba (PLC, inversor, motor) y el celular Iphone a través de una red Wi-Fi<sup>3</sup> con estándar IEEE 802.11. (Palacios Silva & Mera Clavijo, 2012)

---

<sup>3</sup> **Wi-Fi:** Es una tecnología de área local inalámbrica en red con dispositivos basados en la norma IEEE 802.11

## CAPITULO 2

### 2 SISTEMAS SCADA Y COMUNICACIONES INDUSTRIALES

#### 2.1 Introducción

En este capítulo se presenta algunas características del sistema SCADA, todo lo que tiene que ver con su arquitectura y entorno. También se analiza lo que tiene que ver con comunicaciones industriales y los protocolos de comunicación.

#### 2.2 SCADA

El sistema SCADA “*Supervisory Control And Data Acquisition*” es una aplicación software diseñado especialmente para funcionar sobre ordenadores, en el control y monitoreo de producción a distancia. Además, genera toda la información del proceso productivo a diversos usuarios, en tiempo real, con niveles jerárquicos diferentes como operadores y supervisores como también se puede tener acceso al historial de alarmas y variables de control con mayor claridad. (Rodríguez Penin, Sistemas SCADA, 2007)

##### 2.2.1 Objetivo de un sistema SCADA

Lo que se busca en un sistema SCADA es que sea lo más eficiente posible y compense costo beneficio, cuanto se invierte en un inicio y en qué tiempo se compensaría, ahorrando mantenimientos con la visita innecesaria de técnicos a la planta de procesos, ya que se puede instalar un sistema de control automático que alerte cuando hay que inspeccionar las máquinas y si tienen algún fallo que se prenda una alarma. A su vez se tendría modernos ordenadores supliendo a grandes paneles repletos de cables, con lo cual se gestionaría los datos recopilados de tal forma de explotar el sistema de una forma más eficaz.

##### 2.2.2 Prestaciones

Un paquete SCADA, tiene funciones y utilidades que permiten obtener una comunicación clara y rápida entre el proceso y el operador. Con las herramientas que ofrece se puede monitorear datos en tiempo real, supervisar y modificar el control establecido así se evitaría supervisión humana. Otro tipo de prestaciones que tiene este paquete es guardar los valores obtenidos y poderlos evaluar en un futuro como

(parámetros de temperatura y velocidad), a su vez se visualizan las señales de alarmas y eventos con lo cual el operador puede tomar las medidas necesarias para su correcto funcionamiento.

La figura 2.1 es un sistema donde se controla y supervisa los procesos de toda la planta a través de un panel táctil, como también se puede hacer de forma individual cada elemento de la planta.

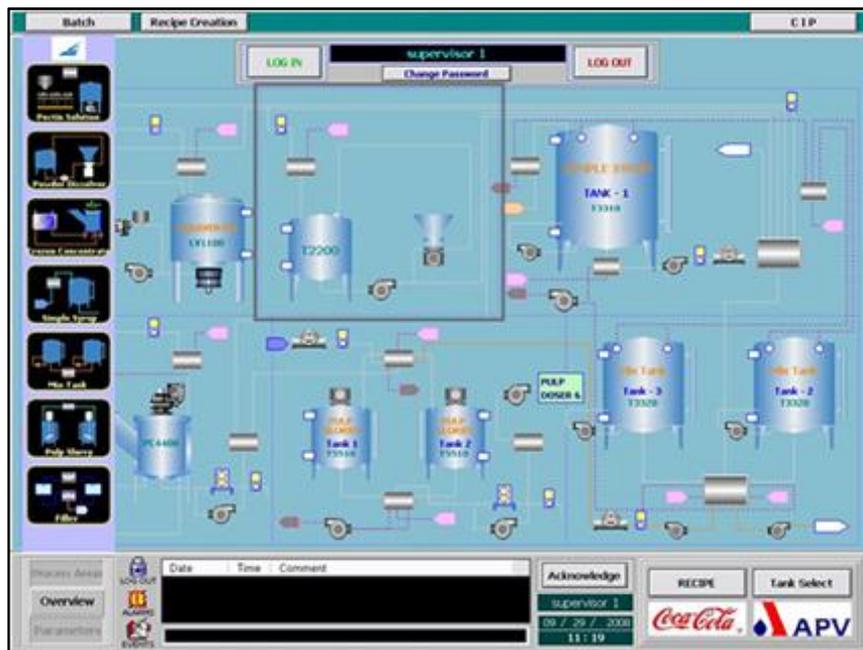


Figura 2.1: Embotelladora de Coca Cola en Filipinas

Fuente: <http://exponentcontrols.com.ph/projects/>

### 2.2.3 Ventajas

Cuando se habla de un sistema SCADA, no es solo la pantalla que da la información, es mucho más, tras las pantallas se encuentran elementos de control y avanzados sistemas de software. (Rodríguez Penin, Sistemas SCADA, 2007)

Hay muchísimas ventajas en un sistema SCADA en lo cual nos referiremos a las más importantes.

- No se necesita ser un experto en la materia para crear un sistema SCADA gracias a que el nivel de desarrollo visual no es complicado.

- Gracias a las múltiples tarjetas de adquisición disponibles (voltaje, corriente, temperatura, etc.) se puede integrar con los programas PLC con todos los sensores y actuadores.
- Las herramientas de diagnóstico son una gran ayuda que permiten localizar los errores más rápido y con lo que no se perdería valioso tiempo.
- El sistema puede recoger y exportar datos a formatos diferentes como hojas de cálculo.
- Ahora con los sistemas de comunicación estandarizados y modernos se puede integrar los sistemas más rápido.
- Con el internet podemos tener el control de nuestro sistema desde cualquier parte del planeta.
- A través de la información de las estaciones remotas que se evalúa en el centro de control, se realizan menor número de mantenimientos, con lo cual se necesita menos personal.

#### **2.2.4 El Entorno**

La automatización de sistemas ha pasado a formar parte de las empresas con el fin de mejorar su productividad y calidad.

En la figura 2.2 representa los flujos de información dentro de la empresa y representa como se realiza la integración a todos los niveles es similar a la pirámide de automatización CIM<sup>4</sup>. (Rodríguez Penin, Sistemas SCADA, 2007)

---

<sup>4</sup> CIM: Computer Integrated Manufacturing

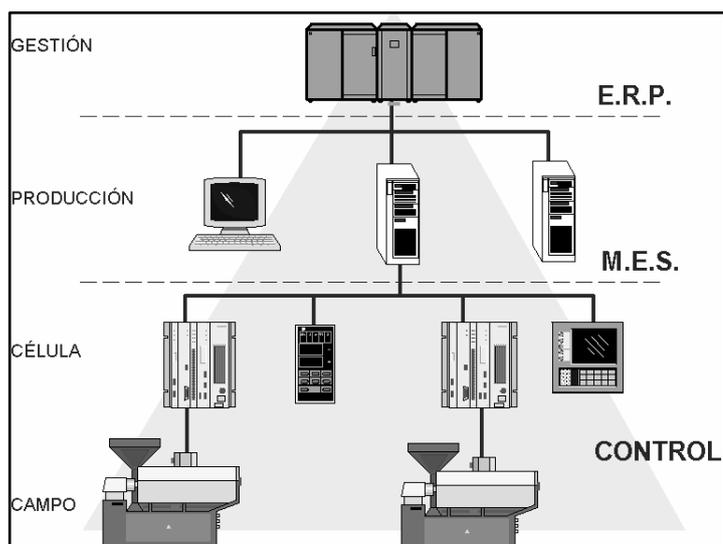


Figura 2.2: Presenta una estructura corporativa dividida en tres niveles que engloban funciones principales.

Fuente: (Rodríguez Penin, Sistemas SCADA, 2007)

La estructura corporativa se divide en tres niveles, cada una engloba funciones especiales.

- E.R.P.<sup>5</sup>: Engloba la parte de gestión de finanzas, compras, ventas.
- M.E.S.<sup>6</sup>: Comprende la gestión de calidad, producción, mantenimiento y documentación.
- CONTROL: Engloba toda la parte de automatización.

La finalidad de esta pirámide es disponer la máxima cantidad de información de la producción, con el fin de saber demandas de productos, existencias de materiales y para poder coordinar ciclos de mantenimiento.

### 2.3 Criterios de selección y diseño

Desde la creación de los primeros sistemas automáticos, los sistemas de control han ido subiendo su grado de dificultad y están por todos los campos desde máquinas expendedoras hasta grandes sistemas de gestión de energía. Han pasado a ser fundamentales en las grandes industrias y nuestro diario vivir, solo percibimos su verdadero valor e importancia cuando estas dejan de funcionar correctamente debido a situaciones imprevistas. (Rodríguez Penin, Sistemas SCADA, 2007)

<sup>5</sup> E.R.P. : (Enterprise Resource Planning o Planificación de Recursos Empresariales)

<sup>6</sup> M.E.S. : (Manufacturing Execution System o Gestión de la Producción)

Cualquier sistema de control es útil mientras funcione adecuadamente, caso contrario puede causar problemas de forma directa o indirecta. La reacción de un sistema de control ante situaciones inesperadas determinará su fiabilidad.

Los parámetros que influyen para una selección determinada de sistema son los siguientes:

- Disponibilidad
- Robustez
- Seguridad
- Prestaciones
- Mantenibilidad
- Escalabilidad

### **2.3.1 Disponibilidad**

Un buen sistema debe tener funcionalidad todo el tiempo basándose en especificaciones del diseño. Si un elemento del sistema llegara a fallar otro deberá tomar las funciones de este de una forma transparente pudiendo así trabajar sin la necesidad de suspender el sistema por lo cual se continuaría trabajando normalmente, este sistema es conocido como principio de redundancia (figura 2.3).

Si una parte del proceso del sistema se llegara a interrumpir provocaría inmediatamente falta de control en los sistemas de producción con lo cual se perdería calidad.

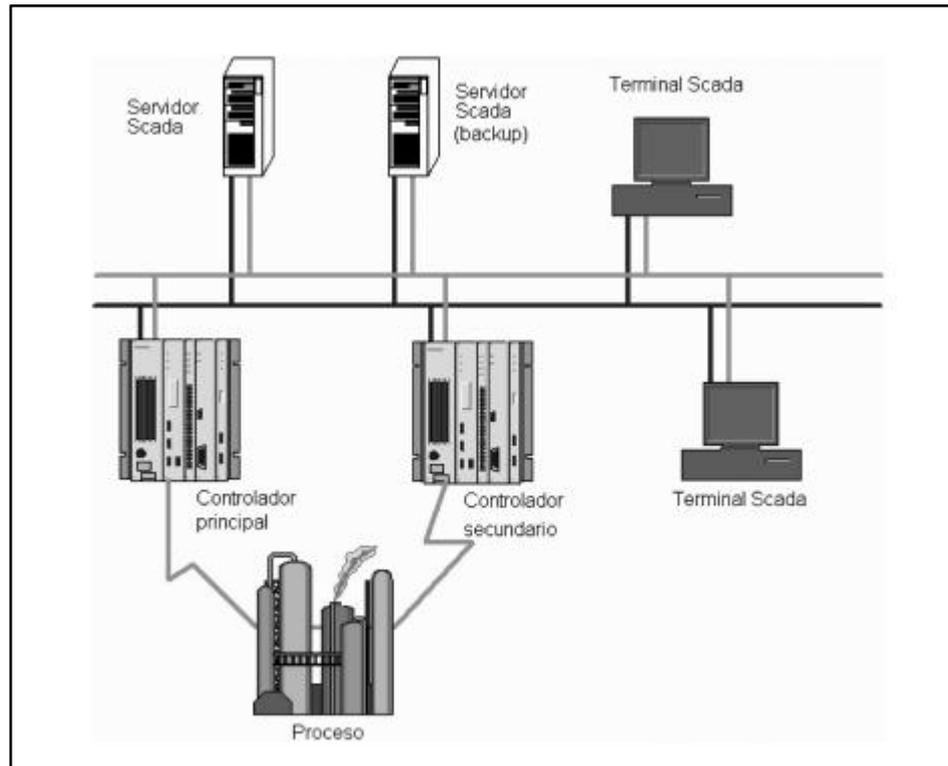


Figura 2.3: Principio de Redundancia

Fuente: (Rodríguez Penin, Sistemas SCADA, 2007)

### 2.3.2 Robustez

Ante cualquier percance, fallo o accidente, un buen sistema de mantener un nivel de operatividad suficiente como para mantener unos mínimos de servicio. Si un sistema de fabricación de piezas falla la comunicación, este debe tener la capacidad de autogestión para poder mantener un mínimo de servicio sobre su área trabajo, hasta restablecerse la comunicación con el área central. (Rodríguez Penin, Sistemas SCADA, 2007)

### 2.3.3 Seguridad

Un sistema de control puede utilizar uno o varios sistemas de comunicación para conectar todos los puntos de control en un proceso. Cuando se utilizan sistemas de comunicación para acceder al control desde varios puntos dentro y fuera de una empresa hay la posibilidad que algunos tipos de acceso no sean deseados y así podría alterar el correcto funcionamiento de un sistema.

Ante situaciones de fallo del sistema por diseño o un usuario malintencionado se debe realizar estrategias para prevenir, detectar y defenderse de acciones no deseadas.

#### **2.3.4 Prestaciones**

Se refiere al tiempo de respuesta del sistema, en caso que la actividad que se desarrolla aumentara considerablemente en equipos informáticos como personal que los maneja. El equipo debe asimilar la información que se genera inclusive en condiciones extremas de tal forma que no se pierda información.

#### **2.3.5 Mantenimiento**

Los mantenimientos pueden reducirse a un mínimo si el sistema está provisto de buenas herramientas para un buen diagnóstico y poder hacer un mantenimiento preventivo, modificaciones y pruebas simultaneas al funcionamiento normal del sistema. (Rodríguez Penin, Sistemas SCADA, 2007)

#### **2.3.6 Escalabilidad**

Se refiere a la posibilidad de ampliar el sistema con nuevas herramientas y requerimientos de tiempo necesarios para implementar estas ampliaciones, debido a:

- Espacio disponible
- Capacidad de equipo informático
- Capacidad del sistema de comunicaciones

### **2.4 Arquitectura de un sistema SCADA**

Las primeras automatizaciones que se realizaron se centraban en la PC que tenían el control de prácticamente todo.

El sistema se divide en tres bloques principales (figura 2.4).

- Software de adquisición de datos y control (SCADA)
- Sistema de adquisición y mando (sensores y actuadores)
- Sistema de interconexión (comunicaciones)

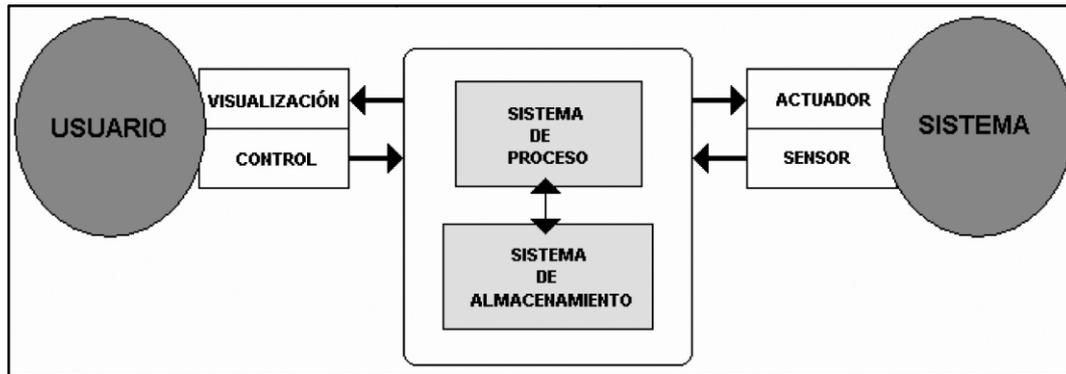


Figura 2.4: Estructura básica de un sistema de supervisión y mando.

Fuente: (Rodríguez Penin, Sistemas SCADA, 2007)

El usuario accede al sistema de proceso mediante herramientas de visualización y control, software (SCADA), con lo cual puede obtener señales de los sensores del sistema. De acuerdo a los comandos ejecutados por el usuario el sistema de proceso inicia acciones a través de los actuadores para tener el control del sistema.

La comunicación entre el sistema de proceso con el usuario suele ser a través de redes de comunicación corporativa (Ethernet), con los elementos de campo (sensores y actuadores) mediante buses de campo.

#### 2.4.1.1 Interface hombre maquina (HMI)

Es la interacción del hombre con la maquina a través de medios computacionales. En un inicio estos sistemas consistían en grandes paneles compuestos por luces piloto, pulsadores, indicadores digitales y análogos. Ahora es posible contar con sistemas de HMI más sofisticados con un mejor rendimiento y se representa en forma gráfica en pantallas de visualización como se puede apreciar en la (figura 2.5).

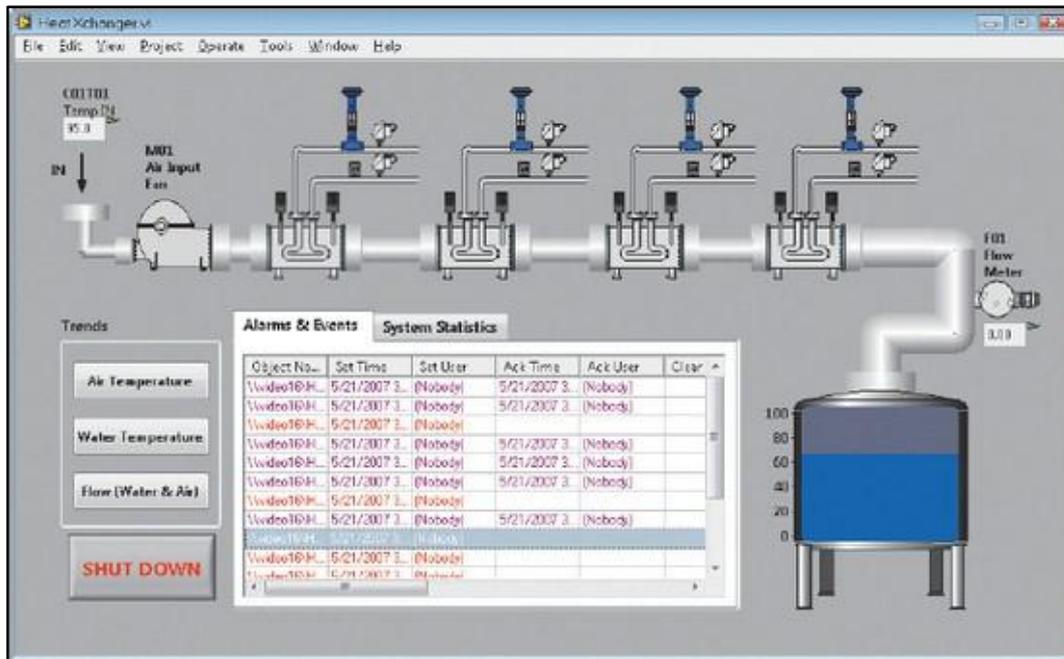


Figura 2.5: Una aplicación HMI integrado con los programas de LabVIEW 8.5 DSC Módulo themultisegmented

Fuente: <http://www.ni.com/newsletter/50160/en/>

#### 2.4.1.2 Unidad Terminal Maestra (MTU)

Centraliza el mando del sistema. Toda información que se genera, almacena y procesa de una forma ordenada, para que puedan disponer diversos usuarios (figura 2.6).

