



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**“IMPLEMENTACIÓN DE UN PROTOTIPO DE WSN CON**  
**NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO**  
**APLICADO A LA AGRICULTURA DE PRECISIÓN PARA EL**  
**CER – ESPOCH”**

**TRABAJO DE TITULACIÓN: PROYECTO TÉCNICO**  
**Para optar al Grado Académico de:**  
**INGENIERA EN ELECTRÓNICA TELECOMUNICACIONES Y**  
**REDES**

**AUTORA: MARJORIE NATALY PAREDES VINUEZA**

**TUTOR: ING. JOSÉ ENRIQUE GUERRA SALAZAR**

Riobamba-Ecuador

2017

©2016, Marjorie Nataly Paredes Vinueza

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES**

El tribunal del trabajo de titulación certifica que: El trabajo de titulación: IMPLEMENTACIÓN DE UN PROTOTIPO DE WSN CON NODOS INTELIGENTES PARA EL SISTEMA DE RIEGO APLICADO A LA AGRICULTURA DE PRECISIÓN PARA EL CER - ESPOCH, de responsabilidad de la señorita MARJORIE NATALY PAREDES VINUEZA, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

**NOMBRE**

**FIRMA**

**FECHA**

Ing. Washington Gilberto Luna

**DECANO DE LA FACULTAD DE  
INFORMÁTICA Y ELECTRÓNICA**

.....

Ing. Franklin Moreno Montenegro

**DIRECTOR DE LA ESCUELA DE  
INGENIERÍA ELECTRÓNICA EN  
TELECOMUNICACIONES Y REDES**

.....

Ing. José Guerra Salazar

**DIRECTOR DEL TRABAJO DE  
TITULACIÓN**

.....

Ing. Wilson Zúñiga Vinueza

**MIEMBRO DEL TRIBUNAL**

.....

Yo, Marjorie Nataly Paredes Vinueza declaro ser responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Marjorie Nataly Paredes Vinueza

## DEDICATORIA

A Dios por ser mi refugio en dificultad, por su infinito amor para conmigo y sus bendiciones reflejadas en mi vida.

A mi madre Piedad Vinueza Villacis por haberme brindado su apoyo incondicional en este recorrido, por sus consejos, por su amor, por su constante motivación y por siempre creer en mí.

A mi padre Bolívar Paredes Fuentes por su insuperable esfuerzo, por nunca rendirse a pesar de las duras pruebas, por siempre pensar en mi futuro y apoyarme en las decisiones y sueños que tengo.

A mi hermano Rubén Paredes Vinueza por su ejemplo de dedicación y esfuerzo, por su apoyo incondicional y por enseñarme que no hay imposibles en la vida.

A mi hermano Danilo Segura Vinueza y cuñada Nury Castillo Paredes por su confianza y por sus consejos de motivación que me han hecho crecer como persona.

A mi pequeña sobrinita Daniela Saraí Segura Castillo por ser mi mayor fuente de inspiración y superación, por alegrarme la vida con sus sonrisas y ocurrencias.

A mi mejor amiga Ximena Lema por su fidelidad, su amistad incondicional, por su paciencia, por ser mi compañera en este recorrido y darme su mano cuando más lo necesité.

A mis abuelos, tíos y primos por sus grandes consejos, por su confianza y cariño.

**Marjorie**

## **AGRADECIMIENTO**

A Dios, por su infinito amor y bondad.

A mis padres Piedad y Bolívar por su apoyo, amor y confianza, por todo el gran esfuerzo que hacen día a día para verme cumplir mis sueños y anhelos. A mis hermanos y familiares por sus consejos que hicieron de mí una mejor persona.

A mis amigas Ximena, Marilú y Lorena por su cariño incondicional, su apoyo su compañía a lo largo de la carrera. A Daniel Vallejo y Jefferson Vallejo por su gran aporte en este trabajo de titulación y por su constante motivación y apoyo.

A mi querida Escuela Superior Politécnica de Chimborazo y Escuela de Ingeniería en Electrónica, Telecomunicaciones y Redes por ser mi segundo hogar, por abrirme sus puertas y ser mi templo del saber.

A mis profesores y área administrativa que con su paciencia y desempeño me formaron académicamente compartiéndome sus conocimientos, en especial a mi tutor al Ingeniero José Guerra.

## TABLA DE CONTENIDO

<b>RESUMEN</b> .....	<b>xiv</b>
<b>SUMMARY</b> .....	<b>xv</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I</b>	
<b>1</b> <b>MARCO REFERENCIAL</b> .....	<b>6</b>
<b>1.1</b> <b>Tipos de Cultivos</b> .....	<b>6</b>
<b>1.2</b> <b>Requerimientos de los Cultivos</b> .....	<b>7</b>
<i>1.2.1</i> <i>Textura del suelo</i> .....	<i>7</i>
<i>1.2.2</i> <i>Profundidad de la raíz</i> .....	<i>7</i>
<i>1.2.3</i> <i>Humedad en el Suelo</i> .....	<i>8</i>
<b>1.3</b> <b>Sistema de Riego</b> .....	<b>10</b>
<i>1.3.1</i> <i>Riego por Aspersión</i> .....	<i>10</i>
<i>1.3.2</i> <i>Riego por Surcos</i> .....	<i>11</i>
<i>1.3.3</i> <i>Riego por Goteo</i> .....	<i>12</i>
<i>1.3.4</i> <i>Tabla comparativa de tipos de riego</i> .....	<i>13</i>
<b>1.4</b> <b>Sistemas existentes para riego</b> .....	<b>14</b>
<i>1.4.1</i> <i>Sistemas comerciales de riego automático</i> .....	<i>14</i>
<b>1.5</b> <b>Análisis de riego en el país</b> .....	<b>15</b>
<i>1.5.1</i> <i>Problemática</i> .....	<i>16</i>
<i>1.5.2</i> <i>Trabajos realizados en Riobamba sobre sistemas de riego</i> .....	<i>16</i>
<b>1.6</b> <b>Redes inalámbricas</b> .....	<b>17</b>
<b>1.7</b> <b>Red de Sensores Inalámbricos</b> .....	<b>17</b>
<i>1.7.1</i> <i>Elementos de una red de sensores inalámbrica (WSN)</i> .....	<i>17</i>
<i>1.7.2</i> <i>Tecnologías</i> .....	<i>18</i>
<i>1.7.3</i> <i>Topologías de WSN</i> .....	<i>19</i>
<i>1.7.4</i> <i>Aplicaciones de WSN</i> .....	<i>21</i>

1.7.5	<i>Tarjetas de desarrollo aplicadas a redes de sensores inalámbricos</i> .....	22
-------	--	----

**CAPÍTULO II**

<b>2</b>	<b>DISEÑO DEL PROTOTIPO DEL SISTEMA INALÁMBRICO DE CONTROL AUTOMÁTICO DE RIEGO</b> .....	<b>26</b>
<b>2.1</b>	<b>Requerimientos del SICAR</b> .....	<b>26</b>
<b>2.1.1</b>	<i>Concepción de la arquitectura general del sistema</i> .....	<b>27</b>
<b>2.1.2</b>	<i>Diseño de la Arquitectura del SICAR</i> .....	<b>28</b>
<b>2.2</b>	<b>Selección de elementos que conforman el SICAR</b> .....	<b>31</b>
<b>2.2.1</b>	<i>Tarjetas de desarrollo</i> .....	<b>31</b>
<b>2.2.2</b>	<i>Módulos XBee</i> .....	<b>32</b>
<b>2.2.3</b>	<i>Módulo ESP8266 Wi-Fi</i> .....	<b>34</b>
<b>2.2.4</b>	<i>Módulo SIM900</i> .....	<b>35</b>
<b>2.2.5</b>	<i>Sensor de Humedad FC-28</i> .....	<b>36</b>
<b>2.2.6</b>	<i>Sensor de humedad DHT11</i> .....	<b>37</b>
<b>2.2.7</b>	<i>Sensor de flujo de agua</i> .....	<b>38</b>
<b>2.2.8</b>	<i>Módulo relé</i> .....	<b>39</b>
<b>2.2.9</b>	<i>Panel solar</i> .....	<b>39</b>
<b>2.2.10</b>	<i>Regulador de voltaje solar</i> .....	<b>40</b>
<b>2.2.11</b>	<i>Batería Seca</i> .....	<b>41</b>
<b>2.2.12</b>	<i>Módulo micro SD</i> .....	<b>42</b>
<b>2.2.13</b>	<i>Equipo de Visualización</i> .....	<b>43</b>
<b>2.2.14</b>	<i>Electroválvula Rain Bird de 1''</i> .....	<b>43</b>
<b>2.2.15</b>	<i>Electroválvula Solenoide de ½''</i> .....	<b>44</b>
<b>2.2.16</b>	<i>Bluetooth HC-06</i> .....	<b>45</b>
<b>2.3</b>	<b>Software para el estudio y análisis de topología de red</b> .....	<b>46</b>
<b>2.3.1</b>	<i>Riverbed Modeler Academic</i> .....	<b>46</b>
<b>2.4</b>	<b>Esquema de conexión del Sistema Inalámbrico</b> .....	<b>47</b>
<b>2.4.1</b>	<i>Diagrama de conexión del nodo sensor</i> .....	<b>47</b>
<b>2.4.2</b>	<i>Diagrama de conexión del nodo actuador</i> .....	<b>48</b>