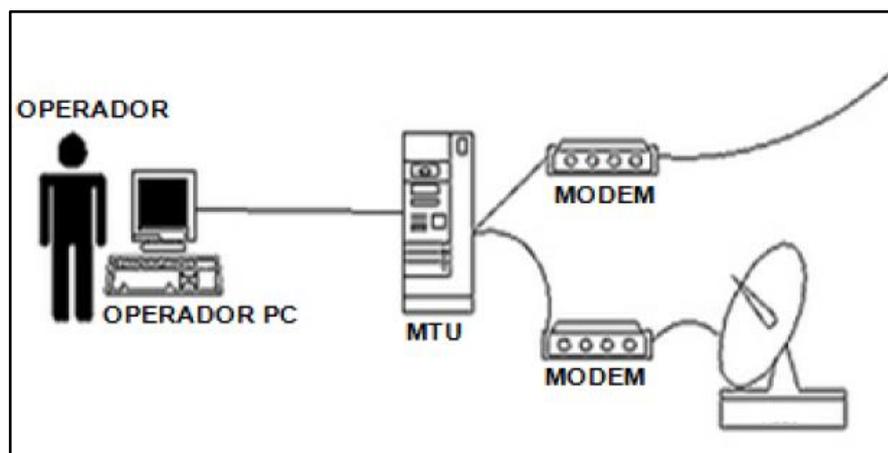


Figura 2.6: Esquema de conexión MTU

Fuente: (Gallegos Reinoso & Marín Buele, 2012)

### 2.4.1.3 Unidad remota (RTU)

Se encarga de recoger datos y procesarlos para enviarlos a la unidad central. Tienen canales de entrada para detección de variables del sistema como de salida para el control y activación de alarmas y a su vez tiene puertos para la comunicación (figura 2.7).



Figura 2.7: RTU

Fuente: <http://www.electronicshub.org/scada-system/>

### 2.4.1.4 Sistemas de comunicación

Con los sistemas de comunicación el operador puede comunicarse con cualquier punto local o remoto así pudiendo intercambiar información en tiempo real. Los sistemas de transporte de información suelen ser cableados e inalámbricos, para grandes distancias de comunicación se suele utilizar por satélite o radio y para una distancia menor el más indicado cableado.

#### **2.4.1.4.1 Topologías**

Es un conjunto de nodos interconectados que intercambian datos. Hay varias formas de combinar los elementos de la red, determinando así las topologías de la red según la configuración entre nodos.

Algunos tipos de topología:

1. Punto a punto
2. Multipunto Dedicado
3. Multipunto Compartido Estrella
4. Multipunto Compartido en Bus
5. Multipunto Compartido en Anillo

#### **2.4.2 Software**

Son programas específicos que permite la comunicación entre dispositivos de campo, entre niveles de supervisión, niveles de gerencia y administración.

Parte del paquete de software contiene los controladores que permiten la comunicación de nuestra aplicación con el exterior, con lo que se ocupa en gestionar enlaces y protocolos de comunicación. (Rodríguez Penin, Sistemas SCADA, 2007)

En un programa de SCADA hay dos bloques diferenciados:

- El programa de desarrollo donde se crea y edita diferentes ventanas para el uso de la aplicación.
- El programa Run – time el cual ejecuta la aplicación creada por el programa de desarrollo.

Una parte de sus características es que puede restringir el acceso de usuarios al sistema y puede generar señales de alarma ante la presencia de fallas.

## 2.5 Comunicaciones Industriales

Estamos en una época en que prácticamente todo se basa en electricidad, por lo cual, la forma más cómoda para transmitir una señal es mediante señal eléctrica a través de un cable que une el sensor con el sistema de control.

Cuando se empezaron utilizar las señales en un sistema en su primer ciclo todas las comunicaciones entre el armario de control y las máquinas se realizaban mediante cables. En un principio funcionan muy bien si hay una máquina, pero al haber varias máquinas se empezaba a complicar por las interferencias, caída de tensión, las canales de distribución eléctrica llenas de cables. (Rodríguez Penin, Comunicaciones Industriales, 2008)

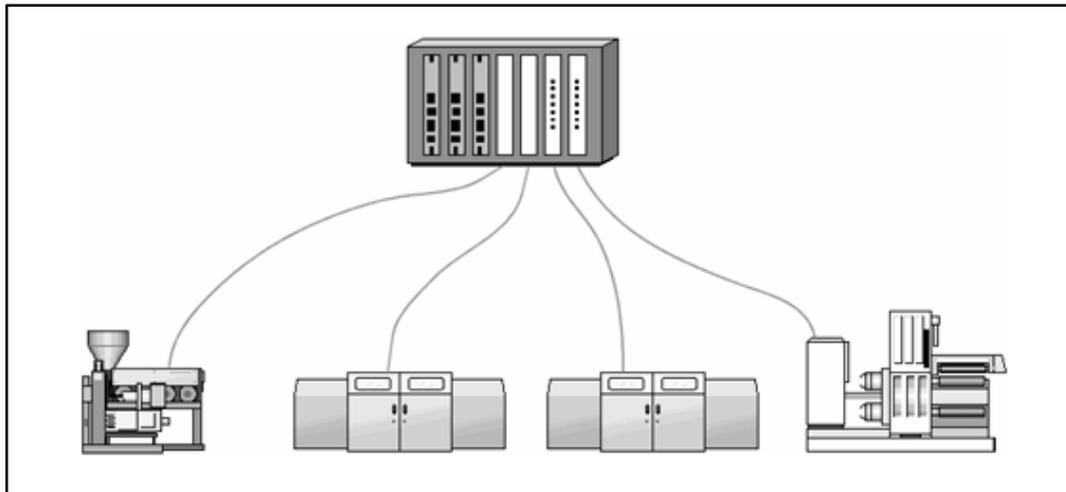


Figura 2.8: Control Centralizado

Fuente: (Rodríguez Penin, Comunicaciones Industriales, 2008)

Al tener varios inconvenientes en tener un control centralizado se vio la necesidad de simplificar las instalaciones y reducir mantenimientos, con lo cual los sistemas de producción complejos se dividieron en subsistemas más sencillos (figura 2.8).

Con la llegada del PLC<sup>7</sup> se pudo reducir en gran cantidad el material que se usa para el control de funcionamiento de una máquina. Al poder conectarse entre sí los PLC permitió eliminar prácticamente todo el cableado de control entre máquinas quedando

---

<sup>7</sup> PLC : Programmable Logic Controllers

una sola línea de comunicación entre ellas (figura 2.9). (Rodríguez Penin, Sistemas SCADA, 2007)

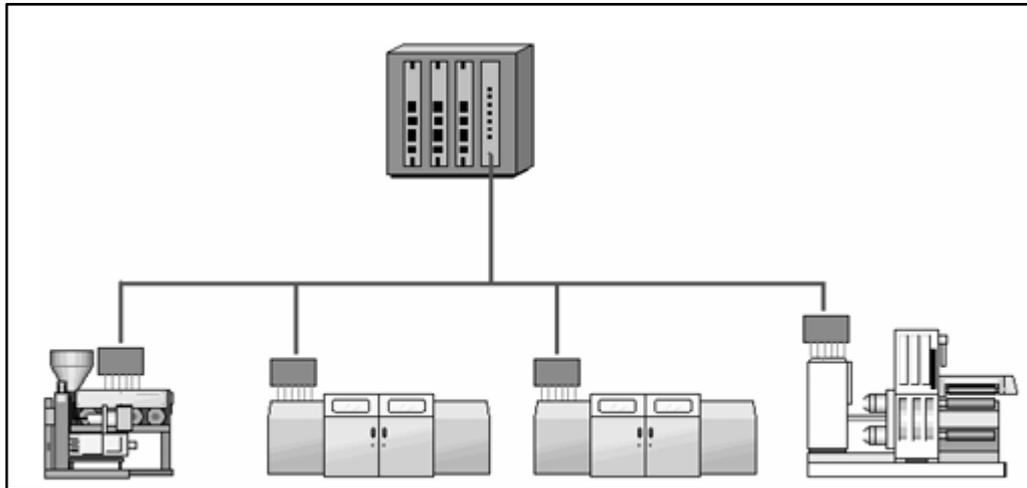


Figura 2.9: Control Distribuido

Fuente: (Rodríguez Penin, Comunicaciones Industriales, 2008)

### 2.5.1 Sistema de transporte de señal

Si se quiere intercambiar información entre dos equipos, es necesario un medio de transporte que sea capaz de contener la energía con esa información. Los medios de transmisión utilizados para encauzar la energía, es la siguiente:

- **Cable eléctrico**
- **Fibra óptica**
- **Enlace óptico**
- **Radiofrecuencia**
- **Microondas**
- **Satélite**

## 2.5.2 Sistema de transmisión de señal

Para tener una buena transmisión de señal entre dos puntos, es importante que la señal recibida llegue en óptimas condiciones al destinatario tal y como se ha enviado por el emisor.

### 2.5.2.1 Niveles de tensión.

La transmisión por señales de tensión no es recomendable en distancias importantes. Ello es debido a que la tensión depende de la resistencia del cable y de las capacidades del mismo. (Rodríguez Penin, Comunicaciones Industriales, 2008)

### 2.5.2.2 Bucle de corriente

Con esta tecnología se puede transmitir señales análogas a grandes distancias, sin perder o modificar la señal original, gracias a que cuenta con un bucle analógico de (4 a 20 miliamperios)

Para realizar un bucle se necesita, al menos, 4 elementos que son:

- El emisor
- La alimentación del bucle
- El cable
- El receptor

Se puede observar en la (figura 2.10) como se enlaza el bucle de corriente.

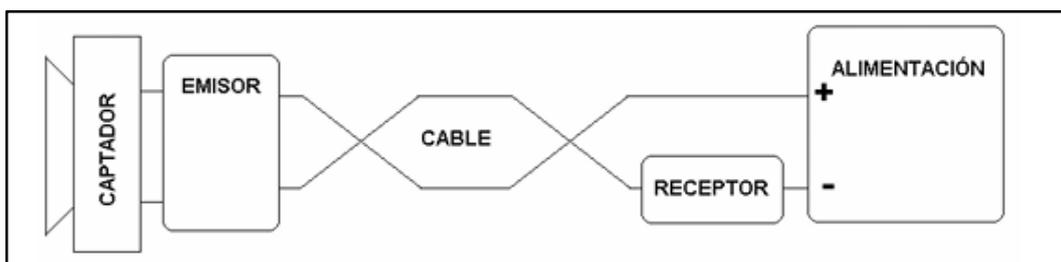


Figura 2.10: Bucle analógico de corriente

Fuente: (Rodríguez Penin, Comunicaciones Industriales, 2008)

### 2.5.2.3 Señal modulada

Estas señales sirven para transmitir a grandes distancias o en redes de altas velocidades donde no se puede utilizar el método variación de nivel de tensión. Por tal motivo se utiliza el método de señales oscilantes que se propaga a grandes distancias.

## 2.6 Protocolos de comunicación

El protocolo de comunicación engloba todas las reglas y convenciones que deben seguir dos equipos cualesquiera para poder intercambiar información. (Rodríguez Penin, Comunicaciones Industriales, 2008)

Para que pueda existir comunicación entre ambos elementos de la red, deben utilizar la misma configuración de protocolos.

Los enlaces de comunicación se pueden estructurar de la siguiente manera (figura 2.11):

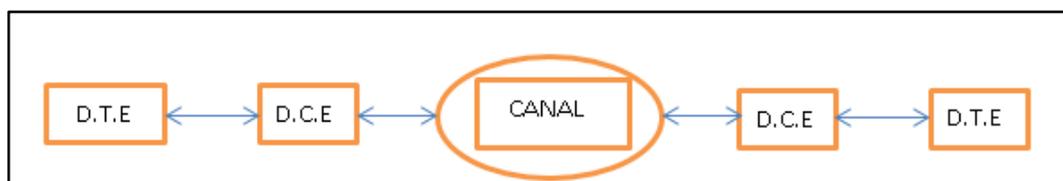


Figura 2.11: Componentes de un enlace de datos

Fuente: (Rodríguez Penin, Comunicaciones Industriales, 2008)

DTE<sup>8</sup>: Equipo Terminal de Datos.

DCE<sup>9</sup>: Equipo de Comunicación de Datos.

El objetivo de un protocolo de comunicación es poder conectar y mantener flujo de información entre dos Equipos Terminal de Datos (DTE), con el que no se tendrían fallos al intercambiar información, siendo segura la comunicación.

Cada fabricante invierte en investigaciones para que sus equipos cubran determinadas necesidades, y que pueda ser la mejor solución para el mercado.

Casi cualquier protocolo puede integrarse en mayor o menor medida en cualquier nivel de la pirámide de automatización, la idea surge en encontrar la relación prestaciones/precio y el equilibrio entre varias tecnologías que se puedan complementarse.

<sup>8</sup> DTE: Data Terminal Equipment

<sup>9</sup> DCE: Data Communication Equipment

### 2.6.1 Protocolo OSI

El término interconexión de sistemas abiertos (OSI, de open system interconnection) es el nombre de un conjunto de normas para comunicaciones entre computadoras.

Es una normativa estandarizada muy útil ya que, al existir muchas tecnologías, y fábricas en el mundo de las telecomunicaciones, se tuvo que crear un método para que todos se pudieran entender de algún modo.

Este modelo se divide en siete capas (figura 2.12):

- **Capa Física**  
Es la encargada de interactuar con el medio físico y establecer reglas de transmisión de bits.
- **Capa de enlace de datos**  
Proporciona los medios para activar, mantener y desactivar un enlace físico fiable. Su principal servicio es la detección y control de errores.
- **Capa de Red**  
Esta capa controla las operaciones de la subred, y está encargada de llevar los paquetes desde el origen hasta su destino dentro de la red de comunicación.
- **Capa de Transporte**  
Esta capa acepta datos de las capas superiores y es encargada de llevar a la capa de red y asegurarse que todas lleguen de forma correcta al otro extremo.
- **Capa de Sesión**  
Permite que usuarios de máquinas diferentes establezcan conexiones entre ellos como también manteniendo y controlando dicho enlace. Administra el token, así impide q dos partes traten de realizar la misma operación crítica al mismo tiempo.
- **Capa de Presentación**  
Define el formato de datos a transmitir. Su función es que ordenadores con diferente representación de datos se puedan comunicar
- **Capa de Aplicación**  
Ofrece a las aplicaciones la posibilidad de acceder a los servicio de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar

datos. Algunos de los protocolos que pertenecen a esta capa HTTP<sup>10</sup>, FTP<sup>11</sup>, SMTP<sup>12</sup>.

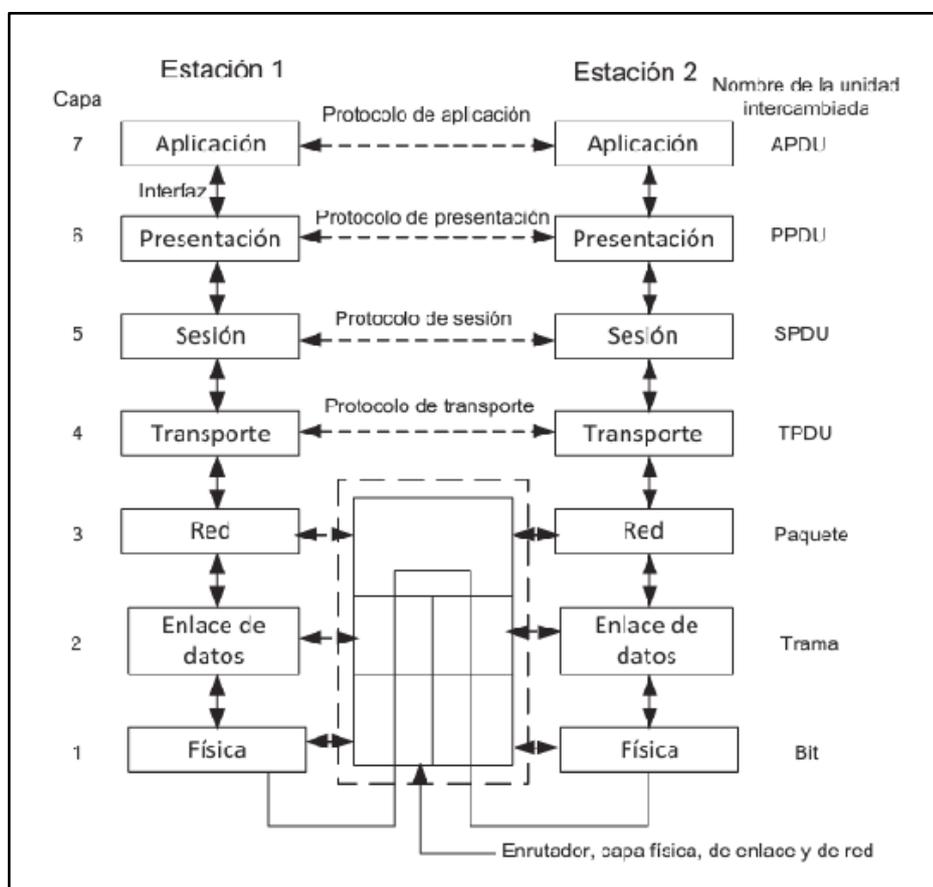


Figura 2.12: Capas y unidades de intercambio del modelo OSI  
Fuente: (Oliva, Castro Gil, & Díaz Orueta, 2013)

## 2.6.2 Protocolo TCP/IP

Este modelo TCP/IP fue el modelo utilizado por ARPANET<sup>13</sup> y ahora es utilizado por el actual internet. Este modelo lo desarrolló el departamento de Defensa de Estados Unidos, en un principio solo utilizaban algunas universidades e instalaciones gubernamentales. Más tarde dicha red se liberó para que diferentes redes de todo el mundo se pudieran conectarse entre sí. (Oliva, Castro Gil, & Díaz Orueta, 2013)

<sup>10</sup> **HTTP:** Hypertext Transfer Protocol

<sup>11</sup> **FTP:** File Transfer Protocol

<sup>12</sup> **SMTP:** Simple Mail Transfer Protocol

<sup>13</sup> **ARPANET:** Advanced Research Projects Agency Network

La arquitectura TCP/IP recibe el nombre por sus principales protocolos. Protocolo de internet IP<sup>14</sup> y el protocolo de control de transmisión TCP<sup>15</sup>. TCP/IP está dividida en 5 capas.

- **Capa Física**

Define la interface física entre el computador, terminal y medio de transmisión o red.



Figura 2.13: Comparación del Modelos OSI con TCP/IP

Fuente: <https://christiansotelo.wordpress.com/2010/04/16/7-capas-de-la-informacion-protocolo-tcpip/>

- **Capa de Acceso a la Red**

Es el responsable del intercambio de información entre el sistema final y la red a la cual está conectado. El tipo de software que se utiliza depende mucho de la red que se disponga porque hay diversos tipos de estándares.

<sup>14</sup> **IP:** Internet Protocol

<sup>15</sup> **TCP:** Transmission Control Protocol

- **Capa de Internet**

La capa de acceso a la red está relacionada con el acceso y el encaminamiento de los datos. Cuando los dos dispositivos que se quieren comunicar están en diferentes redes, se necesitarán una serie de procedimientos para que los datos atraviesen las distintas redes interconectadas. Esta es la función de la capa Internet. (Oliva, Castro Gil, & Díaz Orueta, 2013)

- **Capa de Transporte**

Esta capa permite que las entidades iguales en el host de origen y destino puedan llevar una conversación.

- **Capa de Aplicación**

Contiene todos los protocolos del nivel superior.

## 2.7 Tipos de red según forma (topológica)

Una topología de red es la forma que está diseñada una red sea en el plano físico o lógico. La topología define la forma de conexión de los equipos que hay alrededor del medio de transmisión, determinando unas estructuras de red:

- **Redes Centralizadas**

Todos los equipos están supeditados a un equipo central (Host) que controla todo el sistema. Un fallo de un terminal no afecta el correcto funcionamiento de la red, pero si el fallo de un Host.