2.4.3	<i>Diagrama de conexión del nodo control</i> .....	49
2.5	<b>Requerimiento Software de la Red Inalámbrica de Sensores de Control Automático de Riego</b> .....	50
2.5.1	<i>Requerimientos software para el Nodo Sensor</i> .....	50
2.5.2	<i>Requerimientos software para el Nodo Actuador</i> .....	51
2.5.3	<i>Requerimientos software para el Nodo Control</i> .....	51
2.5.4	<i>Software de desarrollo del sistema</i> .....	52
2.5.5	<i>Programación de los nodos del sistema inalámbrico.</i> .....	53
2.5.6	<i>Configuración de los módulos XBee</i> .....	63
2.5.7	<i>Software para la presentación y visualización de información del SICAR</i> .....	65
<b>CAPÍTULO III</b>		
3	<b>ANÁLISIS Y RESULTADOS</b> .....	70
3.1	<b>Resultados de simulación en Riverbed Modeler</b> .....	70
3.1.1	<i>Análisis de la topología de red de árbol</i> .....	70
3.1.2	<i>Análisis de la topología de red de malla</i> .....	71
3.1.3	<i>Comparación de topologías</i> .....	72
3.2	<b>Pruebas del nodo sensor</b> .....	73
3.2.1	<i>Pruebas de mediciones realizadas por el nodo sensor</i> .....	73
3.2.2	<i>Repetitividad del Nodo Sensor</i> .....	75
3.2.3	<i>Características de las comunicaciones entre dispositivos del Nodo Sensor</i> .....	76
3.3	<b>Pruebas del nodo actuador</b> .....	79
3.4	<b>Pruebas del Nodo Control</b> .....	82
3.4.1	<i>Pruebas de almacenamiento</i> .....	82
3.5	<b>Funcionamiento general SICAR desarrollado para el CER – ESPOCH</b> .....	84
3.6	<b>Análisis económico de la implementación del SICAR</b> .....	85
	<b>CONCLUSIONES</b> .....	88
	<b>RECOMENDACIONES</b> .....	90
<b>BIBLIOGRAFÍA</b>		
<b>ANEXOS</b>		

## ÍNDICE DE TABLAS

<b>Tabla 1-1</b> Textura del suelo.....	7
<b>Tabla 2-1</b> Propiedades físicas de los suelos según la textura .....	10
<b>Tabla 3-1</b> Tabla comparativa de tipos de riego .....	13
<b>Tabla 4-1</b> Comparación entre tecnologías inalámbricas Wi-fi, Bluetooth y zigBee .....	19
<b>Tabla 5-1</b> Comparativa de tarjetas de desarrollo.....	25
<b>Tabla 1-2</b> Parámetros de configuración XBee .....	34
<b>Tabla 1-3</b> Mediciones realizadas con el sensor FC-28 del SICAR y tensiómetro y sensor DHT-11 con la central meteorológica del CER.....	75
<b>Tabla 2-3</b> Repetitividad de los valores leídos por los sensores del Nodo Sensor .....	76
<b>Tabla 3-3</b> Consumo de agua en la parcela y por cada planta. ....	81
<b>Tabla 4-3</b> Presupuesto detallado de cada elemento que conforman los nodos del SICAR.....	86

## ÍNDICE DE FIGURAS

<b>Figura 1-1</b> Tipos de Cultivos .....	6
<b>Figura 2-1</b> Profundidad de la raíz .....	8
<b>Figura 3-1</b> Estados de humedad del suelo.....	9
<b>Figura 4-1</b> Humedad aprovechable .....	9
<b>Figura 5-1</b> Riego por aspersión.....	11
<b>Figura 6-1</b> Riego por surcos.....	12
<b>Figura 7-1</b> Riego por goteo .....	13
<b>Figura 8-1</b> Sistemas comerciales de riego automático.....	15
<b>Figura 9-1</b> Elementos de una WSN.....	18
<b>Figura 10-1</b> Topologías ZigBee: par, estrella, malla y árbol .....	20
<b>Figura 11-1</b> Aplicaciones en la agricultura .....	21
<b>Figura 12-1</b> Tarjetas disponibles de Arduino.....	23
<b>Figura 13-1</b> Modelos de Raspberry Pi .....	24
<b>Figura 14-1</b> Tarjeta Waspote .....	24
<b>Figura 15-1</b> Tarjeta Intel Galileo .....	25
<b>Figura 1-2</b> Arquitectura general del sistema inalámbrico .....	27
<b>Figura 2-2</b> Diagrama de bloques del nodo sensor de SICAR .....	29
<b>Figura 3-2</b> Diagrama de bloques del nodo actuador de SICAR.....	29
<b>Figura 4-2</b> Diagrama de bloques del nodo control de SICAR .....	30
<b>Figura 5-2</b> Arduino UNO.....	31
<b>Figura 6-2</b> Arduino NANO .....	32
<b>Figura 7-2</b> Módulos XBee regular y módulo XBee-PRO .....	33
<b>Figura 8-2</b> XBee y sus elementos principales.....	33
<b>Figura 9-2</b> Módulo Wi-Fi ESP8266.....	35
<b>Figura 10-2</b> SIM900 GSM/GPRS .....	36
<b>Figura 11-2</b> Sensor de humedad FC-28 .....	37
<b>Figura 12-2</b> Sensor de humedad DHT11 .....	37
<b>Figura 13-2</b> Sensor de flujo de agua .....	38
<b>Figura 14-2</b> Módulo relé .....	39
<b>Figura 15-2</b> Panel solar .....	40
<b>Figura 16-2</b> Regulador de voltaje solar.....	41
<b>Figura 17-2</b> Batería Seca.....	42

<b>Figura 18-2</b> Módulo micro SD.....	42
<b>Figura 19-2</b> Pantalla de visualización.....	43
<b>Figura 20-2</b> Electroválvula de 1’’.....	44
<b>Figura 21-1</b> Electroválvula de ½ ’’.....	454
<b>Figura 22-2</b> Bluetooth HC-06.....	465
<b>Figura 23-2</b> Entorno de desarrollo de Riverbed Modeler Academic.....	46
<b>Figura 24-2</b> Esquema de conexión del nodo sensor.....	48
<b>Figura 25-2</b> Esquema de conexión del nodo actuador.....	49
<b>Figura 26-2</b> Esquema de conexión del nodo control.....	50
<b>Figura 27-2</b> Diagrama de flujo del programa para el Nodo Sensor.....	556
<b>Figura 28-2</b> Diagrama de flujo del programa para el Nodo Actuador.....	58
<b>Figura 29-2</b> Diagrama de flujo del programa para el Nodo Control.....	61
<b>Figura 30-2</b> Datos almacenados en un archivo.....	63
<b>Figura 31-2</b> Configuración del módulo control.....	64
<b>Figura 32-2</b> Configuración del módulo actuador.....	64
<b>Figura 33-2</b> Configuración del módulo sensor.....	65
<b>Figura 34-2</b> Canales creados para el SICAR.....	66
<b>Figura 35-2</b> Ventanas de visualización de datos en ThingSpeak.....	66
<b>Figura 36-2</b> Plataforma privada diseñada en HTLM.....	67
<b>Figura 37-2</b> Interfaz de la aplicación en App inventor.....	68
<b>Figura 38-2</b> Código de aplicación desarrollada en App Inventor.....	69
<b>Figura 1-3</b> Simulación de la topología de red de árbol en Riverbed Modeler.....	71
<b>Figura 2-3</b> Simulación de la topología de red de malla en Riverbed Modeler.....	72
<b>Figura 3-3</b> Retraso promedio de datos (izquierda) y carga promedio de datos (derecha).....	72
<b>Figura 4-3</b> Prototipo del Nodo Sensor (izquierda), alimentación solar del nodo (derecha).....	73
<b>Figura 5-3</b> Tensiómetro (izquierda), nodo sensor (derecha).....	74
<b>Figura 6-3</b> Estación meteorológica CER.....	74
<b>Figura 7-3</b> Ingreso del identificador del nodo, combinación del micro switch para 1 (izquierda), y combinación del micro switch para 32 (derecha).....	77
<b>Figura 8-3</b> Envío de notificaciones a través de la red GSM (izquierda) y recepción de mensaje de texto en el móvil (derecha).....	78
<b>Figura 9-3</b> Visualización de datos de temperatura (izquierda) y humedad (derecha) en la plataforma ThingSpeak.....	78
<b>Figura 10-3</b> Datos de temperatura (izquierda) y humedad (derecha) en Microsoft Excel 2013. <b>Realizado por:</b> PAREDES, Marjorie, 2017.....	79
<b>Figura 11-3</b> Prototipo del Nodo Actuador, vista externa (izquierda) y vista interna (derecha).....	79

<b>Figura 12-3</b> Visualización del caudal en la plataforma ThingSpeak (izquierda), almacenamiento de la plataforma en Microsoft Excel 2013 (derecha) .....	80
<b>Figura 13-3</b> Fotografía de la parcela de mora .....	81
<b>Figura 14-3</b> Prototipo del Nodo Control, vista externa (izquierda) y vista interna (derecha)..	82
<b>Figura 15-3</b> Prueba de apertura del .....	82
<b>Figura 16-3</b> Monitoreo en la aplicación móvil (izquierda), y visualización en el nodo sensor (derecha).....	83
<b>Figura 17-3</b> Comunicación nodo control-sensor (izquierda) y comunicación nodo control-actuador (derecha).....	84
<b>Figura 18-3</b> SICAR en funcionamiento en la parcela de mora. ....	85

## ÍNDICE DE ABREVIATURAS

<b>FOA</b>	Food and Agriculture Organization of the United Nations (Organización de las Naciones Unidas para la Alimentación y Agricultura)
<b>UPA</b>	Unión de Pequeños Agricultores y Ganaderos
<b>CER</b>	Centro Experimental del Riego
<b>ESPOCH</b>	Escuela Superior Politécnica de Chimborazo
<b>INEC</b>	Instituto Nacional de Estadísticas y Censos
<b>RF</b>	Radio Frecuencia
<b>CC</b>	Capacidad de Campo
<b>PMP</b>	Punto de Marchitez Permanente
<b>HA</b>	Humedad aprovechable
<b>Ha</b>	Hectáreas
<b>WSN</b>	Wireless Sensor Network (Red de Sensores Inalámbricos)
<b>TCP</b>	Transmission Control Protocol (Protocolo de Control de Transmisión)
<b>IP</b>	Internet Protocol (Protocolo de Internet)
<b>IEEE</b>	Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica)
<b>IoT</b>	Internet of Things (Internet de las cosas)
<b>IDE</b>	Integrated Development Environment (Entorno de Desarrollo Integrado)
<b>SICAR</b>	Sistema Inalámbrico de Control Automático de Riego
<b>LCD</b>	Liquid Cristal Display (Representación visual por cristal líquido)
<b>GSM</b>	Global System for Mobile Communication (Sistema Global para las comunicaciones Móviles)
<b>GPRS</b>	General Packet Radio Service (El servicio general de paquetes vía radio)
<b>RTC</b>	Red telefónica conmutada

## RESUMEN

El objetivo del trabajo de titulación fue el diseño e implementación de una red inalámbrica de sensores inteligentes para la supervisión y control de riego de la parcela de mora en el Centro Experimental del Riego de la Escuela Superior Politécnica de Chimborazo el cual consiste en un sistema electrónico de sensores inalámbricos basados en tarjetas de desarrollo Arduino. El sistema fue estructurado en una topología de red tipo malla; que permite medir la humedad del suelo y la temperatura ambiente teniendo la capacidad de tomar decisiones de cuándo y cuánto regar, se diseñó e implementó tres nodos: sensor, actuador y control. Para el sistema se desarrolló una aplicación móvil que mantiene una comunicación vía Bluetooth para la visualización de las lecturas del nodo sensor; los datos son receptados y visualizados en tiempo real, además la información obtenida es mostrada de manera gráfica y almacenada en la plataforma libre *ThinkSpeak*, asimismo la información es almacenada en una memoria micro SD en el nodo control. La Facultad de Recursos Naturales fue el escenario para la medición y toma de resultados del sistema implementado, tomando muestras por ocho días consecutivos. Se comprobó que el sensor de humedad presenta un error de  $\pm 2$  centibares y el de temperatura de  $\pm 1\%$  demostrando que el sistema no incorpora error a los sensores y que es confiable. Los resultados mostraron que el sistema inalámbrico implementado mantiene una comunicación sin pérdida a distancias de 300 metros en línea de vista directa con una potencia de recepción de -58 dBm. Por las características detalladas este sistema representa ser una herramienta de ayuda para la recolección de información para investigadores aplicable en diversos cultivos. Se recomienda realizar estudios para incrementar la capacidad de recolectar información de luz solar, velocidad del viento, pluviometría entre otros.

**Palabras Clave:** <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TELECOMUNICACIONES>, <COMUNICACIONES INALÁMBRICAS> <RED DE SENSORES INALÁMBRICOS (WSN)>, <SENSOR DE HUMEDAD>, <SENSOR DE TEMPERATURA>, <CONTROL DE RIEGO>, <ARDUINO (SOFTWARE-HARDWARE)>

## SUMMARY

The objective of this degree work was the design and implementation of a wireless network of intelligent sensors for the supervision and control of watering of the blueberry plots in the experimental center of watering of Polytechnic School of Chimborazo that consists in an electronic system of wireless sensors based on cards of Arduino development. The system was structured in a net topology like a grid; that allows to measure ground humidity and room temperature being able to take decisions about when and how to water, three nodes were designed and implemented: sensor, actuator and control. For the system was developed a movie application that maintains a Bluetooth communication for the visualization of node sensor readings; the data were receipted and visualized in real time, besides the obtained information is shown in graphic way and stored in the free platform ThinkSpeak, in the same way the information is stored in a SD micro memory in the control node. The Natural Recourses Faculty was the scenery for the measuring and result taking of the implemented system, taking samples for eight followed days. It was proved that the humidity sensor presents an error of  $\pm 2$  centibars and the temperature one of  $\pm 1\%$  demonstrating that the system doesn't incorporate error to the sensors and that it is reliable. The results showed that the implemented wireless system maintains a communication without losses in distances of 300 meters in line of straight sight with a potency of reception of -58 dBm. For the detailed characteristics this system represents a helpful tool for the information collection for researchers applicable in several crops. It is recommended to develop studies for increasing the capacity to collect information of solar light, wind speed, rainfall among others.

**Keywords:** <TECHNOLOGY AND SCIENCES OF ENGINEERING>, <TELECOMMUNICATIONS>, <WIRELESS COMMUNICATION>, <WIRELESS SENSORS NET (WSN)>, <HUMIDITY SENSOR>, <TEMPERATURE SENSOR>, <WATERING CONTROL>, <ARDUINO (SOFTWARE-HARDWARE)>.



## **INTRODUCCIÓN**

### **ANTECEDENTES**

En el mundo, el 70 % agua disponible se destina a la agricultura, el 20 % a la industria y el 10 % a uso doméstico (El Telégrafo, 2014). FAO (*Food and Agriculture Organization of the United Nations por sus siglas en inglés*) recalca que América Latina desperdicia un 25 por ciento del agua destinada a los sistemas de riego.

China e India son países pioneros en el uso de sistemas de riego por inundación y aspersión, dichos sistemas tienen precios elevados que van desde los \$2.200 en adelante para el uso en jardines y para sembríos de superficie mayor los precios llegan a triplicarse, “Tornado – Sistemas de Riego” empresa líder en ventas sistemas de riego en México oferta servicios agricultura de precisión que oscilan entre \$2.500 y \$15.000 y no incluyen instalación del sistema.

En Ecuador la mayor parte de la demanda de agua se concentra en el sector agrícola, según Dennis García, subsecretario de Riego y Drenaje del Ministerio de Agricultura y Ganadería (Magap) “el problema en Ecuador es que hay un acaparamiento de agua muy grande”. De acuerdo con datos de la oficina de riego, el 97% de las unidades productivas agropecuarias (UPA) del país tiene una superficie menor a 50 hectáreas y utiliza el 49% del agua disponible para riego (El Telégrafo, 2014).

Un estudio desarrollado en la ciudad de Quevedo, entre octubre y diciembre del 2014 por la Universidad Técnica Estatal de Quevedo, toma mediciones en un terreno de 75 metros cuadrados y hace una comparación de la cantidad de agua utilizada en un sistema de riego por aspersión e inundación, teniendo como resultado que se empleó 455,86 metros cúbicos de agua con el primer sistema y 1158,3 metros cúbicos de agua con el segundo. (CEDEÑO, 2014 pág. 141)

El Plan Nacional del Buen Vivir en el objetivo 7.6 menciona: “Gestionar de manera sustentable y participativa el patrimonio hídrico, con enfoque de cuencas y caudales ecológicos para asegurar el derecho humano al agua”

Por tal motivo, el presente trabajo de titulación busca estructurar un sistema electrónico de bajo costo que por medio de una red de sensores inalámbricos; almacene, supervise y suministre datos en tiempo real de porcentajes de humedad actual del suelo, necesarios para el sembrío de mora. En busca de proveer la información necesaria de cuándo y cuánto recurso hídrico necesita la plantación en una zona específica, logrando la optimización de tan valioso recurso.

## **FORMULACIÓN DEL PROBLEMA**

- ¿Es necesario la implementación de un prototipo de WSN con nodos inteligentes para el sistema de riego aplicado a la agricultura de precisión para el CER - ESPOCH?