- **Redes Distribuidas**

En este tipo de red los equipos pueden ser sencillos que comparten cargas de trabajo. El fallo de un terminal no afecta el resto de equipos.

Las redes centralizadas se basan en la potencia de Host<sup>16</sup>, mientras que la las redes de distribución se basan en distribución de los equipos, menos potentes, sus equipos son más manejables y son más tolerantes a fallos.

Existen algunas configuraciones básicas:

- Anillo

---

<sup>16</sup> **HOST:** Un host o anfitrión es un ordenador que funciona como el punto de inicio y final de las transferencias de datos.

- Estrella
- Bus
- Árbol
- Red

### 2.7.1 Anillo

Esta red es un conjunto de equipos conectados unidireccionalmente estableciendo enlaces punto a punto, formando un bucle de anillo cerrado (figura 2.15). Esta topología puede ser utilizada para enlaces de muy alta velocidad sobre grandes distancias.

Un anillo puede proporcionar potencialmente el mejor rendimiento que cualquier otra topología, una desventaja es que el fallo en cualquier parte del anillo el enlace se perdería y se inutiliza toda la red.

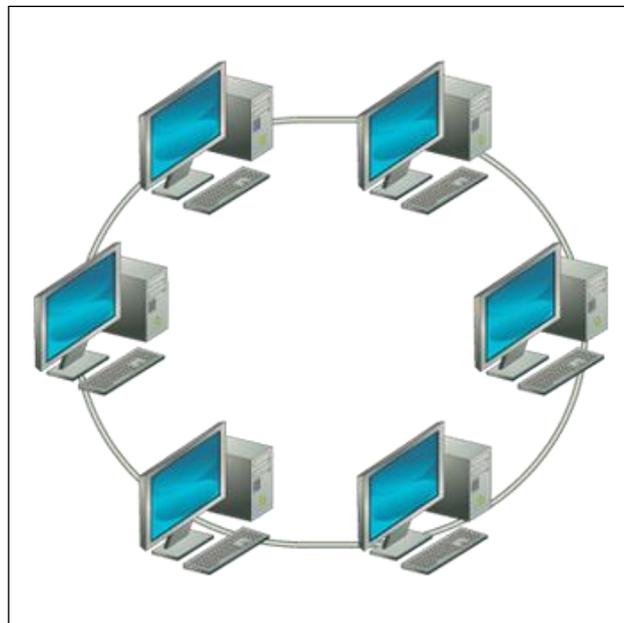


Figura 2.14: Topología de anillo

Fuente: <http://josedejesusmejiagonzalez.blogspot.com/>

### 2.7.2 Estrella

Cada nodo de red se conecta al nodo central común, comúnmente esta conexión se hace bidireccional. Aunque también suele hacerse dos enlaces punto a punto, uno para transmisión de la información, y el otro para la repetición. (Oliva, Castro Gil, & Díaz Orueta, 2013)

Esta topología generalmente es buena para distancias cortas y puede ofrecer velocidades elevadas a un número pequeño de dispositivos.

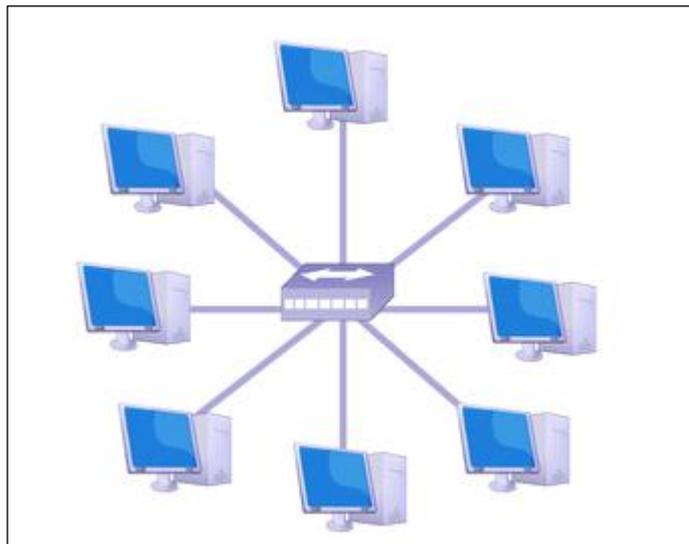


Figura 2.15: Topología Estrella

Fuente: <https://glendasnotepad.wordpress.com/2008/08/10/159/>

### 2.7.3 Bus

Todos los elementos de la red o nodo se conectan de una forma directa, a un medio de transmisión lineal o bus. La comunicación entre el nodo y la toma de conexión es full-duplex, por lo que es posible la transmisión y recepción de datos a través del bus. (Oliva, Castro Gil, & Díaz Orueta, 2013)

Cada nodo envía la información y puede ser recibida por cualquier otro nodo, para no tener este tipo de problemas, a cada nodo tiene una identificación o dirección única de red. Un aspecto a tener en cuenta es cuando existen transmisiones simultáneas pueden existir colisiones, para eso se debe establecer un mecanismo de control de acceso que regule la transmisión de datos.

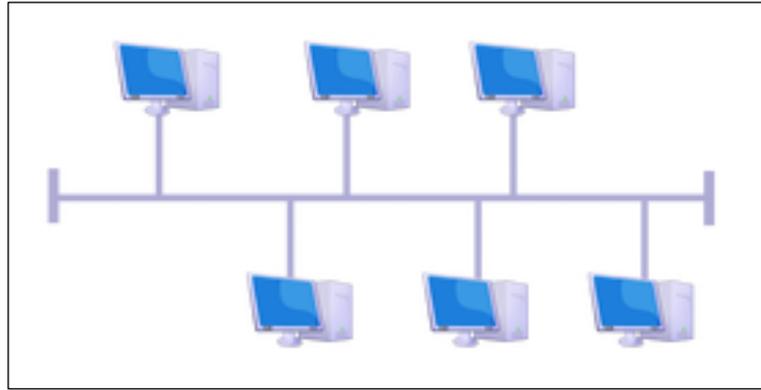


Figura 2.16: Topología tipo Bus

Fuente: <https://glendasnotepad.wordpress.com/2008/08/10/159/>

#### 2.7.4 Árbol

Al igual que la topología en Bus las transmisiones se propagan a cada ramal de una red y llegan a todos los nodos o estaciones de la red. La topología en árbol puede verse como una combinación de varias topologías en estrella. Tanto la de árbol como la de estrella son similares a la de bus cuando el nodo de interconexión trabaja en modo difusión, pues la información se propaga hacia todas las estaciones. (Redes inalámbricas y cableadas, 2014)

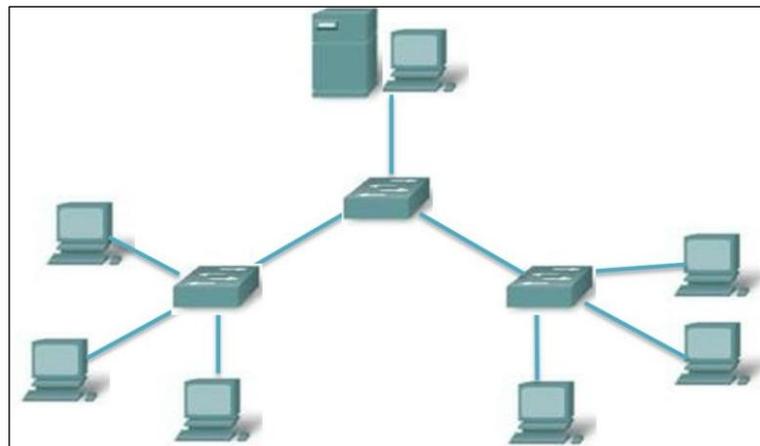


Figura 2.17: Topología tipo Árbol

Fuente: <http://lindaalmanzavilladiego.blogspot.com/>

### 2.7.5 Red

A través de varios caminos se puede establecer la conexión de dos estaciones.

Este tipo de sistema es fiable y tolerante a fallos. Si se llegara a caer una línea de transmisión se re direcciona por otro camino, una desventaja en este sistema es el alto costo en su implementación.

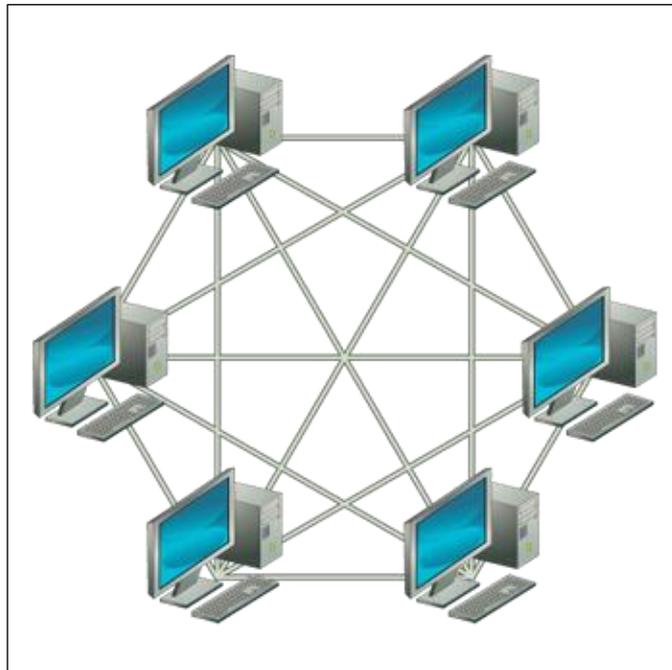


Figura 2.18: Topología tipo Red

Fuente: <https://www.emaze.com/@ACWIFTCF/TIPOS-DE-REDES>

### 2.8 Pirámide de automatización

Los sistemas de automatización pueden ser divididos en distintos niveles, conformando la pirámide de automatización, CIM (Computer Integrated Manufacturing). Se vinculan entre sí, los niveles de pirámide, por medio de distintos lenguajes de comunicación, conocidos como protocolos de comunicación.

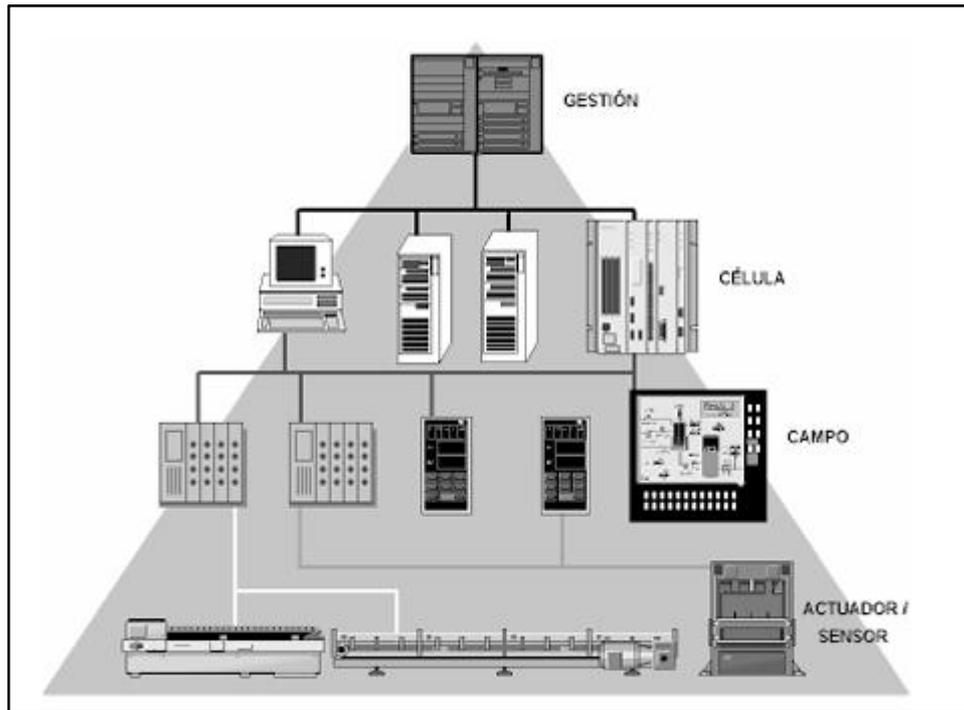


Figura 2.19: Pirámide (CIM)

Fuente: (Rodríguez Penin, Comunicaciones Industriales, 2008)

Los niveles que se dividen, se clasifican en el tipo de tráfico e información que se intercambian cada uno:

- Nivel de gestión
- Nivel de célula
- Nivel de campo
- Nivel actuador/sensor

### 2.8.1 Nivel de Gestión

En este nivel es tipo corporativo, maneja grandes cantidades de información se encuentra alejado de la parte de procesos productivos. A través de este nivel es posible obtener información de los niveles inferiores de una o varias plantas, con la información obtenida es posible sacar estadísticas acerca del proceso de producción.

### **2.8.2 Nivel de célula**

En este nivel se procesa las tareas de automatización, se puede visualizar como se lleva a cabo el proceso de la planta. Se encarga de controlar la interacción entre los distintitos dispositivos ubicados en el nivel de campo, a su vez puede controlar y monitorear varios procesos a la vez.

### **2.8.3 Nivel de campo**

Es la unión entre las instalaciones y los equipos que la controlan, en este nivel permite que funcionen conjuntamente sensores y actuadores para realizar un proceso industrial deseado, poseen autonomía suficiente para realizar procesos productivos por sí mismos.

### **2.8.4 Nivel actuador/sensor**

Este nivel es el más bajo de la pirámide de automatización, está formado por los elementos de medida (sensores) y mando (actuadores) distribuidos en una línea de producción, trabaja con poca información y su gestión es relativamente más sencilla.

## CAPITULO 3.

### 3. DISEÑO DEL SISTEMA SCADA PARA EL CONTROL DE RIEGO.

#### Introducción

En este capítulo se va a explicar cómo se diseñó el sistema SCADA para el control de riego mediante un dispositivo móvil, tanto en software como en hardware para la implementación de este proyecto, además se conocerá datos técnicos sobre los sensores y actuadores utilizados en la implementación del sistema SCADA de riego.

#### 3.1 SOFTWARE

Se utilizaron dos plataformas de software para poder crear el proyecto tanto para una red local como el control a través de Internet.

##### 3.1.1 Blynk

Es una plataforma con aplicaciones para iOS y Android que puede soportar más de 70 plataformas de hardware y los principales tipos de conectividad. (Blynk, 2016)

En este caso se puede construir una interface gráfica para cualquier proyecto en el hardware soportado, simplemente arrastrando los widgets<sup>17</sup> Figura3.1.



Figura 3.1: Logo Blynk

Fuente: (Blynk, 2016)

---

<sup>17</sup> **Widgets** : Pequeña aplicación

### 3.1.2 Arduino

Arduino es una plataforma electrónica de código abierto basado en hardware y software. Con el paso de los años Arduino ha sido el cerebro de muchos proyectos, tanto simples como también complejos. Tiene una gran comunidad que están desde fabricantes, estudiantes, aficionados, artistas, programadores y profesionales, que se han reunido en torno a esta plataforma de código abierto y aportado sus conocimientos que es una ayuda para principiantes y expertos. (Arduino, 2016)

La programación del software de Arduino es fácil, que lo pueden utilizar principiantes, como a su vez flexible para usuarios avanzados. Se basa convenientemente en el entorno de programación Processing<sup>18</sup>.



Figura 3.2: Logo Arduino  
Fuente: (Arduino, 2016)

#### 3.1.2.1 Entorno de desarrollo

Para programar la placa de Arduino es necesario descargarse de su página web el entorno de desarrollo (IDE) Figura 3.3. Se dispone de versiones para Windows y para MAC, así como las fuentes para compilarlas en LINUX.

---

<sup>18</sup>**Processing:** es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización.

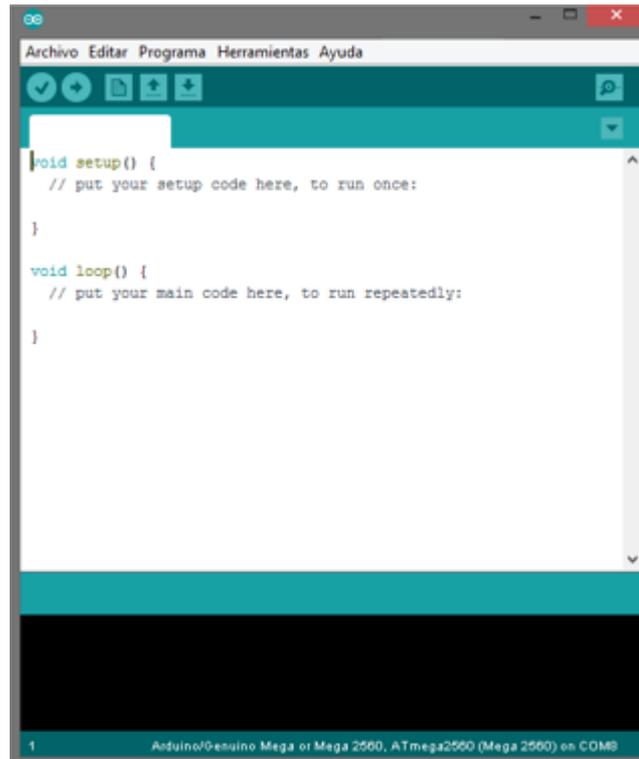


Figura 3.3: Entorno Arduino

Lo primero que se realizara para comenzar a trabajar con el entorno de desarrollo de Arduino, es configurar las comunicaciones entre la placa Arduino y el PC. Para ello se abre en el menú "Tools", la opción "Serial Port". En esta opción se selecciona el puerto serie al que está conectada la placa de Arduino.

### 3.1.2.2 Estructura básica de un programa

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes: setup y loop. Setup() constituye la preparación del programa y loop() es la ejecución.

En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar el pinMode (p. ej. si un determinado pin digital es de entrada o salida) e inicializar la comunicación serie.

La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

Ejemplo:

```
void setup() {  
  pinMode(pin, OUTPUT); // Establece 'pin' como salida  
}  
  
void loop() {  
  digitalWrite(pin, HIGH); // Activa 'pin'  
  delay(1000); // Pausa un segundo  
  digitalWrite(pin, LOW); // Desactiva 'pin'  
  delay(1000);  
}
```

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing que es similar a C++.

### **Las razones del porqué se optaron por Arduino, son:**

- Barato: Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- Entorno de programación simple y clara: El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también.

## **3.2 Hardware**

El Hardware son los componentes físicos: periféricos y equipos que componen un sistema informático, en otras palabras, se refiere a los objetos que se pueden tocar y ver, como una placa de circuito, circuitos integrados, y otros aparatos electrónicos, para la implementación.

### **3.2.1 Arduino**

La plataforma Arduino es una plataforma open-hardware basada en una sencilla placa con entradas y salidas, analógicas y digitales. El elemento principal el

microcontrolador Atmegaxxx, es un chip sencillo y de bajo coste, que permite el desarrollo de múltiples diseños.

Luego de analizar algunas tarjetas de Arduino, se optó por la tarjeta Mega2560, a continuación, tenemos algunas características de las tarjetas Arduino Tabla 3.1.