## **SISTEMATIZACIÓN DEL PROBLEMA**

- ¿Cuál es el sistema de riego que optimiza de mejor manera el recurso hídrico y cumple con las exigencias de los diferentes cultivos que maneja el Centro Experimental de Riego - CER en la ESPOCH?
- ¿Cuál es el diseño que cumple con los requerimientos del sistema electrónico a implementar?
- ¿Cómo cumple el sistema electrónico a implementar los requerimientos del Centro Experimental de Riego - CER?
- ¿Qué comunicación inalámbrica mejor se adapta a las necesidades del sistema electrónico?
- ¿Cuál es la topología de red adecuada para las necesidades del sistema electrónico?

## **JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN**

### **JUSTIFICACIÓN TEÓRICA**

La necesidad de tener un control, monitoreo y a su vez un aviso de la cantidad del recurso hídrico necesario para las plantaciones de mora en el Centro Experimental de Riego – CER ESPOCH es evidente dado que utilizan el método Gravimétrico para la obtención de porcentajes de humedad y de esa manera realizan la distribución del recurso hídrico en la plantación.

Según los datos registrados en el INEC (Instituto Nacional de Estadísticas y Censos), en Ecuador en 2015 se registraron 1945 sistemas de riego con una superficie de 358.981,47 hectáreas, el

74.44 % corresponde a superficie efectivamente regada, dejando una diferencia de 25.56 % de brecha. Estos datos revelan que los sistemas de riego existentes muestran un porcentaje considerable de desperdicio del recurso hídrico.

Entre las investigaciones realizadas en el país, se encuentra una elaborada en la Escuela Superior Politécnica del Litoral, que trata de la implementación de un sistema de riego inteligente que se enfoca en el ahorro de agua y energía utilizando módulos Radio Frecuencia (RF), los datos recolectados son enviados a un servidor de base de datos donde son almacenados temporalmente. Los resultados muestran el alto costo de implementación del sistema, además aún existe desperdicio del recurso hídrico ya que solo se divide en dos zonas de riego.

La propuesta de proyecto busca desarrollar un sistema electrónico que supervise, controle, almacene e informe en tiempo real los porcentajes de humedad del suelo de la plantación de mora para una distribución específica del recurso hídrico. La información se almacena en una base de datos la cual representa una herramienta de ayuda para estudios futuros para poder lograr el máximo aprovechamiento y ahorro del recurso. El proyecto se centra en la implementación de un prototipo para el monitoreo continuo del porcentaje de humedad del suelo en la plantación de mora en el Centro Experimental de Riego en los predios de la Escuela Superior Politécnica de Chimborazo y que por sus características de diseño puede ser aplicado en cualquier tipo de cultivo y territorio.

## **JUSTIFICACIÓN APLICATIVA**

Un estudio realizado por el Centro Experimental de Riego revela que el método Gravimétrico utilizado para la obtención del porcentaje de humedad del suelo es un método preciso, pero tiene desventajas una de las cuales es que se obtienen resultados aproximadamente a las 24 horas, además la muestra tomada del campo es destruida por el calor a la que es sometida para dichas pruebas. Asimismo el proceso de riego es llevado a cabo dependiendo de los resultados de las muestras enviadas al laboratorio hace 24 horas, indudablemente los resultados no muestran las necesidades actuales de la planta.

Considerando la necesidad de que existan dispositivos que muestre los niveles de humedad en un tiempo menor, se debe implementar un sistema electrónico que muestre los niveles de humedad del suelo en tiempo real, controle la distribución del recurso hídrico en la zona necesaria ahorrando el tan valioso recurso, además que almacene la información para estudios

futuros. Este sistema puede hacer uso de una Red de Sensores Inteligentes con comunicación inalámbrica, que reduce costos y facilita la instalación. Esta red será diseñada para crecimientos futuros sin alterar la red inicial, además los nodos son capaces de obtener información de su entorno, procesarla localmente y comunicarla a largas distancias.

El trabajo con Redes de Sensores Inalámbricos requiere de un control por medio de un dispositivo para determinar la forma de funcionamiento y las acciones a realizar, todo esto se puede lograr con el uso de tarjetas de desarrollo que tengan compatibilidad con dispositivos dentro de una red de sensores. Además de un nodo transmisor que visualiza y transfiere la información al control para el respectivo funcionamiento. La integración de estas tecnologías hacen viable desarrollar un sistema de monitoreo en tiempo real de los niveles de humedad del suelo para determinar cuándo y cuánto recurso hídrico necesita la plantación de mora en una zona determinada, además que la información sea almacenada y esté a disposición del administrador, el sistema electrónico será elaborado con mano de obra nacional y El Plan Nacional del Buen Vivir en el objetivo 10.1 menciona: “Diversificar y generar mayor valor agregado en la producción nacional”.

## **OBETIVOS**

### **OBJETIVO GENERAL**

Implementar un prototipo de WSN con nodos inteligentes para el sistema de riego aplicado a la agricultura de precisión para el CER - ESPOCH.

### **OBJETIVOS ESPECÍFICOS**

- Investigar el sistema de riego que optimice de mejor manera el recurso hídrico y cumpla con las exigencias de los diferentes cultivos que maneja el Centro Experimental de Riego - CER en la ESPOCH.
- Estudiar la comunicación inalámbrica que mejor se adapte a los requerimientos del sistema electrónico.

- Seleccionar la topología de red adecuada que satisfaga las necesidades del sistema electrónico.
- Diseñar un sistema electrónico que cumpla con los requerimientos del Centro Experimental de Riego – CER.

El epítome descriptivo del trabajo de titulación consta de introducción, tres capítulos, conclusiones y recomendaciones. El primer capítulo denominado marco referencial abarca temas relacionados con los tipos de cultivos, investigaciones realizadas en la ciudad de Riobamba, problemática de riego, red de sensores inalámbricos y tarjetas de desarrollo. En el segundo capítulo aborda los requerimientos *hardware* y *software* del sistema, diseño de cada nodo: sensor, actuador y control, así como también el almacenamiento de la información. El tercer capítulo se dedica a la validación del sistema y análisis de resultados, se incluye un análisis de costos. En las conclusiones se agrupa un resumen de resultados alcanzados dentro de la investigación y para culminar en las recomendaciones se plantea futuros objetivos que podrían implementarse en el futuro para la mejora del sistema actual.

# CAPÍTULO I

## 1 MARCO REFERENCIAL

En este capítulo se investiga los diferentes tipos de cultivos y requerimientos, además los sistemas de riego existentes en el país; así como también la red de sensores inalámbricos, sus características y tecnologías existentes para dicha comunicación.

### 1.1 Tipos de Cultivos

Los cultivos se clasifican según su ciclo, requerimientos de clima y finalidad como se muestra en la *Figura 1-1*, dentro del ciclo puede ser: anual, de media estación; por los requerimientos de clima puede ser: clima cálido, criófilos, frío y por su finalidad serán: forrajeras, frutales entre otros.



**Figura 1-1** Tipos de Cultivos  
Realizado por: PAREDES, Marjorie, 2017

## 1.2 Requerimientos de los Cultivos

Los rangos de medida de los requerimientos puede variar dependiendo de la región y del cultivo, los más importantes son textura del suelo, profundidad y humedad.

### 1.2.1 Textura del suelo

Es la cantidad de arcilla, arena y limo que posee el suelo, la textura es una característica que no puede ser cambiada ya que esta depende de la región y otros factores; existen tres tipos de suelo según su textura como son suelo arenoso, suelo arcilloso y suelo franco. Las características de las diferentes texturas se muestran en la *Tabla 1-1*.

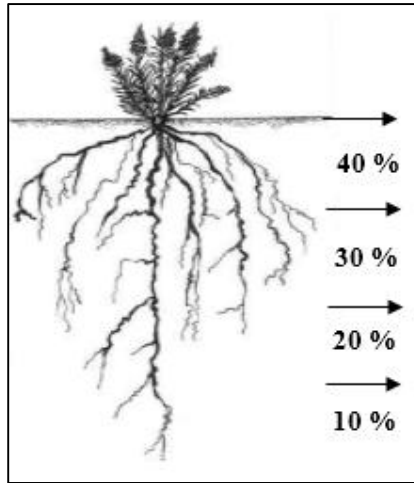
**Tabla 1-1** Textura del suelo

<b>ARENOSO</b>	<b>ARCILLOSO</b>	<b>FRANCO</b>
Áspero al tacto	Seco, se puede formar terrones duros, que no pueden romperse fácilmente	Seco, parece en terrenos suaves
Puede romperse fácilmente	Suave al tacto	Húmedo, se amasa y es muy pegajoso
No es pegajoso al tacto	Húmedo pegajoso	Suave al tacto, pegajoso
No mancha los dedos	Mancha los dedos	Mancha los dedos

Fuente: (Navarro, 2014 pág. 28)

### 1.2.2 Profundidad de la raíz

A mayor profundidad del suelo la planta es más estable además las raíces podrán extenderse formando un triángulo. En la *Figura 2-1* se observa que raíz está dividida en cuatro zonas, la primera zona (zona efectiva) que corresponde al 40 % es donde se encuentran la mayor parte de las raíces, si se riega tres cuartas partes del total de la raíz estamos regando un equivalente al 90%, con esto se asegura un crecimiento y desarrollo eficaz de la planta. (Navarro, 2014)



**Figura 2-1** Profundidad de la raíz  
 Realizado por: PAREDES, Marjorie, 2017

### 1.2.3 *Humedad en el Suelo*

Es muy cambiante y depende de algunas variables como las plantas, la profundidad del suelo y el clima; además depende de la ubicación del terreno. (Forsythe, 1985)

Se define tres estados de humedad que son:

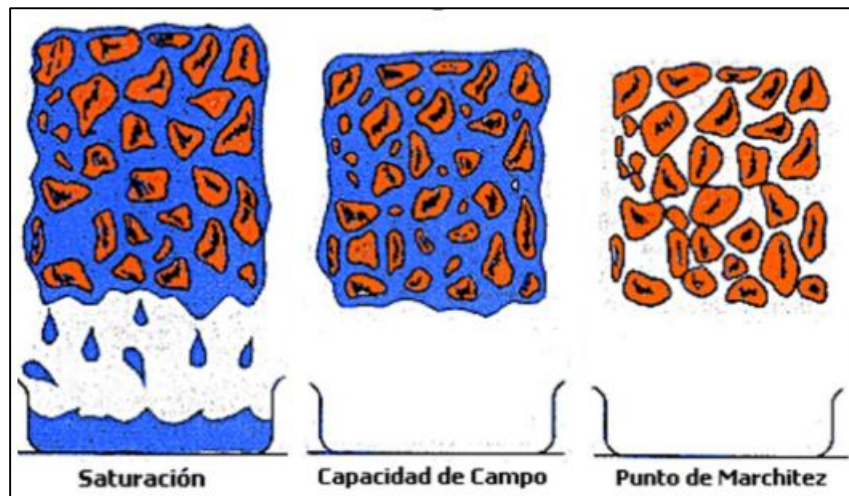
**Punto de Saturación.-** Es la capacidad máxima de absorción del recurso hídrico que la planta puede soportar antes de que se produzca un desbordamiento o filtración profunda.

**Capacidad de Campo (CC).-** Se refiere al contenido de humedad presente en un suelo en contra de las fuerzas de gravedad, dos o cuatro días después de una lluvia fuerte o un riego abundante y de que ha drenado el agua gravitacional; depende de la textura, contenido de materia orgánica y compactación del suelo. (Leitón, 1985)

**Punto de Marchitez Permanente (PMP).-** Es el punto de humedad mínima donde la planta no puede extraer agua y causa su marchitez permanente sin opción a recuperarse.



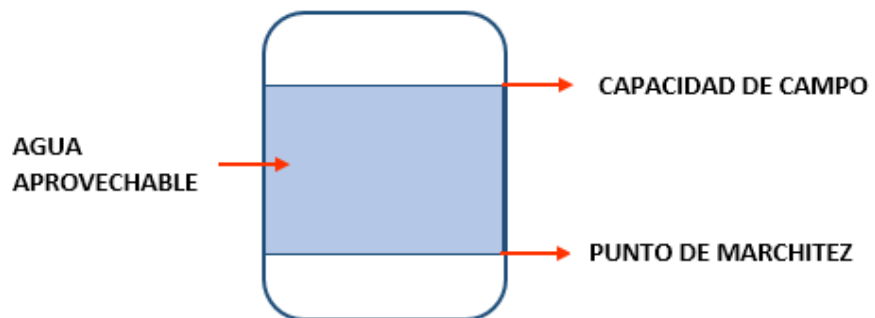
Los diferentes estados de humedad se observa en la *Figura 3-1*.



**Figura 3-1** Estados de humedad del suelo  
Fuente: (Agrounica, 2011)

#### 1.2.3.1 *Humedad aprovechable (HA)*

La diferencia entre la Capacidad de Campo y el Punto de Marchitez equivale a la cantidad de humedad aprovechable para el cultivo, si la textura es fina mayor es el porcentaje de agua en el suelo. En la *Figura 4-1* se muestra que la fracción de humedad aprovechable aumenta dependiendo de la estructura del suelo.



**Figura 2-1** Humedad aprovechable  
Realizado por: PAREDES, Marjorie, 2017

Los valores promedios de las propiedades físicas de los suelos según la textura se pueden observar en la *Tabla 1-2* donde la textura franco arenoso tiene una velocidad de infiltración de 25 mm/h.

**Tabla 2-1** Propiedades físicas de los suelos según la textura

<b>TEXTURA</b>	<b>VELOCIDAD INFILTRACIÓN mm/h</b>	<b>CAPACIDAD DE CAMPO %</b>	<b>PUNTO DE MARCHITEZ %</b>
<b>ARENOSO</b>	50 (25 o más)	9 (6 – 14)	4 (2 - 6)
<b>FRANCO ARENOSO</b>	25 (13-40)	14 (10 -18)	6 (4 - 8)
<b>FRANCO</b>	13 (7-20)	22 (18 – 26)	10 (8 - 12)
<b>FRANCO ARCILLOSO</b>	8 (2-15)	27 (23 - -31)	13 (11 - 15)
<b>ARCILLOSO</b>	0.5 (0.1-1)	35 (31 - 39)	17 (15 - 19)

Fuente: (Navarro, 2014 pág. 64)

### **1.3 Sistema de Riego**

Es el conjunto de estructuras que permiten satisfacer las necesidades hídricas de los cultivos a nivel de las raíces. En la actualidad existe una gran variedad de sistemas de riego los cuales se detallan a continuación.

#### **1.3.1 Riego por Aspersión**

Este tipo de riego consiste en un aspergado superficial sobre las plantas imitando a la lluvia con la finalidad de absorción en el lugar donde cae el agua, teniendo un alcance de seis metros desde el centro del aspersor como se muestra en la *Figura 5-1*. Existen dos tipos de aspersores, los emergentes que son aquellos que se levantan del suelo cuando surge la necesidad de riego y al culminar se retraen; los móviles se adaptan a una manguera para ser movilizadas de un lugar a otro. El inconveniente de este sistema es que los vientos dificultan la uniformidad del riego.



**Figura 3-1** Riego por aspersión  
Realizado por: PAREDES, Marjorie, 2017

#### Ventajas del riego por aspersión

- Se adapta a todo tipo de terreno y cultivo
- Ahorro en mano de obra por la automatización
- Se adapta a rotaciones de cultivos
- Tiene la posibilidad de sectorizar el riego
- Simula el riego por lluvia

#### **1.3.2** *Riego por Surcos*

Es la técnica que mejor se adecua a los cultivos sembrados en hileras como se muestra en la *Figura 6-1*, consiste en la movilización del agua por gravitación por canales con pendientes paralelos al cultivo, este sistema no utilizan ninguna fuente de energía para su funcionamiento. Los principales inconvenientes de esta técnica son el alto desperdicio de agua por infiltración, el desgaste del suelo por la movilización en los surcos y el sobre humedecimiento de ciertas áreas además que el terreno debe tener pendientes para la movilización del agua.



**Figura 4-1** Riego por surcos  
Realizado por: PAREDES, Marjorie, 2017

Ventajas del riego por surcos

- No necesita grandes inversiones en equipos.
- Ahorro de energía
- No necesita automatización
- Reducción de la escorrentía superficial
- Tiene una eficiencia de aplicación del 70 %

### **1.3.3 Riego por Goteo**

Consiste en la aplicación del agua gota a gota a una presión casi nula, en el área radicular de la planta, por medio de un sistema de mangueras que tiene acoplados o insertos unos dispositivos llamados “goteros”, colocados a distancias relativamente cortas como se muestra en la *Figura 7-1* (Leitón, 1985). El sistema permite mantener el suelo en condiciones de humedad óptima entre saturación y capacidad de campo, para alcanzar un buen desarrollo del cultivo y facilita labores como fertilización y aplicación de agroquímicos.



**Figura 5-1** Riego por goteo  
 Realizado por: PAREDES, Marjorie, 2017

Las ventajas de este sistema según (Medina, y otros, 2005) son:

- Permite un ahorro considerable de agua debido a la reducción de la evapo-transpiración y de las pérdidas de agua en las conducciones y durante la aplicación.
- Permite el crecimiento uniforme de las plantas ya que reciben volúmenes iguales de agua.
- Es posible mantener el nivel de humedad en el suelo constante.
- Facilita el control de malas hierbas, ya que éstas se localizan tan solo en el área húmeda.
- Reduce el consumo de agua en un 60 por ciento.
- Permite la aplicación de fertilizantes a través del riego.
- Consigue una mejor calidad del producto y aumenta las cosechas hasta en un 40 por ciento.
- Se impulsa el agua mediante el bombeo, el gasto energético es menor, debido a la reducción de los consumos de agua y a las menores necesidades de presión.
- Los equipos tienen larga vida útil, superior a los 10 o 15 años.