Tabla 3.1: Características de algunas tarjetas de Arduino

Característica de ARDUINO	UNO	Mega 2560	Leonardo	DUE
Tipo de microcontrolador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales de E/S	14	54	20	54
Entradas analógicas	6	16	12	12
Salidas analógicas	0	0	0	2 (DAC)
Memoria de programa (Flash)	32 Kb	256 Kb	32 Kb	512 Kb
Memoria de datos (SRAM)	2 Kb	8 Kb	2.5 Kb	96 Kb
Memoria auxiliar (EEPROM)	1 Kb	4 Kb	1 Kb	0 Kb

Fuente: (Arduino, 2016)

### 3.2.1.1 Tarjeta de adquisición Arduino Mega 2560 R3

El Arduino Mega está basado en el microcontrolador ATmega2560. Incorpora todo lo necesario para que el microcontrolador trabaje; se puede conectar con un cable USB a la PC o si no con una fuente de alimentación externa. Esta tarjeta es compatible con la mayoría de los shields diseñados para Arduino: Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor, el Arduino Mega, ahora utiliza un microcontrolador ATmega8U2 en vez del chip FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows), además, ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc. (Arduino, 2016)

Las características: de la tarjeta Arduino Mega 2560 R3, se indican en la Tabla 3.2

Tabla 3.2: Características Arduino Mega

Microcontrolador	ATMega2560
Voltaje operativo	5V
Voltaje de entrada	7-12V
Voltaje de entrada (límite)	6-20V
E / S digitales prendedores	54 (15 proporcionan salida PWM)
Pines de entrada analógica	16
Corriente continua para Pin I / O	20 mA
Corriente CC para Pin 3.3V	50 mA
Memoria flash	256 KB( 8 KB ocupa el bootloader) <sup>19</sup>
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Fuente: (Arduino, 2016)

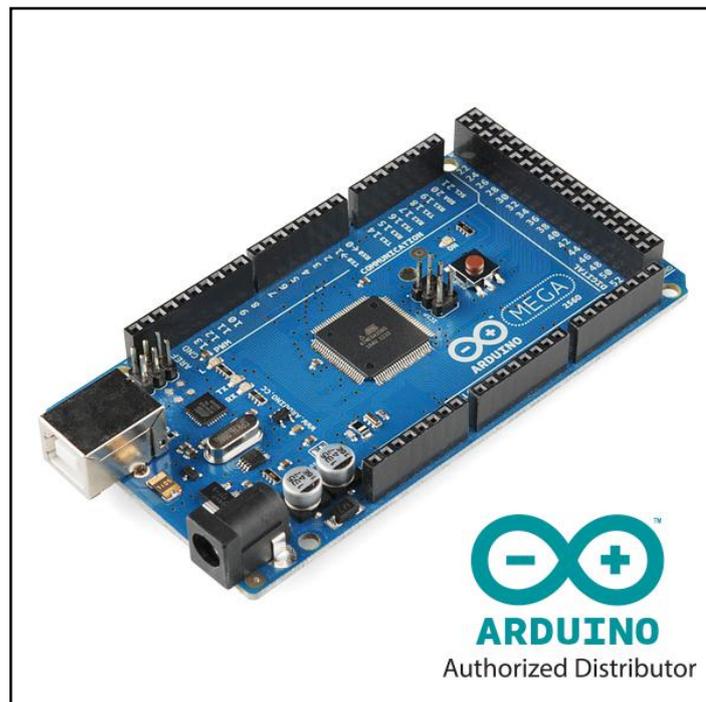


Figura 3.4: Tarjeta Arduino Mega 2560 R3

Fuente: <http://arduino.cl/arduino-mega-2560/>

<sup>19</sup> **bootloader** o cargador de arranque: es un programa que se encarga de cargar y ejecutar el sistema operativo luego de completar varios tests automáticos.

### 3.2.1.2 Arduino Ethernet Shield

El Arduino Ethernet Shield Figura 3.5, permite a una placa de Arduino conectarse a internet en minutos a través de un cable Ethernet RJ45.

Esta tarjeta tiene una ranura para una tarjeta mini-SD, que permite almacenar archivos a través de la red. Es compatible con Arduino UNO y Mega (utilizando la librería Ethernet), a continuación, presentamos las características de la tarjeta Arduino Ethernet Shield Tabla 3.3.

Tabla 3.3: Características de la Tarjeta Arduino Ethernet Shield

Voltaje operativo	5V
Controlador Ethernet	W5100 con tampón interno de 16K
Velocidad de conexión	10/100mb
Compatible	IEEE802.3af
Voltaje de entrada	36 a 57V
salida	9V
Aislamiento	1500V
Rizado de salida y ruido bajos	100mVpp
Controlador Ethernet	W5100 con tampón interno de 16K

Fuente: (Arduino, 2016)

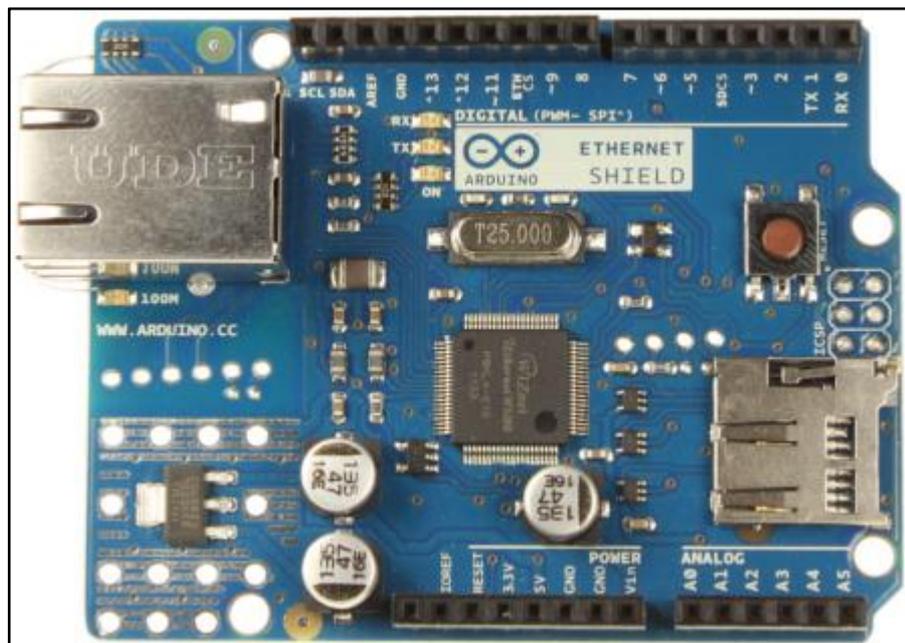


Figura 3.5: Tarjeta Arduino Ethernet Shield

### 3.2.1.3 Módulo Relé de 4 Canales

El módulo relé de Arduino conmuta cargas de potencia, tales como lámparas eléctricas, motores, etc. Como se puede ver en la Figura 3.6, la tarjeta tiene pines para VCC, GND y 4 entradas (In). Hay que tener en cuenta que los relés se activan en nivel bajo y se desactiva en nivel alto.



Figura 3.6: Modulo Relé de 4 canales

Esta tarjeta tiene un LED para cada canal, que indica si se activó cada relé. En la figura 3.7 se observa que los circuitos de control y de carga, se encuentran aisladas mediante opto acopladores, para minimizar el ruido percibido por el circuito de control, mientras se realiza la conmutación de la carga.

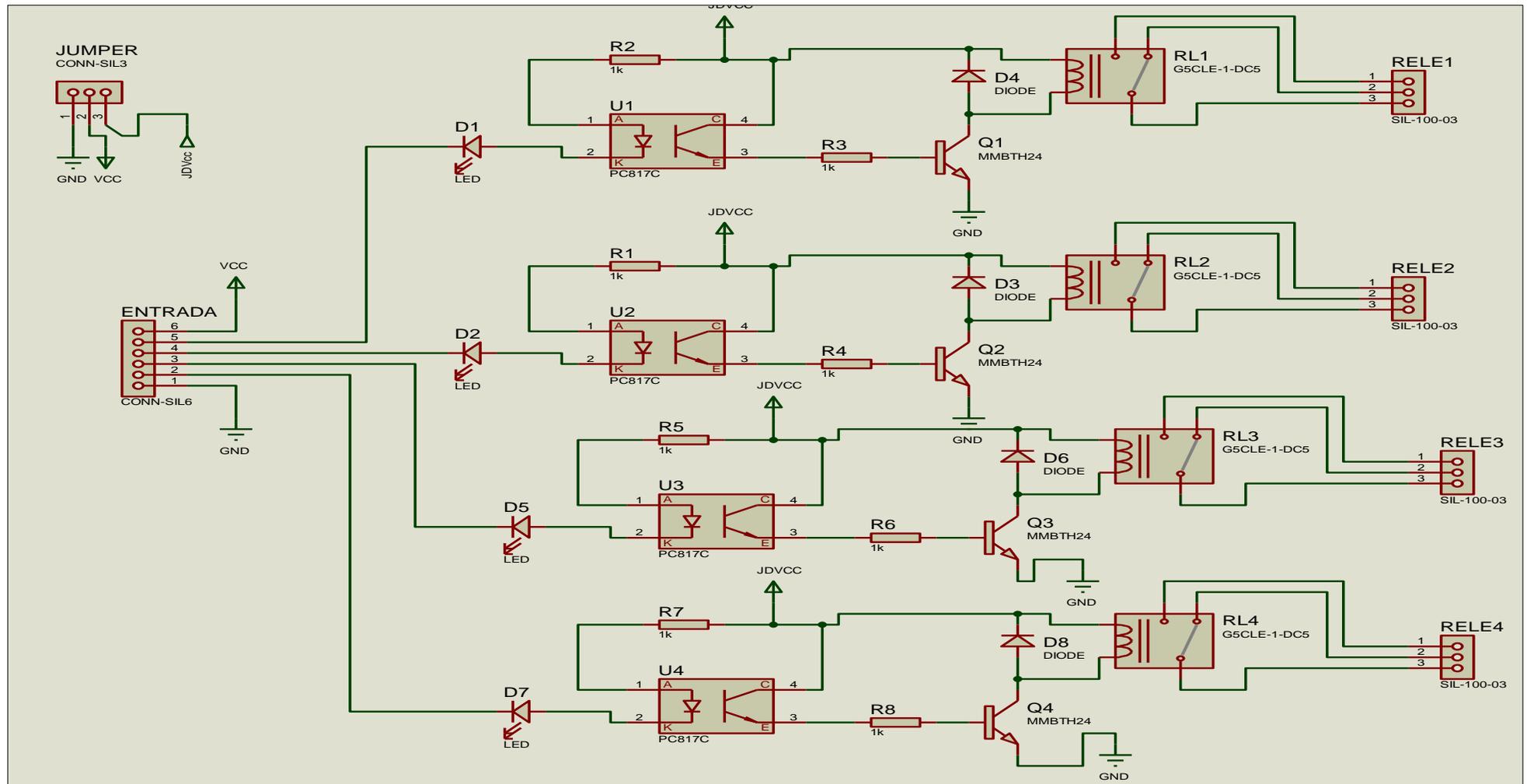


Figura 3.7: Esquema del módulo relé 4 canales

### 3.3 Sensores

Un sensor es un objeto capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas

Un sensor consta de algún elemento sensible a una magnitud física (como por ejemplo la intensidad o color de la luz, temperatura, presión, magnetismo, humedad) y debe ser capaz, por su propias características, o por medio de dispositivos intermedios, de transformar esa magnitud física en una señal eléctrica, que se pueda alimentar en un circuito que la utilice directamente, o sino en una etapa previa que la condicione (amplificando, filtrando, etc.), para que finalmente se la pueda utilizar en el sistema de control. (Carletti, 2016)

Características de un sensor

- *Rango de medida:* dominio en la magnitud medida en el que puede aplicarse el sensor.
- *Precisión:* es el error de medida máximo esperado.
- *Sensibilidad de un sensor:* suponiendo que es de entrada a salida y la variación de la magnitud de entrada.
- *Resolución:* mínima variación de la magnitud de entrada que puede detectarse a la salida.
- *Rapidez de respuesta:* puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- *Derivas:* son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, tales como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- *Repetitividad:* error esperado al repetir varias veces la misma medida.

Los sensores que se emplearon fueron los siguientes, sensor de humedad de suelo FC-28 y sensor de temperatura ds18b20 debido a sus características específicas necesarias

para la medición de las variables que consideramos para la implementación del proyecto.

### 3.3.1 Sensor de Humedad FC-28

El módulo FC-28, es un sensor de humedad de suelo, que utiliza la conductividad entre dos terminales, para determinar ciertos parámetros relacionados al agua, líquidos y humedad.

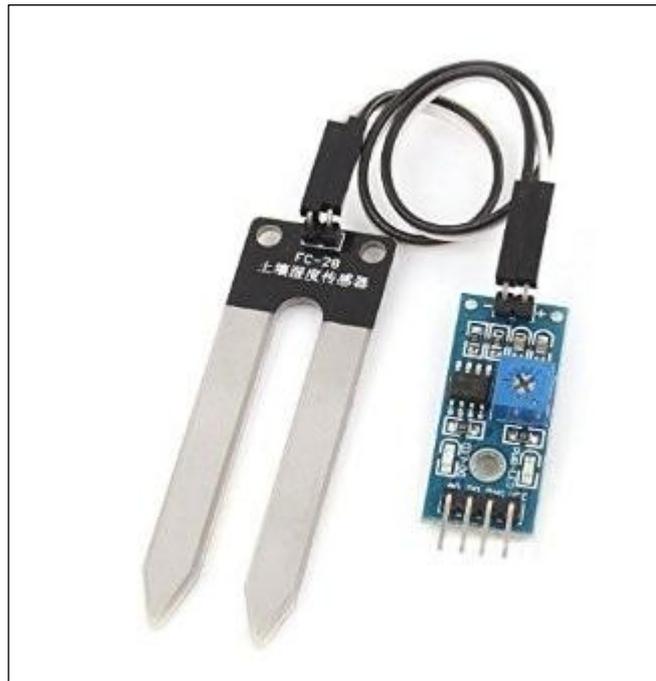


Figura 3.8: Sensor de Humedad FC-28

Fuente: ( $\mu$ ConnectME, 2016)

El sensor FC-28 mostrado en la Figura 3.8, consiste en dos placas separadas entre sí a distancia determinada. Ambas placas están recubiertas de una capa de material conductor. Si existe humedad en el suelo, se creará un puente de conducción entre una punta y otra, lo que será detectado por un circuito de control con un amplificador operacional LM393, que será el encargado de transformar la conductividad registrada a un valor analógico, que podrá ser procesado por la tarjeta ARDUINO.

Habrán dos tipos de salidas, una analógica y una digital. La salida digital entregará un pulso bajo cuando haya conductividad suficiente entre cada una de las puntas del sensor de humedad. El umbral de disparo se puede establecer ajustando el potenciómetro del circuito de control. Para la salida analógica, el nivel de voltaje

dependerá directamente de cuanta humedad haya en el suelo, es decir, dependiendo de cuanta conductividad (producto del agua en el suelo) haya entre las puntas del módulo, así variará el valor entregado. ( $\mu$ ConnectME, 2016)

### **Características del sensor de humedad FC-28**

- Voltaje de 3,3 a 5V cc
- Salidas analógica y comparadora
- Ajuste de sensibilidad
- Dimensiones del sensor 6x-20mm contactos 45mm
- Dimensión del comparador 30x14mm

### **3.3.2 Sensor de temperatura DS18B20**

El sensor de temperatura digital DS18B20 ofrece una Resolución de 9 bits a 12 bits programable

El sensor DS18B20 figura 3.9, se comunica a través de un bus 1-Wire, que por definición sólo requiere una línea de datos y tierra para la comunicación con un microprocesador central. Tiene un rango de temperatura de funcionamiento de  $-55^{\circ}\text{C}$  a  $+125^{\circ}\text{C}$  y tiene una exactitud de  $\pm 0,5^{\circ}\text{C}$  en el rango de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ .

Cada DS18B20 tiene un código de serie de 64 bits único, lo que permite que varios DS18B20s funcionen al mismo tiempo.

Una de las técnicas utilizadas se denomina RTD (Resistance temperature detector), que consiste en detectar los cambios de resistencia de un resistor interno debido a la temperatura (es una medición indirecta). (Maxim Integrated, 2008)

PINES:

GND (0V);

VDD (5V);

DQ (Pin de datos, es por donde se realizará la lectura)

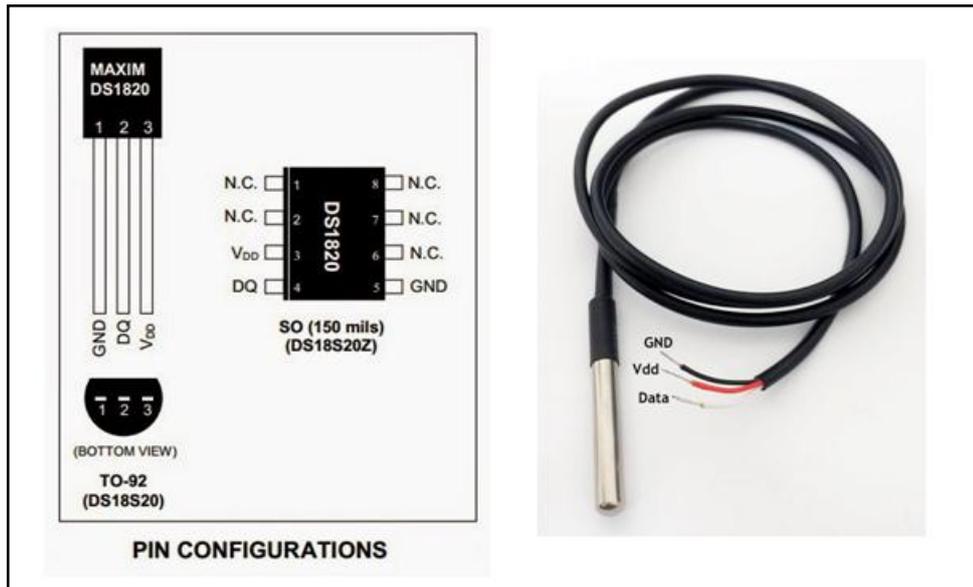


Figura 3.9: Sensor de Temperatura Ds1820

Fuente: (Maxim Integrated, 2008)

### 3.4 Actuador

Un actuador es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía. (Corona Ramírez, Abarca Jiménez, & Mares Carreño, 2014)

Los actuadores por lo general se clasifican en dos grupos:

- Por el tipo de energía utilizada: actuador neumático, hidráulico y eléctrico
- Por el tipo de movimiento que generan: actuador lineal y rotatorio.

#### 3.4.1 Electroválvula

Es un dispositivo que se intercalan en las tuberías de los suministros de agua y gas, con la misión de cerrar o abrir el conducto de dichos fluidos, pertenecen al grupo de todo o nada. La electroválvula se controla variando la corriente que circula a través de un solenoide. Esta corriente al circular por el solenoide, genera un campo magnético que atrae un émbolo móvil. Al finalizar el efecto del campo magnético, el émbolo vuelve a su posición por efecto de la gravedad o por resorte. (Tobajas García, 2012)

Existen dos tipos de electroválvulas:

- **Normalmente abiertas:** Deja pasar el suministro de fluidos siempre y cuando no haya tensión.
- **Normalmente cerradas:** Estará cortado el suministro de fluidos siempre y cuando no haya tensión.



Figura 3.10: Electroválvula

#### **Características de la Electroválvula:**

- 1/2" Nominal NPS
- Presión de trabajo: 0.02 Mpa – 0.8 Mpa
- Temperatura de trabajo: 1 C – 75 C
- Tiempo de respuesta (apertura):  $\leq 0.15$  seg
- Tiempo de respuesta (cierre):  $\leq 0.3$  seg
- Voltaje de actuación: 12VDC
- Vida útil:  $\geq 50$  millones de ciclos
- Peso: 4.3 oz
- Dimensiones: 3" x 2.25" x 2"

### 3.4.2 Aspersor

Un aspersor o sorpersor como el que se muestra en la Figura 3.11, es un dispositivo mecánico que en la mayoría de los casos el flujo líquido presurizado, lo transforma en rocío, asperjándolo para fines de riego.