#### 1.3.4 *Tabla comparativa de tipos de riego*

En la *Tabla 1-3* se aprecia la comparación de los diferentes tipos de riego determinando que el riego por goteo tiene una eficiencia del 95 % pero tiene un costo de instalación elevado.

**Tabla 2-1** Tabla comparativa de tipos de riego

	<b>Riego por aspersión</b>	<b>Riego por surcos</b>	<b>Riego por goteo</b>
<b>Eficiencia de aplicación (%)</b>	60	40	95
<b>Vida útil de equipos (años)</b>	5-7	Por temporada	10-15
<b>Ahorro de agua (%)</b>	40	30	80
<b>Costo de instalación</b>	Alto	Bajo	Alto
<b>Costo mantenimiento</b>	Alto	Bajo	Bajo

Realizado por: PAREDES, Marjorie, 2017

## **1.4 Sistemas existentes para riego**

Existen dos tipos de sistemas, manuales y automáticos. Los manuales son aquellos que necesitan la intervención de un individuo para su funcionamiento, a diferencia de los automáticos que requieren de un diseño e implementación previa para acoplarse a un tipo de riego en particular y hacer del funcionamiento sin la intervención humana.

### **1.4.1 Sistemas comerciales de riego automático**

Consiste en la sustitución de controles manuales por automáticos, las principales ventajas son el ahorro de agua, ahorro de mano de obra y aumento de productividad del cultivo. Existen una gran variedad de sistemas automáticos disponibles en el mercado como se muestran en la *Figura 8-1* los cuales se detallan a continuación:

**SISTEMA AUTOMÁTICO DE RIEGO POR GOTEO DIY:** Este es un sistema que logra satisfacer sembríos amplios cuenta con cabezales que ofrece un ahorro del 70 % de agua, incluye con un temporizador eléctrico para programar el riego y se incluye un sistema que permite enviar nutrientes directamente a la raíz.

**AQUA CONTROL C4061 CON PROGRAMADOR:** Este sistema cuenta con goteros autocompensados, tienen el mismo caudal en todos los goteros sin importar la distancia de la fuente de agua, cuentan con un kit para 12 plantas; incluye un reductor de presión, micro tubos, adaptadores y un programador de riego el cual es el encargado del suministro de agua previo a una configuración.

**DGREAT CON KIT DE REFRIGERACIÓN POR BRUMA:** Si de precisión se trata este sistema es el más óptimo, su estructura permite un riego gota a gota ahorrando un 75 % de agua en comparación de otros sistemas, cuenta con cabezales de repuesto cubriendo grandes áreas, consta además con un flujo constante aumentando la absorción y la productividad, asimismo se puede enviar soluciones de nutrientes directamente a la raíz.

**AQUA CONTROL POR GOTEO:** Es un sistema de goteo por precisión que cuenta con un programador de grifo además se puede fijar la frecuencia y duración de riego, proporciona un caudal de 16 litros por minuto, además tiene más de diez frecuencias programables.

**BLUMFELDT GREENKEEPER:** Este sistema cuenta con una bomba de baja tensión de 12 voltios que incluye un filtro, se puede ajustar la frecuencia de riego y el intervalo de duración individualmente, la frecuencia de riego puede ser de una a tres veces por día con una duración de 15 segundos a 10 minutos; cuenta además con su respectivo control que es quien programa de acuerdo a las necesidades.

Los sistemas detallados anteriormente son en su mayoría para uso doméstico, los de uso industrial no poseen detalle de equipos ni precios al público sin antes un análisis del terreno y su extensión.



**Figura 6-1** Sistemas comerciales de riego automático

Realizado por: PAREDES, Marjorie, 2017

## 1.5 Análisis de riego en el país

En el Ecuador la demanda de agua en su mayoría se concentra en el sector agrícola con un 81 %, 13 % en el uso doméstico y un 6 % para el uso industrial. Ecuador cuenta con una superficie cultivada de 6,3 millones de hectáreas (ha) y el potencial de riego corresponde a la mitad con 3,1 millones de hectáreas. Sin embargo los sistemas implementados por el país cubren la necesidad de 1,5 millones de hectáreas y tan solo 942 mil hectáreas son efectivamente regadas, la falta de mantenimiento a la infraestructura da como resultado 600 mil hectáreas subutilizadas. El riego influye de gran manera a la productividad, los terrenos que no poseen un sistema de riego pueden producir hasta tres productos mientras que uno que posee puede llegar a 67. (El Telégrafo, 2014)

### **1.5.1 Problemática**

En Ecuador la gran mayoría de cultivos principalmente en la región sierra son cultivados por medianos y pequeños agricultores que no cuentan con ningún sistema de riego dado a su elevado costo de adquisición. La superficie regada de estos cultivos corresponde a 300 mil hectáreas y solo el 5 % tiene un riego tecnificado. Los agricultores de la sierra ecuatoriana en su gran mayoría dependen aún del agua de lluvia para la hidratación de sus cultivos, por consecuencia sus productos dependen de la época del año. Si los agricultores tuvieran acceso a un sistema de riego podrían sembrar y producir todo el año aumentando así la productividad y a la vez existiría una regulación de precios de entrega al público (El Telégrafo, 2014).

### **1.5.2 Trabajos realizados en Riobamba sobre sistemas de riego**

Riobamba siendo la capital de Chimborazo provincia de gran actividad agrícola del país presenta problemas al momento de regar los cultivos, ya que no cuenta con sistemas de riego a disposición de los agricultores. La implementación de sistemas de riego automático es necesario para facilitar el correcto crecimiento y desarrollo de las plantas, además del ahorro significativo del agua. El Centro Experimental del Riego se encuentra dentro de los predios de la ESPOCH perteneciente a la Facultad de Recursos Naturales realiza estudios de los diferentes tipos de sistemas de riego existentes con el propósito de utilizar el que mayor ahorro de recurso hídrico presente.

Entre los trabajos que se han realizado en la ciudad de Riobamba se menciona una propuesta sobre un sistema de riego automático utilizando un algoritmo de lógica difusa que activa o desactiva las electroválvulas para el riego dependiendo de las lecturas de los sensores ambientales (GÁLVEZ MOROCHO, y otros, 2016). Esta investigación se centra únicamente en la automatización del riego y no orienta su diseño a la generación de alertas a dispositivos móviles, ni al uso de energía solar como medio de alimentación para el sistema; además no genera información por zonas, información que es necesaria para el tratamiento por sectores del cultivo ni cuenta con un registro del caudal de agua.



## **1.6 Redes inalámbricas**

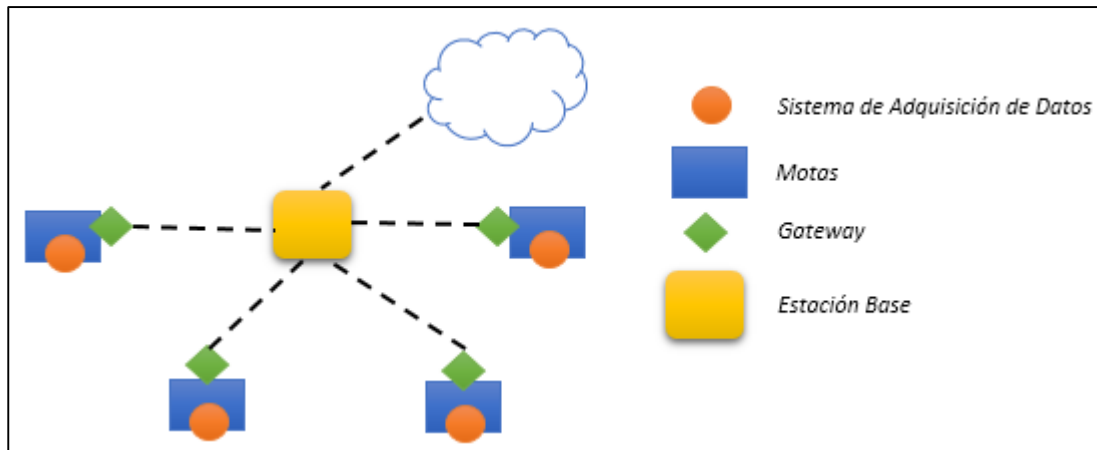
Son aquellas que se comunican por un medio de transmisión no guiado (sin cables) mediante ondas electromagnéticas. Las redes inalámbricas, tienen como ventaja, la rápida instalación de la red sin la necesidad de usar cableado, permiten la movilidad y tienen menos costes de mantenimiento que una red convencional (Ruiz, y otros, 2010).