Un chorro de agua asperjado es un conjunto aleatorio de gotas de agua que son expulsadas de un medio presurizado a otro con presión atmosférica, donde este conjunto de agua pulverizada guarda direcciones similares y velocidades diferentes (depende de los tipos de boquilla utilizados) con el único objetivo de conseguir una cortina de agua lanzada al espacio de la manera más uniforme posible. (Jardines Modernos, 2012)



Figura 3.11: Aspersor de Agua

Fuente: <https://humusyfertilizantes.com/producto/aspersor-de-impacto-plastico/>

#### Características y ventajas:

- Fabricado en plástico ABS de gran calidad.
- Arco de riego circular o sectorial.
- Ajuste del sector de riego de 30° a 360°.
- Diámetro de aspersión: 26 m.
- Área de riego: 530 m<sup>2</sup>.
- El aspersor trabaja a partir de 1 bar de presión de agua.

### 3.5 Diseño del proyecto

Para el diseño se tomó en cuenta varias alternativas que se podría usar. La opción más conveniente para nuestro proyecto fue trabajar con las plataformas de Blynk y Arduino.

Para poder utilizar la aplicación por lo general se utiliza el servidor de Blynk de la nube. La otra alternativa es instalar un servidor de Blynk localmente y restringir el acceso a la red, por lo que nadie puede tener acceso a nuestros datos excepto el administrador.

De este modo podemos utilizar un servidor externo o un local.

#### 3.5.1 Servidor local

El servidor de Blynk es de código abierto Netty<sup>20</sup> basado en un servidor java, este tiene la función del envío de mensajes entre la aplicación Blynk y varios tableros de microcontroladores como ejemplo Arduino.

##### **Ventajas de usar un servidor local:**

- Mayor seguridad
- Mejor estabilidad
- Menor latencia
- Máxima privacidad

Para hacer que funcione el servidor local podemos instalar en una PC con sistema Windows o en una tarjeta Raspberry. En nuestro caso tomamos la segunda alternativa, por su menor costo.

---

<sup>20</sup> **Netty** : Es un marco de cliente-servidor NIO que permite el desarrollo rápido y sencillo de aplicaciones de red, tales como servidores de protocolo y clientes.

### 3.5.1.1 Raspberry pi 2B

#### 3.5.1.1.1 Características de la Raspberry pi 2 B

La tarjeta de Raspberry pi 2 es una mini computadora de bajo costo, que fue desarrollada con el objetivo de estimular la enseñanza en ciencias de la educación.

El Raspberry pi puede utilizar varios sistemas operativos, pero en su mayor parte utilizan los sistemas GNU/Linux. Por su parte para este proyecto utilizamos Raspbian una distribución derivada de Debian<sup>21</sup> que está optimizada para el hardware de Raspberry Pi.

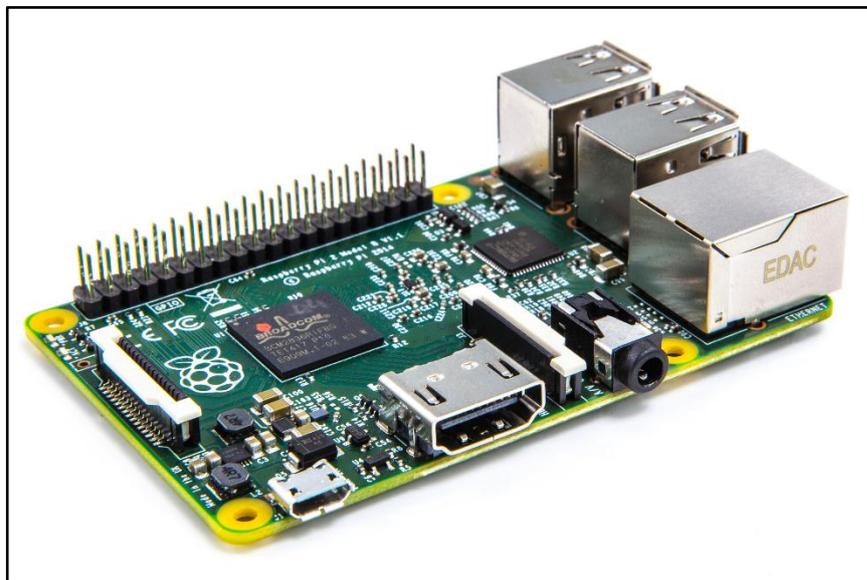


Figura 3.12: Tarjeta Raspberry pi2

Fuente:<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/#buy-now-modal>

Una de las ventajas de las tarjetas Raspberry pi2 Figura 3.12, son su reducido tamaño, su alta capacidad de proceso en relación a su tamaño. En la tabla 3.4 podemos observar las características principales de esta pequeña tarjeta Raspeberry pi2 Modelo B.

---

<sup>21</sup> **Debian** : Sistema operativo libre

Tabla 3.4: Características de Raspberry Pi2

<b>RASPERRY PI 2 MODELO B</b>	
<b>SOC<sup>22</sup></b>	Broadcom BCM2836
<b>CPU</b>	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz
<b>GPU</b>	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
<b>RAM</b>	1 GB LPDDR2 SDRAM 450 MHz
<b>USB 2.0</b>	4
<b>SALIDAS DE VÍDEO</b>	HDMI 1.4 @ 1920x1200 píxeles
<b>ALMACENAMIENTO</b>	microSD
<b>ETHERNET</b>	Sí, 10/100 Mbps
<b>TAMAÑO</b>	85,60x56,5 mm
<b>PESO</b>	45g
<b>CONSUMO</b>	5v, 900mA

Fuente: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>

### 3.5.1.1.2 Configuración de la Raspberry pi 2 B

Para la configuración, se descarga desde la página [www.raspberrypi.org](http://www.raspberrypi.org) el sistema operativo Rasbian. (Figura3.13)

<sup>22</sup> **SOC:** System-on-chip

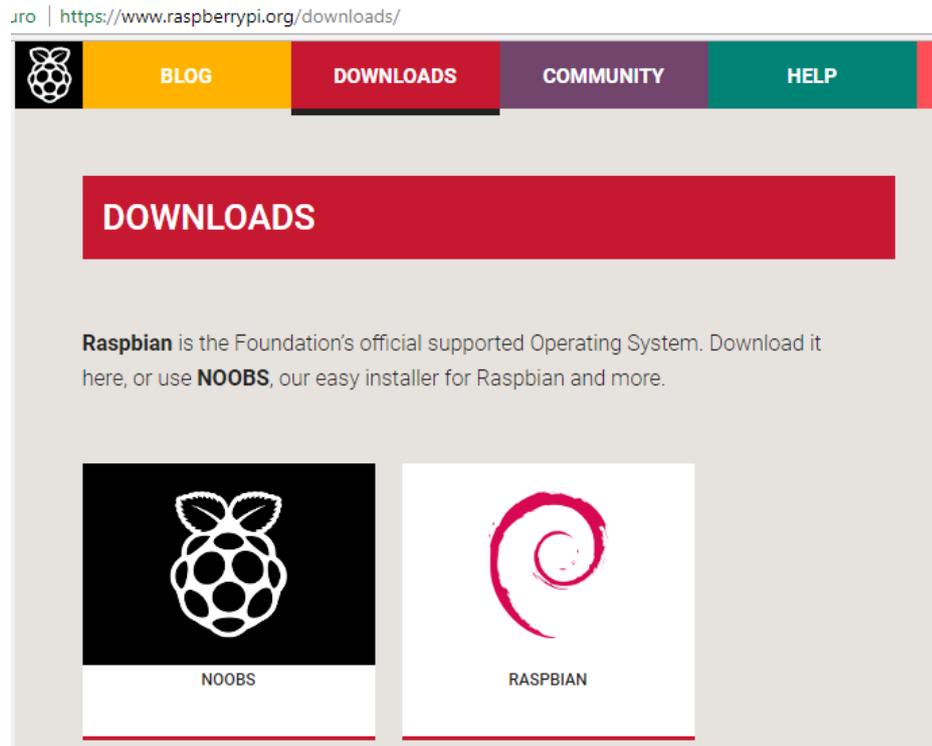


Figura 3.13: Pagina para descargar sistema operativo Raspbian.

Fuente: <https://www.raspberrypi.org/downloads/>

Una vez instalado el sistema operativo en la tarjeta Raspberry pi, se inicia sesión a través de SSH<sup>23</sup> utilizando PuTTY.<sup>24</sup> (Figura 3.14)

<sup>23</sup> **SSH** : (Secure SHell) es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente.

<sup>24</sup> **PuTTY**: Es un cliente SSH y Telnet con el que podemos conectarnos a servidores remotos iniciando una sesión en ellos que nos permite ejecutar comandos

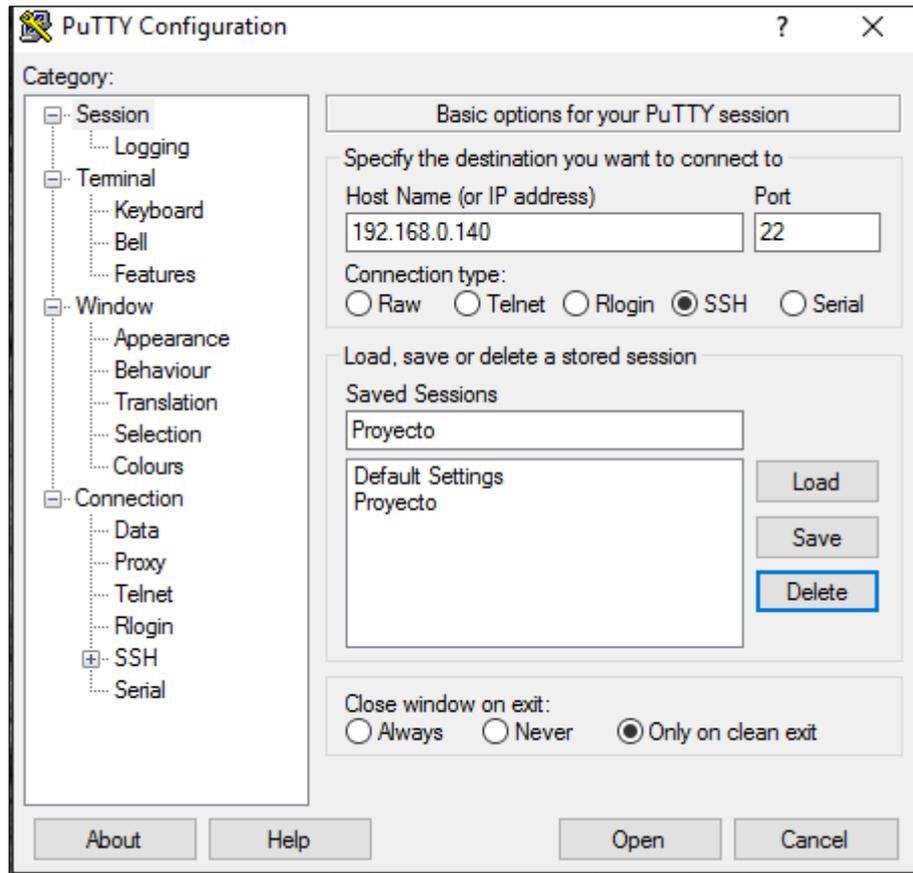


Figura 3.14: Configuración PuTTY

Luego, hay q ingresar nombre de usuario y contraseña.



Figura 3.15: Ingreso al terminal de Raspberry Pi

Luego se verifica si está instalada java y que versión tiene, caso contrario hay que actualizarla.

```
pi@raspberrypi:~ $ java -version
java version "1.8.0_65"
```

Figura 3.16: Versión java

El siguiente paso, es poner una dirección IP fija en la tarjeta Raspberry pi2.

Luego se ejecuta el comando.

```
sudo nano -w /etc/network/interfaces
```

Ahora, saldrá una pantalla, donde se debe modificar la palabra “dhcp” por “static” y poner la IP que es necesario que quede fija.

```
auto eth0
iface lo inet loopback

iface eth0 inet static
address 192.168.0.140
netmask 255.255.255.0
gateway 192.168.0.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Figura 3.17: Configuración de la IP fija en la tarjeta Raspberry PI2

Después de cambiar la IP fija, se guarda los cambios a continuación se procede a reiniciar la tarjeta Raspberry pi2. Luego se ejecuta el servidor Blynk en la tarjeta utilizando el comando:

```
java -jar server-0.20.0.jar -dataFolder /home/pi/Blynk
```

Para hacer el lanzamiento del servidor automáticamente cada vez que se inicie la Raspberry pi2, se cuelga o si se reinicia, encontrar el archivo /etc/init.d/rc.local y añadir una línea:

```
java -jar /home/pi/server-0.20.0.jar -dataFolder /home/pi/Blynk &
```

### 3.5.1.2 Dispositivo móvil

Una vez configurado el servidor, se pasa a configurar el dispositivo móvil, para que se inicie con servidor local.

El primer paso es descargar la aplicación de Google Play Store, una vez descargado e instalado se procederá a abrir la aplicación.

Al abrir Blynk Figura 3.18 se debe registrar el usuario, poniendo un correo electrónico y una clave de acceso, se selecciona que queremos utilizar, si servidor de la nube Blynk o en este caso un servidor local. Seleccionada la segunda opción, se pone la dirección IP del servidor local y el puerto de enlace, que en este caso se coloca la dirección IP del servidor 192.168.0.140 y el puerto (debe ser de 8443 por defecto).

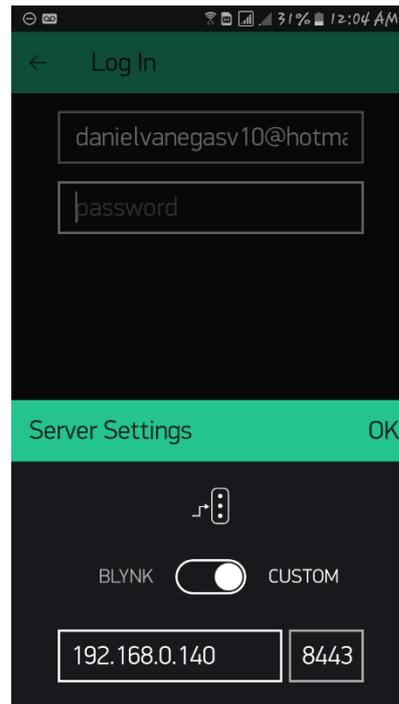


Figura 3.18: Configuración dispositivo móvil para comunicación con servidor local

### 3.5.1.3 Arduino

A continuación, se configura la tarjeta de Arduino, para que se pueda conectar al servidor local.

Al configurar Arduino, lo primero que se debe hacer es instalar las bibliotecas de Blynk en Arduino que lo descargaremos de <https://github.com/blynkkk/blynk-library/releases/tag/v0.4.4>.

Una vez instalado se prosigue a abrir el software de Arduino Figura 3.19 y seleccionamos abrir un ejemplo de Blynk.

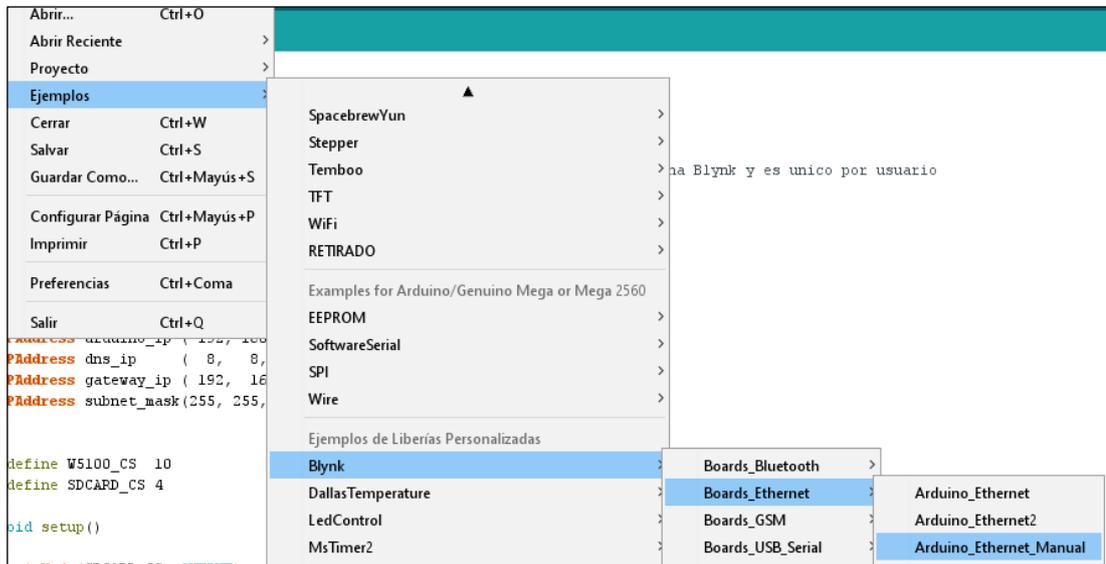


Figura 3.19: Ejemplo Arduino

Una vez abierto el ejemplo figura 3.20, se procede a configurar la red interna, donde se coloca la dirección IP del servidor 192.168.0.140, también se debe configurar la puerta de enlace y asignarle una IP a la tarjeta Arduino.

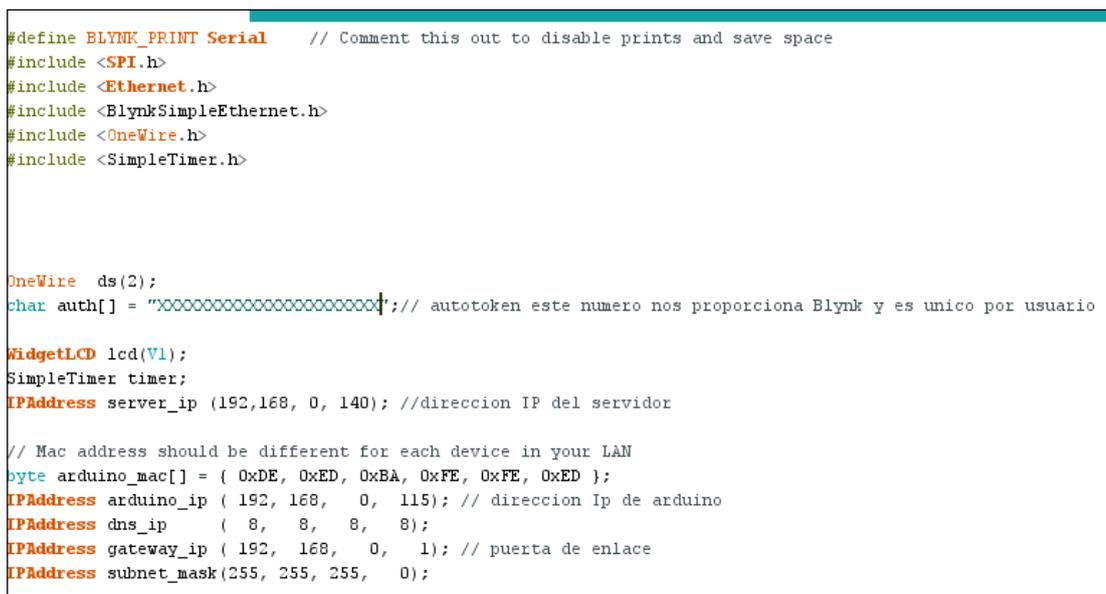


Figura 3.20: Configuración de Arduino con el servidor Blynk

### 3.5.1.4 Router

Una vez configurado el servidor local Blynk, la tarjeta Raspberry y la tarjeta de Arduino, se procede a configurar el router.

Para nuestro proyecto utilizamos el router D-Link DIR-610 Figura 3.21.



Figura 3.21: Router Wireless D-link Dir-610 150mbps 4 Puertos Wifi IPv6

Se procede a conectar todos los equipos al router. Desde una PC se entra al terminal y se escribe el comando ipconfig y así se observara la puerta de enlace del router que para este caso es 192.168.0.1, con este dato obtenido se coloca en la barra de cualquier navegador de internet y da el acceso al router.

Luego se configura una IP fija para el servidor, para que ningún otro dispositivo pueda ocupar ese puerto.

LISTA DE RESERVAS DHCP					
Enable	Nombre de host	Dirección IP	Dirección MAC		
<input checked="" type="checkbox"/>	raspberrypi	192.168.0.140	b8:27:eb:5d:53:ca		

Figura 3.22: Muestra de reserva de IP para la tarjeta Raspberry pi 2

Ahora se observa en el panel de conexión Figura 3.22, que equipos están conectadas a nuestra red privada y que IP están asignadas.

**LAN INALÁMBRICA**

**Radio inalámbrica** : Activada  
**Dirección MAC** : c8:d3:a3:1a:6f:6c  
**Modo 802.11** : Mezcla de 802.11n, 802.11g y 802.11b  
**Anchura de canal** : 20/40MHz  
**Canal** : 10  
**Nombre de red (SSID)** : Riego  
**Configuración protegida Wi-Fi** : Activada/Configurado  
**Seguridad** : WPA/WPA2-PSK  
**Estado WDS** : Desactivada

---

**ORDENADORES LAN**

Dirección MAC	Dirección IP	Nombre (si hubiera)
de:ed:ba:fe:fe:ed	192.168.0.115	Arduino
b8:27:eb:5d:53:ca	192.168.0.140	Raspberry pi
78:c3:e9:3d:4e:1c	192.168.0.103	Dispositivo Movil
74:e6:e2:53:90:a9	192.168.0.100	FANTASTIC PC

Figura 3.23: Puertos conectados al Router

### 3.5.2 Servidor Blynk Internet

Al utilizar el servidor de la nube de Blynk Figura 3.24, se puede conectar y controlar nuestro proyecto desde cualquier parte del mundo a través del internet.

Una desventaja de este método es la seguridad que tiene el sistema, ya que hay la posibilidad que otras personas puedan hackear y desestabilizar el proyecto, otra cuestión es que se depende demasiado del operador de internet, para que pueda funcionar en tiempo real y sea constante.

La configuración del dispositivo móvil para funcionar casi igual que con el servidor local, hay que registrarse, lo único que cambia es la forma de conectarse.

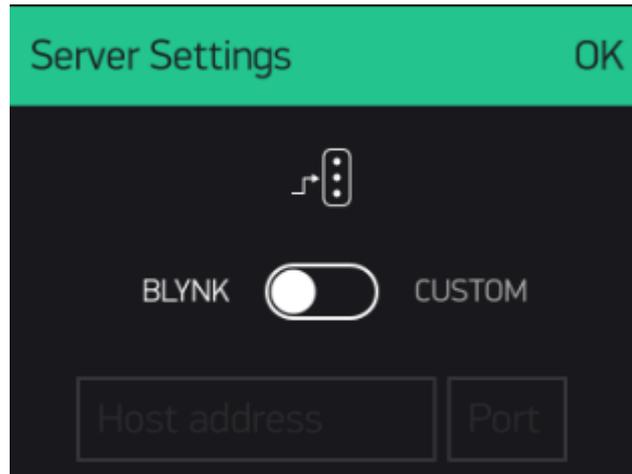


Figura 3.24: Servidor Blynk Nube

### 3.5.3 Programación de control de riego.

Una vez establecido la comunicación de nuestro Arduino ya sea con servidor local o servidor Blynk de la nube, se procede a establecer las entradas y salidas que son necesarios para implementar el proyecto:

Pines que se necesitan de la tarjeta Arduino:

- Dos salidas digitales (pines 4 y 5 del Arduino) para activar los relés, que controlan las electroválvulas.
- Una entrada digital para la lectura del sensor de temperatura (pin 2 del Arduino)
- Dos entradas analógicas (Pines A0 y A1) para los sensores de humedad uno del patio y otro del jardín.

En Anexo 1 consta la programación de Arduino para este proyecto. En el diagrama de flujo presentado en la Figura 3.25, se observa la secuencia de operaciones para codificar las instrucciones de Arduino que permite controlar el sistema de riego.

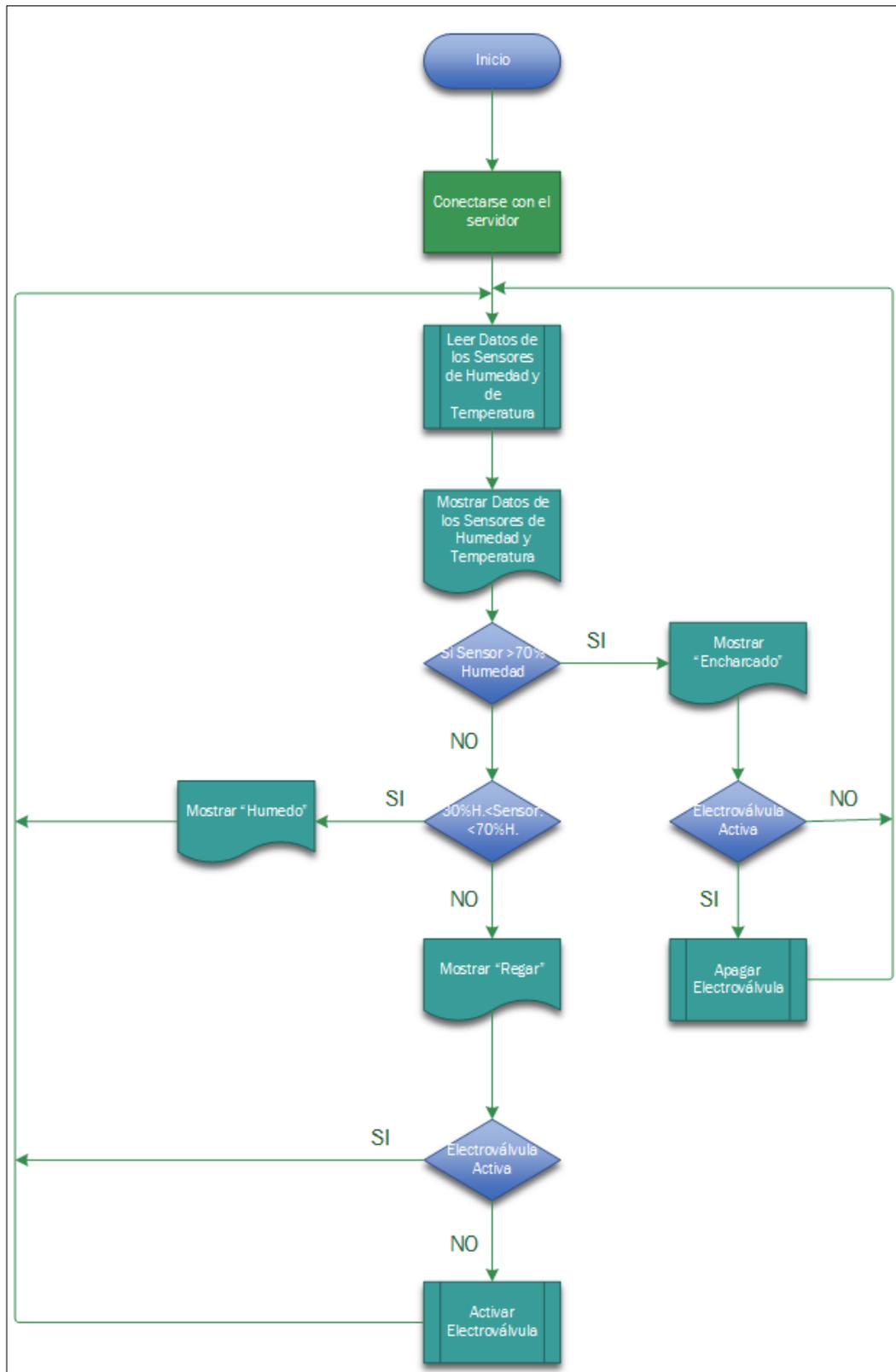


Figura 3.25: Diagrama de flujo

En cuanto a la programación y configuración de Blynk desde el móvil Figura 3.26, se empieza digitando el nombre del proyecto, luego hay que escoger el hardware y por último la forma de conexión.

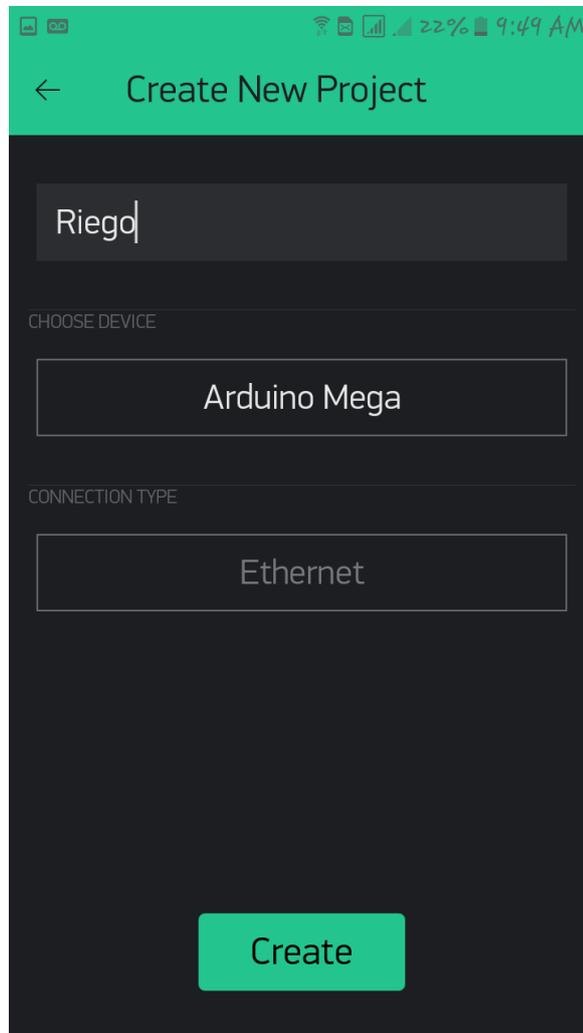


Figura 3.26: Creación de proyecto

Luego de configurar el dispositivo móvil, se crea un código de AUTHTOKEN que nos servirá para poner en el IDE de Arduino, que permite establecer la conexión.

Ahora, al abrir Blynk se tiene la pantalla de la figura 3.27 y se precede a escoger un widget de botón de una lista desplegable de comandos, cabe recalcar que al iniciar una cuenta en Blynk se otorga un nivel de energía y cada widget tiene un precio de energía y si se requiere incrementar el número de widgets su puede realizar la compra de energía.

Ahora se configurará un widget de botón, que corresponde a la salida asignada, es decir si es un pin analógico, virtual o digital.

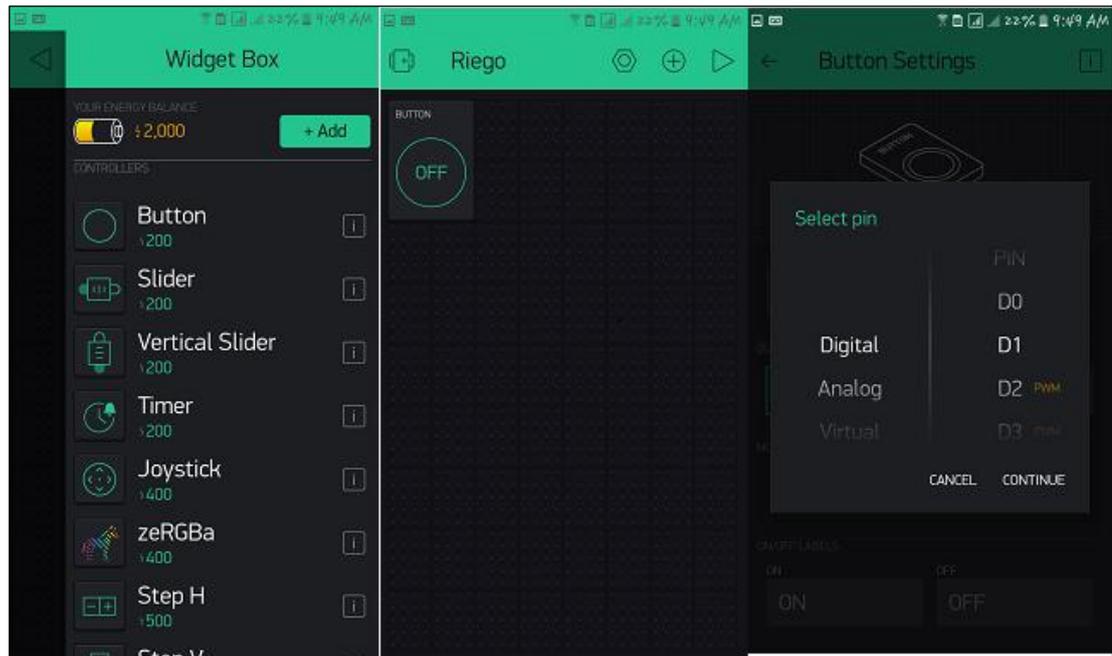


Figura 3.27: Pasos de Blynk

En la programación de un widget Figura 3.28, se puede cambiar el color del botón, escribir un nombre para que aparezca en la pantalla, en este caso, cuando esta prendido tiene un cero lógico que permite funcionar el relé que se activa con un cero lógico.

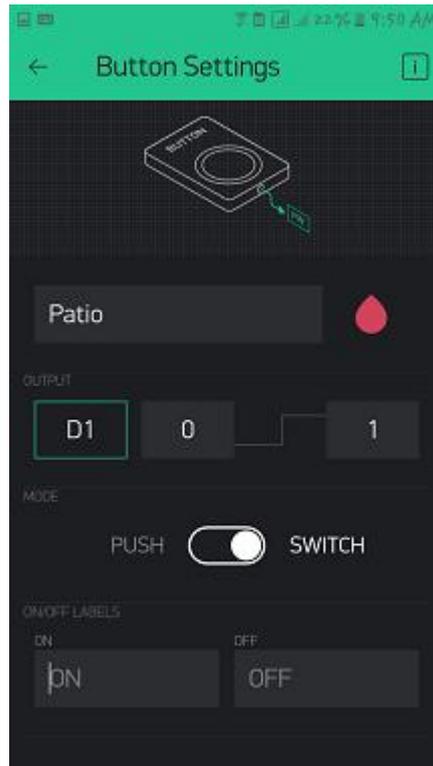


Figura 3.28: Configuración widget botón

De esta forma iremos configurando cada uno de los widget de Blynk, que a su vez iremos programando en Arduino.

En esta aplicación se puede crear un código QR Figura 3.29 y pasar todo el programa de un dispositivo móvil a otro dispositivo móvil.

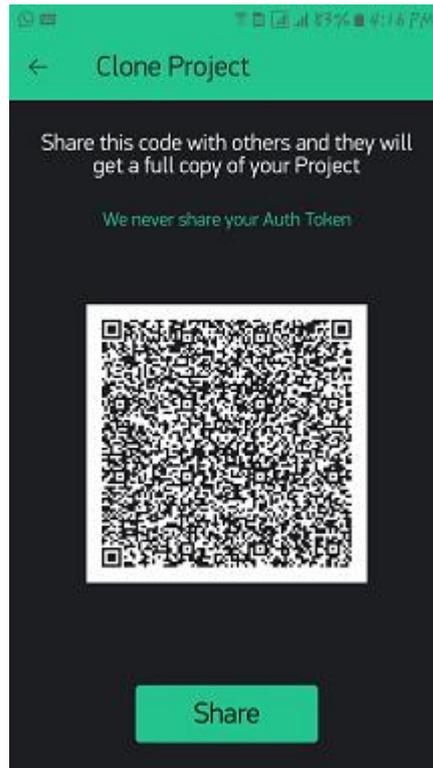


Figura 3.29: Clonar Proyecto

## CAPITULO 4

### 4. IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA DE RIEGO.

#### **Introducción:**

En este capítulo se describe todo el proceso de implementación del sistema de riego en la propiedad donde se realiza el proyecto, se hace las pruebas y evaluaciones pertinentes sobre el sistema.

#### **4.1 Implementación**

La implementación se hizo en una propiedad a 15 minutos de la ciudad de Cuenca, en el sector conocido como Zhullin de la provincia del Cañar canto Azogues. En la Figura 4.1 podemos apreciar la distribución de los espacios en los cuales se van a realizar los diferentes tipos de riego, ya que para la cancha se realizará un riego por aspersión y para la jardinera que bordea la propiedad un riego por goteo.

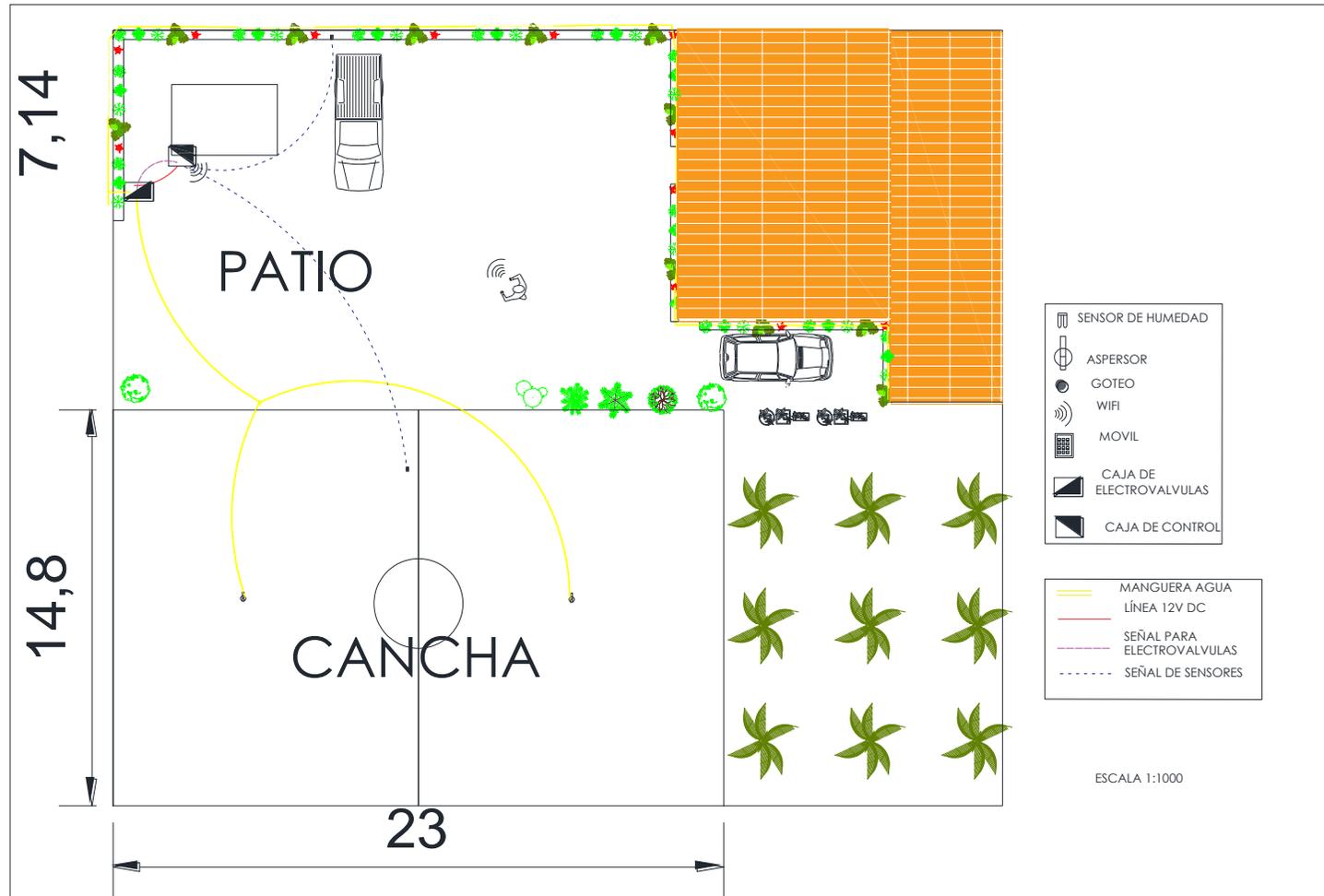


Figura 4.1: Plano del lugar donde se implementó el sistema de riego.