## **1.7 Red de Sensores Inalámbricos**

Una WSN (*Wireless Sensor Network*, por sus siglas en inglés) es un red inalámbrica de dispositivos de sensado. Son sistemas distribuidos constituidos por dispositivos de bajo consumo de energía, con capacidades de sensado y comunicación. Los dispositivos que forman dichas redes se les denominan nodos sensores o motas (motes) y están limitados en su capacidad computacional y de comunicación. Sin embargo trabajan de forma colaborativa para llevar la información de un punto a otro de la red transmitiendo pequeños mensajes de un nodo a otro. Pueden estar constituidas por cientos o miles de nodos sensores, la mayoría de las aplicaciones de las WSN están relacionadas al monitoreo y detección de fenómenos. En estas aplicaciones las redes de sensores no pueden operar completamente aislados y deben contar con alguna forma de monitoreo y acceso a la información adquirida por los nodos de la red de sensores (Ruiz, y otros, 2010).

### **1.7.1 Elementos de una red de sensores inalámbrica (WSN)**

Una red de sensores cuenta con cinco elementos fundamentales como se muestra en la *Figura 9-1* como son: sistema de adquisición de datos, motas, gateway y estación base.



**Figura 9-1** Elementos de una WSN  
 Realizado por: PAREDES, Marjorie, 2017

**Sistema de adquisición de datos:** En el mercado existe una amplia gama de sensores para la medición de diversos parámetros como humedad, luz, temperatura, caudal de agua, velocidad del viento, etc. Estos sensores toman la información que posteriormente es convertida en señales eléctricas.

**Motas:** son la agrupación de sensores y otros dispositivos electrónicos, encargados de realizar la comunicación del nodo sensor, toma de datos y envío de información a la estación base.

**Gateway:** Es el que logra una interconexión entre una red TCP/IP y la red de sensores, opera como puente entre redes diferentes.

**Estación base:** Es la encargada de la recolección de información que es enviada desde las motas y a la vez puede transmitir dicha información a una página web o a una interfaz gráfica de un software.

### 1.7.2 Tecnologías

Los estándares más conocidos utilizados en redes WSN son: IEEE 802.11b (WiFi), IEEE 802.15.1 (Bluetooth) e IEEE 802.15.4 (ZigBee), usan la banda 2.4 GHz, a continuación en la *Tabla 1-4* se muestra la comparativa de estas tecnologías resaltando a la tecnología Zigbee por su mayor alcance a comparación de las otras tecnologías.

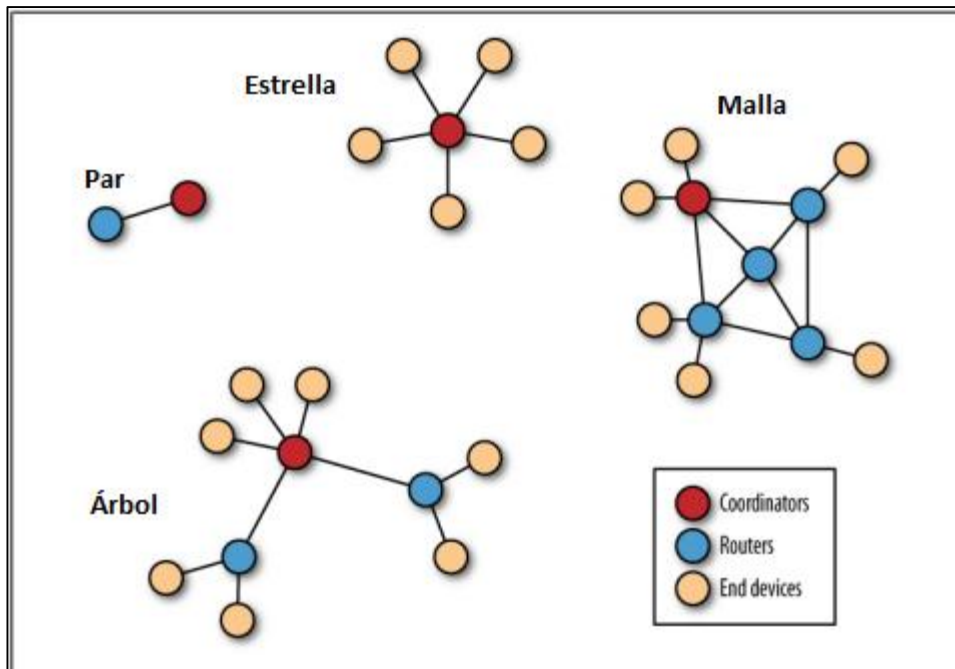
**Tabla 4-1** Comparación entre tecnologías inalámbricas Wi-fi, Bluetooth y zigBee

<b>COMPARACIÓN ENTRE TECNOLOGÍAS</b>			
	WiFi (IEEE 802.11 g)	Bluetooth (IEEE 802.15.1)	ZigBee (IEEE 802.15.4)
Técnica de propagación	DSS (Direct sequence spread spectrum)	FHSS (frequency hopping spread spectrum)	DSS (Direct sequence spread spectrum)
Velocidad	54 Mbps	24 Mbps (v3.0)	250 Kbps
Nº de nodos	32	7	64.000
Latencia	Up to 3 s	Up to 10 s	30 ms
Tipo de datos	Video, audio, gráfico, película, archivos.	Audio, gráficos, película, archivos.	Datos
Alcance (m)	100	50-150 (Smart/LE)	70-100
Expansión	Roaming	No	si
Complejidad	Complejo	Muy complejo	Sencillo
Aplicación Principal	WLAN	WPAN	Control y monitorización
Memoria necesaria	1MB +	250 KB+	4 KB – 32 KB
Parámetros importantes	Velocidad y Flexibilidad	Costes y perfiles de aplicación	Fiabilidad, bajo consumo, bajo coste
Consumo de potencia	400 ma transmitiendo 20 ma en reposo	40 ma transmitiendo 0.2 ma en reposo	30 ma transmitiendo 3 ma en reposo

Fuente: (MFBARCELL, 2016)

### 1.7.3 Topologías de WSN

Es la agrupación y comunicación entre nodos que sean capaces de transmitir y recibir información de manera inalámbrica, cada una de las agrupaciones posee un nodo control y un número de nodos esclavos, mediante una configuración de hardware y software se logra la comunicación que puede ser usada para la implementación de una red de sensores inalámbricos. Existen cuatro topologías según (Faludi, 2010) como se muestra en la *Figura 10-1*, como son Par (Pair), estrella (Star), malla (Mesh) y árbol (Tree), donde las topologías par y árbol son las menos representativas.



**Figura 7-1** Topologías ZigBee: par, estrella, malla y árbol  
Fuente: (Faludi, 2010)

**Topología Par (Pair):** Es la red más simple cuenta con un nodo control y un nodo final, es la topología menos usada.

**Topología Árbol (Tree):** Consiste en combinar topologías como estrella y malla, resaltando la simplicidad y bajo consumo de la primera y la escalabilidad y tolerancia a fallos de la segunda.

**Topología Estrella (Star-Monosalto):** Es la topología más usada por su simplicidad, consiste en un nodo coordinador que es el centro de la topología y este se conecta a un círculo de dispositivos finales, la información del sistema pasa por el coordinador ya que los nodos finales no se comunican entre sí directamente. La información en esta topología da un solo salto y los nodos comúnmente se encuentran a una distancia de 30 a 100 metros. El nodo coordinador cumple con la función de transmitir datos hacia el exterior para un monitoreo de la red. Es la que mayor ahorro energético presenta, pero sin embargo la distancia es una de sus limitaciones además debido a su diseño no cuentan con caminos alternativos en caso de que un nodo final caiga esto ocasionaría la pérdida permanente de información de dicho nodo.

**Topología Malla (Mech)**

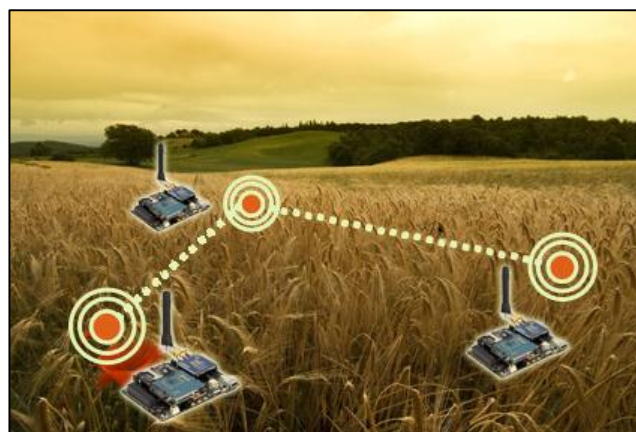
Esta topología es denominada multisalto ya que existe una comunicación entre nodos sin la necesidad de la intervención del coordinador, además del coordinador se incrementa un

enrutador que es el encargado de gestionar la red, estos nodos son capaces de enviar información a otros enrutadores y dispositivos finales dependiendo de la necesidad de la red. Los nodos finales de esta topología pueden adaptarse a un enrutador cualesquiera o al nodo central, dichos nodos pueden enviar y recibir información pero tienen dependencia de sus nodos padres para comunicarse con otros nodos. Este tipo de topología es tolerante a fallos dado que los nodos cuentan con varios caminos y si algún nodo falla la información será enviada por una nueva ruta, dado a la redirección de información puede existir periodos elevados de espera.