#### 4.1.1 Sistema de riego por aspersión

La implementación del sistema de riego por aspersión, se la realizó en un terreno de aproximadamente de 340m<sup>2</sup> Figura 4.1, que hay una cancha de juego, los materiales utilizados para la implementación del sistema fueron: 20m de manguera de ½ pulgada, 30m de manguera de ¼ de pulgada, 4 acoples para mangueras, de ½ pulgada y una reducción de ½ a ¼ de pulgada, 2 electroválvulas, 15 m de cable # 20 para sensores y 10m de cable #14 para electroválvulas, 2 sensores de humedad y 2 aspersores, se realizó la instalación de la tubería negra para el cableado de alimentación de la electroválvula de accionamiento , la cual va desde la caja de mando hacia la electroválvula que se encuentra situada junto a una llave que está conectada al suministro de red pública de agua, bordeando el terreno en el cual se va a realizar el riego por aspersión. Además de la instalación del sensor de humedad, el cual dará el porcentaje de humedad en el cual el terreno se encuentra en ese momento.



Figura 4.2: Terreno para riego por aspersión

Desde la llave de paso de la red pública de agua se conectó mediante acoples y manguera hacia la electroválvula Figura 4.3, la cual es la encargada de dejar pasar el líquido vital para el riego, a continuación, se procedió a instalar la siguiente etapa mediante acoples y manguera hacia el aspersor Figura 4.4, el cual es el encargado de realizar las labores de riego y la distribución del agua en el terreno dispuesto.



Figura 4.3: Electroválvula



Figura 4.4: Aspersor

La caja de Mando o conexión Figura 4.5, está situada en la entrada de la vivienda, en esta caja se encuentra alojada la tarjeta ARDUINO, la cual comanda por medio de relés a las electroválvulas y adquiere los datos por medio de sensores, esta a su vez está conectada a la tarjeta Raspberry pi2, la cual cumple la función de servidor web

previamente ya configurado y por último el router que es encargado de dar el servicio de Wi-Fi para la conexión por medio de internet para la aplicación.

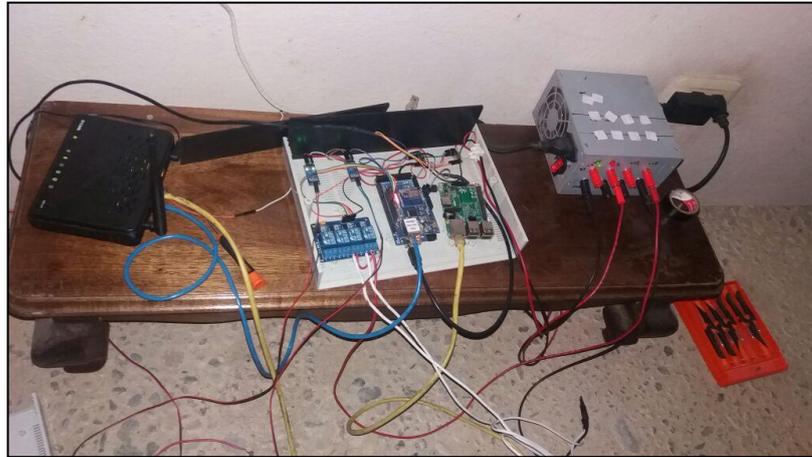


Figura 4.5: Caja de Mando

La tarjeta de Arduino Mega puede comandar varios actuadores y recibir varias señales de sensores, pero para este proyecto se utilizó dos pines digitales para las salidas de las electroválvulas, un pin digital para ingreso de señal de temperatura y dos pines analógicos como ingreso de datos. En la Figura 4.6, se observa la conexión de la tarjeta Arduino con los sensores y los actuadores

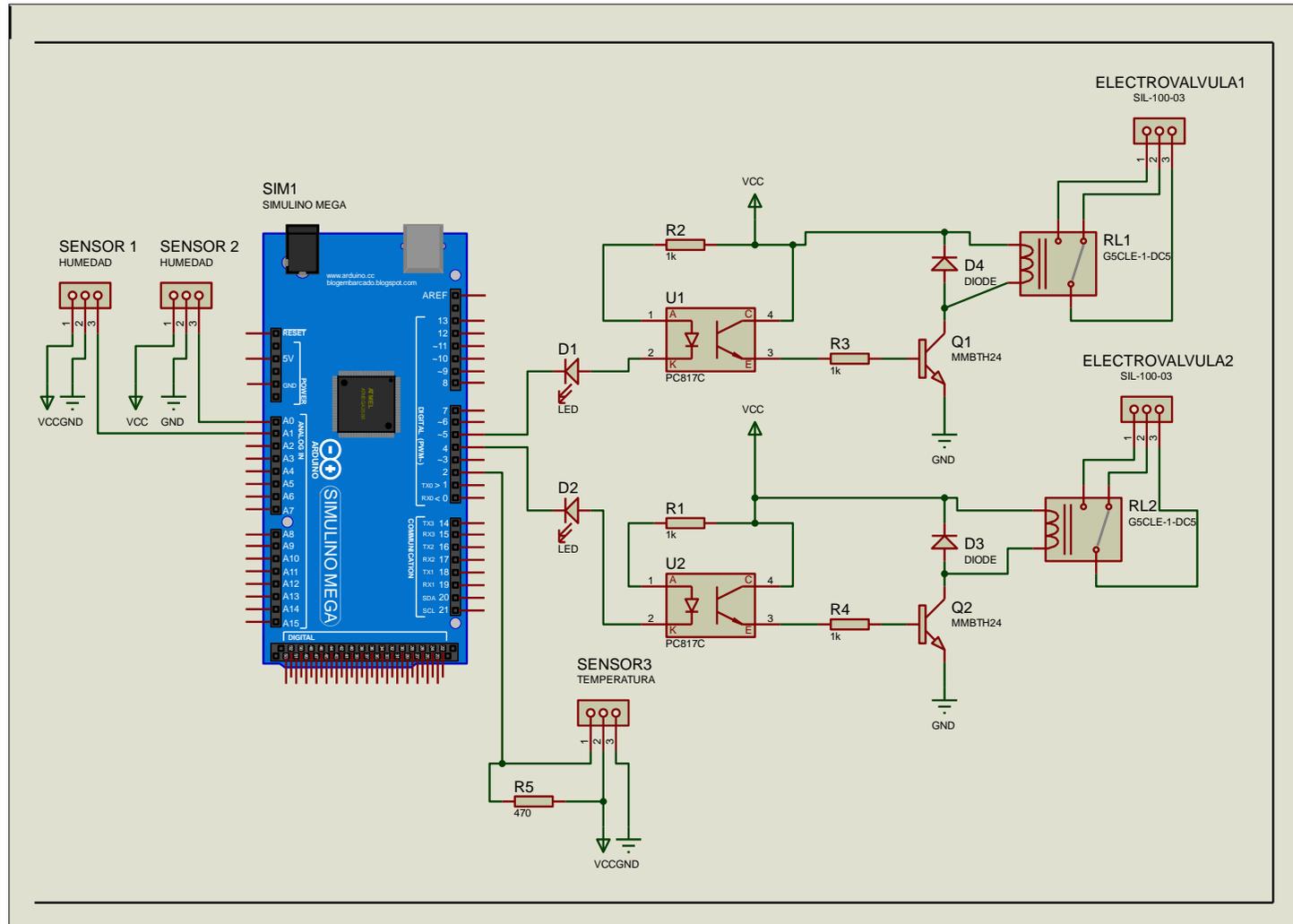


Figura 4.6: Diagrama de la conexión de la tarjeta ARDUINO

#### 4.1.2 Sistema de riego por goteo

El sistema de riego por goteo se lo realizó en una jardinera ubicada en la propiedad cuyas dimensiones son de aproximadamente 30m de largo por 0.30m de ancho Figura 4.7, en la cual están sembradas varias plantas a las que se dota de este sistema de riego y así tener un control adecuado sobre la humedad del suelo y conservar las plantas en buen estado.



Figura 4.7: Jardinera

De la misma manera, desde la llave de paso de la red pública de agua, se conecta por medio de acoples y manguera a la otra electroválvula, esta se conecta mediante tubería hacia una manguera que se sitúa en la jardinera Figura 4.8, la cual tiene varias salidas de agua con sus aditamentos para realizar las labores de riego, mediante el sistema de goteo, la alimentación de dicha electroválvula se realizó por la misma tubería negra previamente instalada para el sistema de riego por aspersión Figura 4.9 y se encuentra situada junto a la llave conectada a red pública de agua.



Figura 4.8: Funcionamiento de riego por goteo



Figura 4.9: Aditamentos para riego por goteo

### 4.1.3 Instalación de Sensores

Para realizar las mediciones de las variables físicas del sistema, se utilizó los sensores tanto de humedad como temperatura, los cuales fueron situados en el terreno, para su medición se colocó los sensores dentro de la propiedad en los lugares que se consideró que serían los óptimos para realizar las mediciones, el sensor de humedad 1 se instaló en el terreno Figura 4.10, en el cual se realiza las labores de riego por aspersión, este se alimenta con un voltaje igual a 5 v, y entrega un valor numérico que luego se convierte en porcentaje de humedad. El sensor de humedad 2 se ubicó en la jardinera junto a la segunda planta Figura 4.11, de igual manera su voltaje de funcionamiento es de 5v. además de estos sensores, se ubicó un sensor de temperatura encapsulado Figura 4.12, el cual está en la propiedad para así obtener datos de la temperatura ambiental,

su voltaje de funcionamiento es de 5v, al igual que los de humedad este sensor está alimentado desde la caja de mando.



Figura 4.10: Sensor de humedad “1” patio



Figura 4.11: Sensor de humedad “2”Jardinera



Figura 4.12: Sensor de temperatura

## 4.2 Pruebas del Sistema.

### 4.2.1 Pruebas de Accionamiento.

Para actuar sobre el sistema dentro de la aplicación de Blynk se configuró dos botones, uno para encendido de riego por aspersión y otro para el riego por goteo. En la Figura 4.13, se muestra una captura de pantalla mediante un dispositivo móvil, en este caso un teléfono celular con acceso a internet, en la cual podemos observar la aplicación con los dos botones.

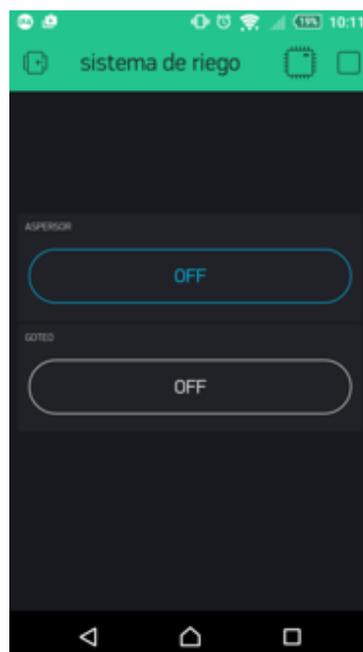


Figura 4.13: Configuración de dos botones de aspersión y goteo

Una vez que se acciona el botón que está en estado OFF, el cual indica aspersor, se realiza el cambio de estado a “Regando “Figura 4.14, y el sistema empieza a funcionar, enviando una señal por medio de la web a la placa Arduino, la cual acciona su salida digital, que conectada a un módulo con relés, es la encargada de accionar la electroválvula, la cual cumple con su función de dejar pasar el agua hacia el aspersor y realizar un riego a 360 grados sobre dicho terreno.

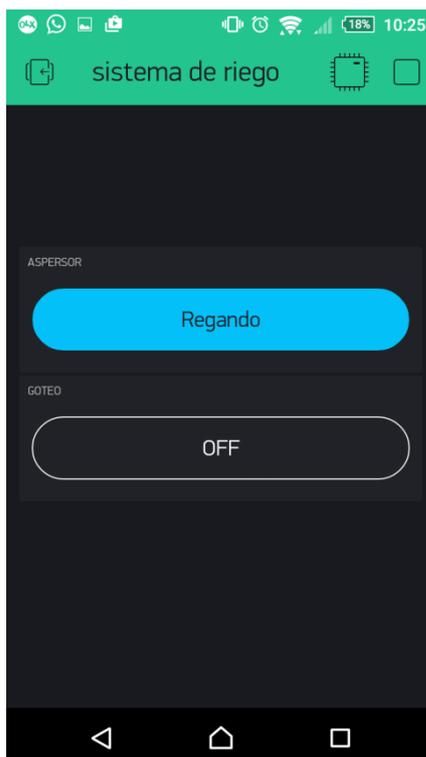


Figura 4.14: Cambio de estado aspersor a Regando

De igual manera al accionar el botón goteo que está en estado OFF se realiza un cambio de estado a “Regando “Figura 4.15, el cual se puede apreciar en la parte inferior, cuando se presiona el botón, le envía una señal por medio de la web a la placa Arduino mega, la cual acciona su salida digital, que conectada a un módulo con relés es la encargada de accionar la electroválvula, ésta deja pasar el agua hacia la jardinera y empieza a realizarse el riego por goteo.

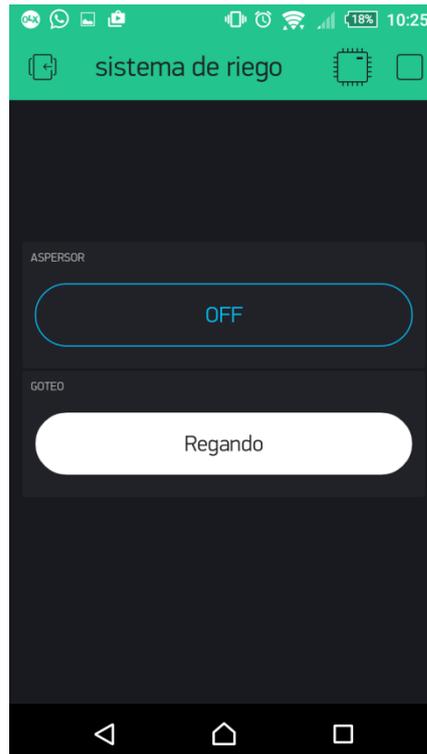
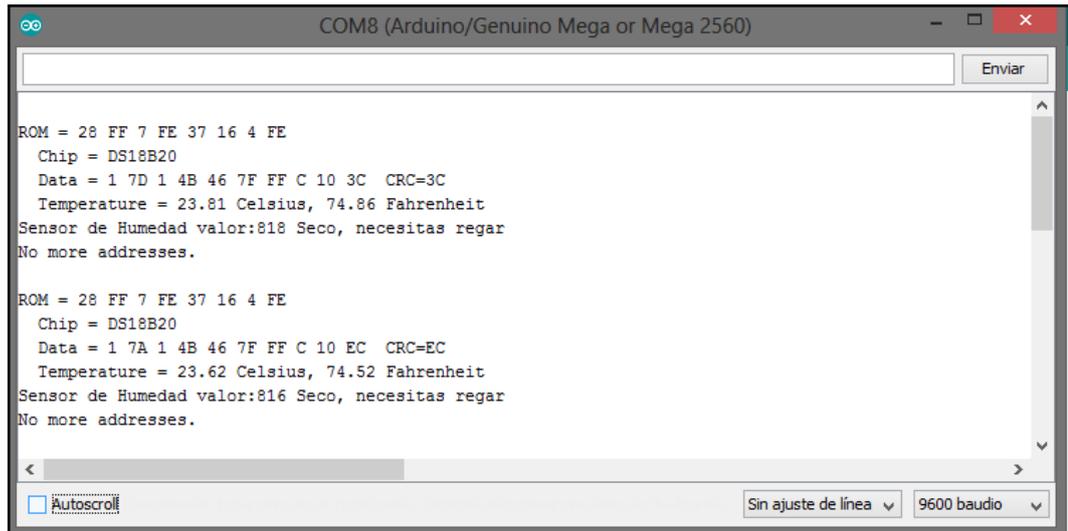


Figura 4.15: Cambio de estado Goteo a Regando

Las pruebas de accionamiento en un principio se realizaron conectados dos LEDS en un protoboard , luego de obtener resultados adecuados, se precedió a realizar la conexión a los relés y a su vez a las electrovalvulas, comprobando su correcto funcionamiento.

#### 4.2.2 Pruebas de lecturas de Sensores.

En el entorno de desarrollo Arduino IDE, existe una función llamada “Monitor Serie” la cual permite visualizar los datos que están capturando los sensores, desde las entradas tanto analógicas como digitales. Se realiza la prueba de temperatura adquirida por medio del monitor serie, en la cual el sensor de temperatura está conectado en un protoboard y a la placa de adquisición Arduino.



The screenshot shows the Serial Monitor window in Arduino IDE, titled "COM8 (Arduino/Genuino Mega or Mega 2560)". The window displays two lines of sensor data. The first line shows: ROM = 28 FF 7 FE 37 16 4 FE, Chip = DS18B20, Data = 1 7D 1 4B 46 7F FF C 10 3C CRC=3C, Temperature = 23.81 Celsius, 74.86 Fahrenheit, and Sensor de Humedad valor:818 Seco, necesitas regar. The second line shows: ROM = 28 FF 7 FE 37 16 4 FE, Chip = DS18B20, Data = 1 7A 1 4B 46 7F FF C 10 EC CRC=EC, Temperature = 23.62 Celsius, 74.52 Fahrenheit, and Sensor de Humedad valor:816 Seco, necesitas regar. The window also features an "Enviar" button, an "Autoscroll" checkbox, and dropdown menus for "Sin ajuste de línea" and "9600 baudio".

```
COM8 (Arduino/Genuino Mega or Mega 2560)
Enviar
ROM = 28 FF 7 FE 37 16 4 FE
Chip = DS18B20
Data = 1 7D 1 4B 46 7F FF C 10 3C CRC=3C
Temperature = 23.81 Celsius, 74.86 Fahrenheit
Sensor de Humedad valor:818 Seco, necesitas regar
No more addresses.

ROM = 28 FF 7 FE 37 16 4 FE
Chip = DS18B20
Data = 1 7A 1 4B 46 7F FF C 10 EC CRC=EC
Temperature = 23.62 Celsius, 74.52 Fahrenheit
Sensor de Humedad valor:816 Seco, necesitas regar
No more addresses.

Autoscroll Sin ajuste de línea 9600 baudio
```

Figura 4.16: Monitor Serie Lectura de Valores de los sensores

Como se ve en la captura de pantalla del monitor serie del entorno de desarrollo Arduino IDE Figura 4.16, se puede apreciar que en la cuarta línea la temperatura en el momento de la captura es de 23.81 grados Celsius y 74.86 grados Fahrenheit.

Además de tener el valor de humedad que es de 818, ya que el sensor de humedad nos entrega valores de 1 a 1023, donde el valor más alto significa la ausencia de humedad y a su vez muestra su estado en el cual nos indica que está seco y que se necesita realizar las labores de riego.

Una vez que se obtuvieron resultados adecuados en la adquisición y lectura de los datos en las pruebas realizadas en protoboard, se procedió a realizar la configuración para la lectura de datos en Blynk donde para la humedad tanto del patio como de la jardinera (goteo), se incluyó un gráfico de estado con valores Figura 4.17, para el sensor de temperatura se configuró un label para visualizar el valor de temperatura en ese momento y se realizó las pruebas y mediciones en el terreno obteniendo los siguientes resultados, presentados en la Figura 4.17.



Figura 4.17: Prueba de Sensores

Gracias a la gran flexibilidad y widgets que contiene la aplicación de Blynk se continúa configurando la aplicación para obtener la mejor presentación y visualización de datos, a su vez se procede incluir una pantalla LCD, en la cual se muestra el estado tanto del patio como de la jardinera, en ésta se visualiza si hay que realizar las actividades de riego o no.

Teniendo en cuenta que el sensor de humedad entrega valores de 1 a 1023, se procedió a realizar la conversión a porcentajes, para tener una mejor visualización de la humedad tanto en el patio como en la jardinera.

Para esta prueba se realizó un pequeño riego en el patio por aspersion y en la jardinera se colocó un sensor sobre tierra completamente seca para así tener una lectura con 0 % de humedad como se muestra en la Figura4.18.



Figura 4.18: Prueba de Sensores de Humedad y temperatura

Como se puede observar en la pantalla LCD el estado del patio se encuentra en “HÚMEDO” teniendo un porcentaje de 64 % de humedad, mientras que en el sensor del jardín se tiene un estado que indica “Regar”, ya que el porcentaje de humedad se encuentra en 0%. Para el sensor de temperatura se lo sostuvo por un tiempo aproximado de 1 minuto sobre la mano para así calentarlo y tener una medición diferente a la de la temperatura ambiental en ese momento.

El resultado final de cómo se ve la aplicación de Blynk para el accionamiento, lectura y visualización de datos se muestra en la Figura 4.19.

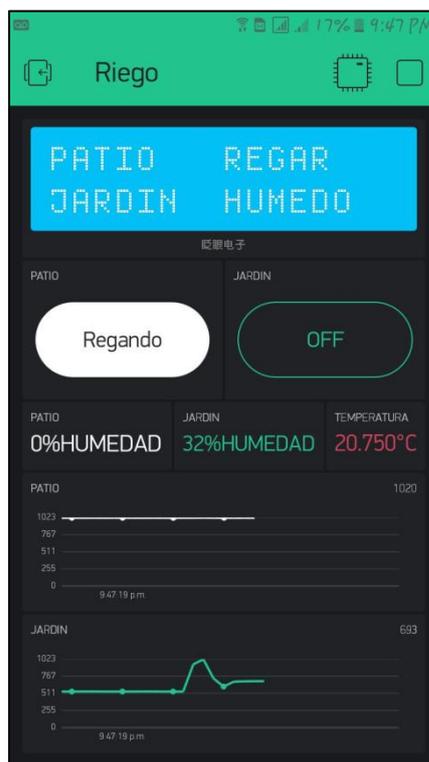


Figura 4.19: Visualización final de la aplicación

### 4.3 Operaciones del funcionamiento del sistema.

Primero: Entra en funcionamiento es el servidor que funciona en la tarjeta Raspberry pi2 que se conecta con un cable RJ45 al router D-Link DIR-610.

Segundo: Entra en funcionamiento la tarjeta de Arduino que se conectará con el router a través de un cable RJ45 y a su vez se enlaza con el servidor.

Tercero: Se enlaza la comunicación del dispositivo móvil con el router a través de señal Wi-Fi, que a su vez se podrá comunicar con el servidor y la tarjeta de Arduino.

Cuarto: Se podrá recibir datos a través de los sensores de humedad y temperatura, que emitirán una señal a la tarjeta de Arduino. Y ésta enviará datos al servidor que, a su vez, se podrá visualizar en el dispositivo móvil.

Quinto: una vez visualizado los valores en el dispositivo móvil se procede a tomar decisiones, se envía una orden que empezará desde el dispositivo móvil que se comunicará con la tarjeta Arduino, a través del servidor, una vez con las ordenes la tarjeta Arduino activa/desactiva los actuadores que en este caso son las electroválvulas.

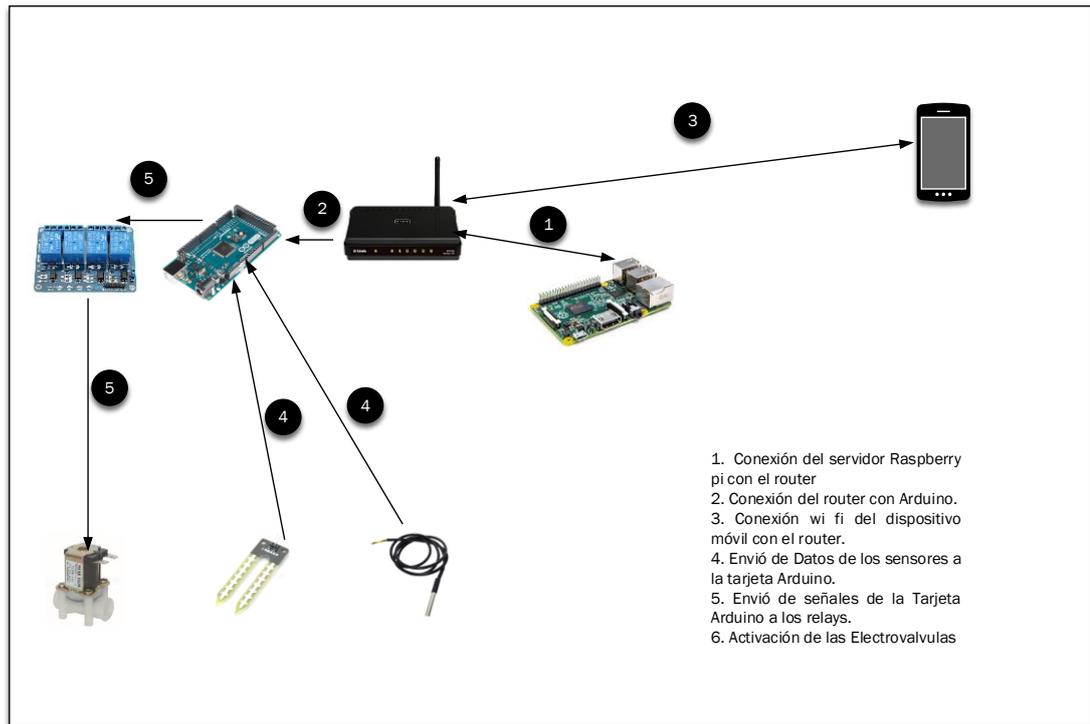


Figura 4.20: Proceso de funcionamiento

## **CONCLUSIONES Y RECOMENDACIONES.**

### **CONCLUSIONES.**

Con los resultados obtenidos luego de las diferentes pruebas del sistema de riego, se puede apreciar que el sistema es completamente funcional, ya que tanto en el accionamiento como en la lectura y visualización de los datos opera correctamente en tiempo real, con una respuesta en el accionamiento como en la lectura de los datos de alrededor de 10 a 15 segundos, cumpliendo así con el objetivo de tener un sistema viable para el control de riego mediante un dispositivo móvil.

Las pruebas del sistema se realizaron tanto con servidor local como con acceso desde fuera de la red, teniendo como resultado su óptimo funcionamiento de las 2 opciones aplicadas. Se decidió utilizar el servicio de servidor web proporcionado por Blynk, ya que este es gratuito, mientras que si se utilizaba el servidor local para acceso desde fuera de la red se necesitaba contratar una IP fija con el proveedor de internet, en este caso se debe contratar un servicio de DNS para el re-direccionamiento hacia nuestro servidor con un costo adicional.

Como conclusión final luego de evaluar el proyecto, se tiene un sistema práctico para realizar el riego dentro de una propiedad es fácil de usar, tiene una respuesta rápida en tiempo real que permite mejorar el control y manejo sobre los espacios verdes, obteniéndose una alternativa diferente para realizar las labores de riego y para el cuidado y mantenimiento de dichos espacios verdes.

### **RECOMENDACIONES.**

Como recomendaciones se puede decir que los sensores utilizados inicialmente podrían no ser lo más óptimos para este tipo de aplicación, ya que en el mercado existe sensores de humedad y temperatura de mayor precisión, pero con un costo más elevado, que daría un grado de mayor confiabilidad sobre el estado del suelo y temperatura ambiental.

De igual manera se podría mejorar las electroválvulas de accionamiento, ya que estas funcionan a 12v y se las podría sustituir por unas de mejor calidad que hay en el mercado y que funcionan a 110v, que se podría tomar directamente la alimentación de la red de suministro de energía eléctrica en la propiedad.

Dentro de la programación de la tarjeta Arduino, se recomienda tener las últimas actualizaciones, tanto del entorno de desarrollo de Arduino ID como de sus librerías, para evitar fallos en la comunicación del programa del sistema de control de riego.

## REFERENCIAS BIBLIOGRÁFICAS

- Arduino. (2016). *Arduino*. Obtenido de Arduino: [www.arduino.cc](http://www.arduino.cc)
- Armijos Torres, F. M., & Pozo Morillo, B. V. (2006). *Diseño y construcción de un cañon de riego por aspersión*. Quito.
- Avenza Álvarez, Á. (2013). *Manejo, riego y abonado del suelo (UF0012)*. Málaga: IC Editorial.
- Blumsztein, E. C. (16 de JULIO de 2013). *revista.seguridad*. Obtenido de <http://revista.seguridad.unam.mx/numero-18/sistemas-scada-consideraciones-de-seguridad>
- Blynk. (2016). *How Blynk Works*. Obtenido de Blynk: <http://docs.blynk.cc/#license>
- Carletti, E. (2016). *ROBOTS*. Obtenido de Sensores - Conceptos generales: [http://robots-argentina.com.ar/Sensores\\_general.htm](http://robots-argentina.com.ar/Sensores_general.htm)
- Castro Gil, M. A., Mur Pérez, F., & Díaz Orueta, G. (2007). *Comunicaciones industriales: principios básicos*. Madrid: UNED - Universidad Nacional de Educación a Distancia.
- Corona Ramírez, L. G., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores*. Mexico: PATRIA, S.A. DE C.V.
- Gallegos Reinoso, N. A., & Marín Buele, C. A. (2012). *Estructura del centro de control SCADA-EMS para centrales de generación hidroeléctrica*. Cuenca.
- García Lozano, F. (1967). *Riegos por aspersión y sus distintas aplicaciones*. Madrid: Dossat .
- Hidalgo. (2012). *Observatorio de Sequía*. Obtenido de Eval de eficiencia Sistema de Riego: <http://www.observatoriodga.cl/gallery/4/3.27%20%20Hidalgo,%202012.%20Eval%20de%20eficiencia%20Sistema%20de%20Riego%20TESIS.pdf>
- Jardines Modernos. (11 de 11 de 2012). *Jardines Modernos*. Obtenido de Aspersores: [http://jardinesmodernos1.blogspot.com/2012/11/aspersores\\_11.html](http://jardinesmodernos1.blogspot.com/2012/11/aspersores_11.html)
- Llamas, R. S. (2012). *Implementacion de protocolo de comunicaciones MODBUS/TCP para linux en lenguaje C++*. Cartagena: Universidad Politécnica de Cartagena.
- Losada Villasante, A. (2005). *El riego: II. fundamentos de su hidrología y su práctica*. Madrid: Mundi-Prensa.

- Maxim Integrated. (2008). *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*. San Jose.
- Molina Robles, F. J. (2014). *Redes locales*. Madrid: RA-MA Editorial.
- Moya, J. T., & Hoyos, D. (2010). RED DE SENSORES Y CONTROL INALÁMBRICA PARA UN SISTEMA DE GENERACIÓN DE VAPOR. *ASADES*, 14(0329-5184 ), 7.
- Oliva, N., Castro Gil, M. A., & Díaz Orueta, G. (2013). *Redes de comunicaciones industriales*. Madrid: UNED - Universidad Nacional de Educación a Distancia.
- Palacios Silva, F. X., & Mera Clavijo, F. J. (2012). *Diseño e implementación de un Sistema SCADA*. Sangolquí.
- Redes inalámbricas y cableadas*. (2014). Obtenido de Topologia de árbol: <https://redesinalambricasycableadas.wordpress.com/redes-cableadas/diferentes-topologias-de-red/topologia-de-arbol/>
- Riegos a pie*. (23 de Noviembre de 2012). Obtenido de Los riegos por superficie: [https://previa.uclm.es/area/ing\\_rural/Hidraulica/PresentacionesPDF\\_STR/RiegosAPie.pdf](https://previa.uclm.es/area/ing_rural/Hidraulica/PresentacionesPDF_STR/RiegosAPie.pdf)
- Rodríguez Penin, A. (2007). *Sistemas SCADA*. Barcelona: Marcombo,.
- Rodríguez Penin, A. (2008). *Comunicaciones Industriales*. Barcelona: Marcombo.
- Tarjuelo Martín-Benito, J. M. (2005). *El riego por aspersión y su tecnología (3a. ed.)*. Madrid: Mundi-Prensa.
- Tobajas García, C. (2012). *Instalaciones domóticas*. Barcelona: Ediciones Ceysa.
- Torrente Artero, Ó. (2013). *Arduino Curso Practico de Formacion*. México D.F.: Alfaomega Grupo Editor.
- Vargas Reinares, S., Jiménez Macías, E., & Pérez de la Parte, M. (2007). Sistema de control remoto de riego por goteo y por aspersión por telefonía móvil. *Escuela Técnica Superior de Ingeniería Industrial. Universidad de La Rioja*.
- Visualization and control of an irrigation control system for the Mandurah suburb in Perth, Western Australia*. (2011). Obtenido de Reliance SCADA/HMI system: <https://www.reliance-scada.com/en/success-stories/other/visualization-and-control-of-an-irrigation-control-system-for-the-mandurah-suburb-in-perth-western-australia>
- Zimmermann, J. (1975). *El riego*. Mexico: Continental.
- μConnectME*. (6 de abril de 2016). Obtenido de Sensor de humedad del suelo FC-28: <http://www.uconnectme.com/producto/sensor-de-humedad-del-suelo/>

## ANEXOS

### Anexo 1

Incluimos las librerías a utilizarse en el programa Arduino.

```
#define BLYNK_PRINT Serial  
  
#include <SPI.h>  
  
#include <Ethernet.h>  
  
#include <BlynkSimpleEthernet.h>  
  
#include <OneWire.h>  
  
#include <SimpleTimer.h>
```

Definimos las variables necesarias para la programación de los sistemas

```
int sensorhum = A0;  
  
int sensorvalor = 0;  
  
int sensorhum2 = A1;  
  
int sensorvalor2 = 0;  
  
int valvula = 5;  
  
int valvula2 = 4;  
  
long porcentaje = 0;  
  
long porcentaje1 = 0;  
  
OneWire ds(2);  
  
char auth[] = "d8072b031c72447183949126c4b25fd6";  
  
  
WidgetLCD lcd(V1);  
  
SimpleTimer timer;  
  
#define W5100_CS 10  
  
#define SDCARD_CS 4  
  
  
void setup()
```

```
{
  pinMode(SDCARD_CS, OUTPUT);
  digitalWrite(SDCARD_CS, HIGH);
  pinMode (valvula,OUTPUT);// Declaramos como salida la variable válvula
  timer.setInterval(1000L, sendUptime);
  timer.setInterval(1000L, sendUptime1);
  timer.setInterval(1000L, temperaturatime);
  Serial.begin(9600);
  Blynk.begin(auth);
  pinMode(valvula2, OUTPUT); // Declaramos como salida la variable válvula
}
//Programación para la adquisición del valor de temperatura .
void temperaturatime()
{
  byte i;
  byte present = 0;
  byte type_s;
  byte data[12];
  byte addr[8];
  float celsius, fahrenheit;
  if ( !ds.search(addr) ) {
    Serial.println("No more addresses.");
    Serial.println();
    ds.reset_search();
    delay(250);
    return;
  }
```

```
Serial.print("ROM =");
for ( i = 0; i < 8; i++) {
    Serial.write(' ');
    Serial.print(addr[i], HEX);
}
if (OneWire::crc8(addr, 7) != addr[7]) {
    Serial.println("CRC is not valid!");
    return;
}
Serial.println();
// the first ROM byte indicates which chip
switch (addr[0]) {
    case 0x10:
        Serial.println(" Chip = DS18S20"); // or old DS1820
        type_s = 1;
        break;
    case 0x28:
        Serial.println(" Chip = DS18B20");
        type_s = 0;
        break;
    case 0x22:
        Serial.println(" Chip = DS1822");
        type_s = 0;
        break;
    default:
        Serial.println("Device is not a DS18x20 family device.");
        return;
}
```

```
ds.reset();

ds.select(addr);

ds.write(0x44);

end

delay(1000);

present = ds.reset();

ds.select(addr);

ds.write(0xBE);    // Read Scratchpad

Serial.print(" Data = ");

Serial.print(present, HEX);

Serial.print(" ");

  for ( i = 0; i < 9; i++) {           // Necesitamos 9 bytes

    data[i] = ds.read();

    Serial.print(data[i], HEX);

    Serial.print(" ");

  }

Serial.print(" CRC=");

Serial.print(OneWire::crc8(data, 8), HEX);

Serial.println();

// Convertir los datos a la temperatura real

// porque el resultado es un entero de 16 bits,

// se almacena en un tipo "int16_t", que siempre tiene 16 bits

// incluso cuando se compila en un procesador de 32 bits.

int16_t raw = (data[1] << 8) | data[0];

if (type_s) {

  raw = raw << 3; // 9 bit de resolución por defecto
```

```
    if (data[7] == 0x10) {  
// "count remain" proporciona una resolución completa de 12 bits  
  
        raw = (raw & 0xFFF0) + 12 - data[6];  
  
    }  
  
    } else {  
  
        byte cfg = (data[4] & 0x60);  
  
//los bits bajos son indefinidos, así que los llevamos a cero a  
ellos  
  
        if (cfg == 0x00) raw = raw & ~7; // 9 bit resolution, 93.75 ms  
  
        else if (cfg == 0x20) raw = raw & ~3; // 10 bit res, 187.5 ms  
  
        else if (cfg == 0x40) raw = raw & ~1; // 11 bit res, 375 ms  
  
//// por defecto es la resolución de 12 bits, 750 ms de tiempo de  
conversión  
    }  
  
    celsius = (float)raw / 16.0;  
  
    fahrenheit = celsius * 1.8 + 32.0;  
  
    Serial.print(" Temperature = ");  
  
    Serial.print(celsius);  
  
    Serial.print(" Celsius, ");  
  
    Serial.print(fahrenheit);  
  
    Serial.println(" Fahrenheit");  
  
    Blynk.virtualWrite(0, celsius, "°C");  
  
    }  
  
void sendUptime()  
  
{  
  
long sensorvalor = analogRead(A0);  
  
  
    lcd.print(0, 0, "PATIO");  
  
    if ((sensorvalor > 300) and (sensorvalor <= 700))
```

```
    lcd.print(8, 0, "HUMEDO");
    if (sensorvalor > 700)
    lcd.print(8, 0, "REGAR ");
    porcentaje = 100-(sensorvalor/10.23);
    Blynk.virtualWrite(3, porcentaje, "% HUMEDAD");
  }
void sendUptime1()
{
long sensorvalor2 = analogRead(A1);
  lcd.print(8, 1, "Humedo");
  if ((sensorvalor2 > 300) and (sensorvalor2 <= 700))
  lcd.print(8, 1, "Humedo");
  if (sensorvalor2 > 700)
  lcd.print(8, 1, "REGAR ");
  porcentaje1 = 100-(sensorvalor2/10.23);
  Blynk.virtualWrite(4, porcentaje1, "% HUMEDAD");
}
void loop()
{
  Blynk.run();
  timer.run();
}
```