#### 1.7.4 Aplicaciones de WSN

Las aplicaciones principales se detallan a continuación:

**Agricultura:** En la agricultura es donde más apogeo ha tenido la red de sensores ya que con la gran variedad existente en el mercado han facilitado la vida al hombre con la recolección de datos que no están a simple vista, colaborando con la reducción en el consumo de agua y pesticidas y de esta manera contribuyen con el medio ambiente, con esta tecnología se puede monitorear parámetros como la humedad del suelo para saber la cantidad de agua necesaria para dicho cultivo, otros factores a tomar en cuenta es la cantidad de fertilizante o pesticida necesario para la planta, control de flujo de agua y decisión del momento de la cosecha como se muestra en la *Figura 11-1*. La ventaja del uso de estos sensores es la detección temprana y el accionar inmediato dando soluciones óptimas en las zonas donde surge la necesidad.



**Figura 11-1** Aplicaciones en la agricultura  
Fuente: (Management, 2017)

**Monitoreo del Medio Ambiente:** La red de sensores son empleados para el seguimiento de animales, monitoreo de condiciones ambientales y prevención de desastres, además para estudios de contaminación, detección de inundaciones, incendios forestales y sismos.

**Salud:** Las WSN en esta área son utilizadas para diversas actividades como diagnóstico de enfermedades a tiempo, monitoreo de pacientes, administración de medicina y un seguimiento continuo de enfermedades.

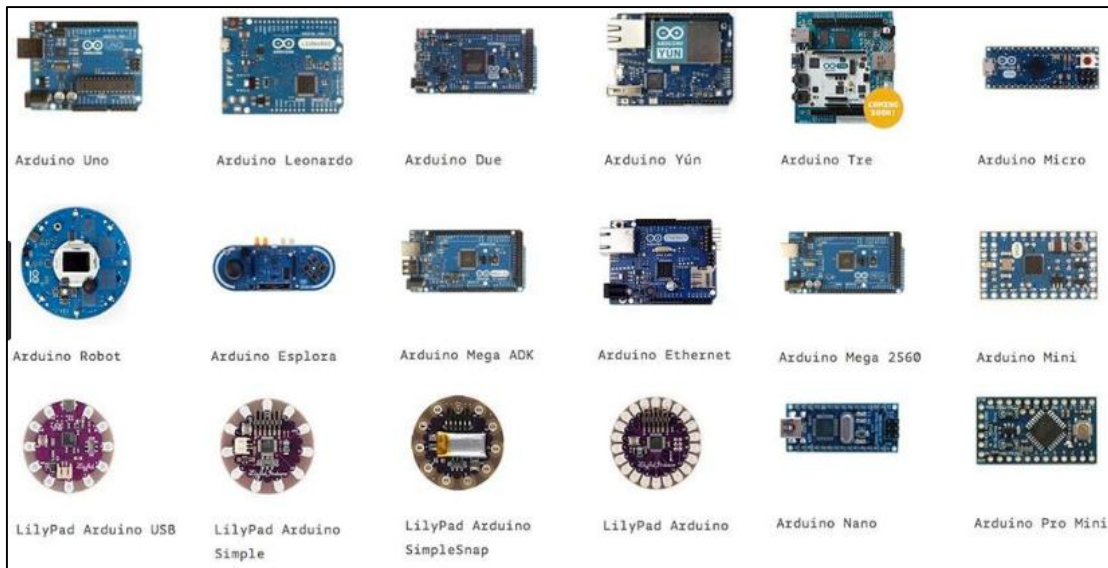
**Militar:** Esta área de las redes de sensores inalámbricos empezó a ser investigada gracias a las necesidades militares, ya que requerían tener conocimiento en tiempo real del campo de batalla para el control, las comunicaciones y toma de decisiones, además de la vigilancia del campo de batalla, seguimiento de soldados y detección de ataques químicos.

### **1.7.5 Tarjetas de desarrollo aplicadas a redes de sensores inalámbricos**

Son placas electrónicas con circuitos impresos, cumplen diferentes funciones que son asignadas por los fabricantes, son compatibles con una gran cantidad de dispositivos como son sensores, tarjetas micro SD y módulos inalámbricos con diferentes tecnologías. Constan de entradas y salidas analógicas y digitales, reguladores de voltaje, USB, seriales, indicadores de transmisión y recepción, memoria de almacenamiento, plug de alimentación entre otras características incluidas por el fabricante, además todas las tarjetas cuentan con un *Software* de programación propio. En el mercado se puede encontrar una gran variedad de tarjetas de desarrollo que a lo largo del tiempo han ido mejorando sus diseños y versiones para poder adaptarse a las necesidades de los usuarios; las tarjetas destacadas son: Arduino, Raspberry Pi, Libelium Waspote y Intel Galileo.

#### **1.7.5.1 Arduino**

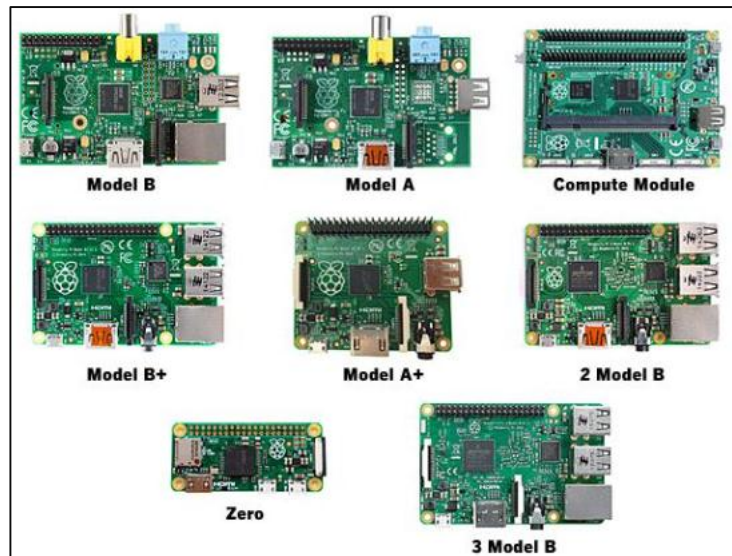
Es una plataforma electrónica de código abierto basado en hardware y software fácil de usar. Las placas arduino son capaces de leer las entradas y convertirla en una salida. Su placa envía un conjunto de instrucciones al microcontrolador y para ello se utiliza el lenguaje de programación de Arduino y el *software* de Arduino (IDE), en la *Figura 12-1* se presenta las tarjetas disponibles de Arduino. (ARDUINO, 2017).



**Figura 8-1** Tarjetas disponibles de Arduino  
**Fuente:** (César, 2014)

### 1.7.5.2 *Rapsberry Pi*

Es un mini ordenador de costo moderado a comparación de los ordenadores comunes, fue desarrollado en el reino unido en el 2011 con la finalidad de incentivar el estudio informático en escuelas ya que es una herramienta muy útil para el aprendizaje de electrónica y programación, además se puede agregar o quitar accesorios sin alterar su funcionamiento. Utiliza Linux como sistema operativo, además de los puertos de entrada y salida, entrada USB, *Ethernet* y salida HDMI tiene a disposición accesorios como pantallas, teclado y *mouse*; Raspberry Pi tiene una variedad de modelos como se muestra en la *Figura 13-1*.



**Figura 9-1** Modelos de Raspberry Pi  
 Fuente: (Gibbs, 2016)

### 1.7.5.3 Libelium Waspote

Es una plataforma modular de *software* libre que permite crear redes de sensores inalámbricos, su arquitectura fue diseñada para funcionar con un consumo muy bajo como es ( $7\mu A$ ) siendo esta su mayor ventaja entre las demás tarjetas de desarrollo disponibles en el mercado, cuenta con 7 entradas analógicas y 8 entradas/salidas analógicas. Waspote que se aprecia en la *Figura 14-1* es el producto estrella de la gran compañía Libelium que se centra en realizar proyectos con tecnología IoT (*Internet of Things*, por sus siglas en inglés).

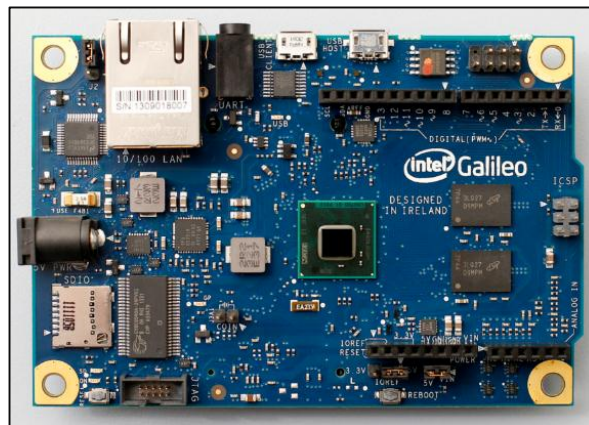


**Figura 10-1** Tarjeta Waspote  
 Fuente: (LIBELIUM, 2017)



#### 1.7.5.4 Intel Galileo

Esta tarjeta fue diseñada por arduino, es un microcontrolador basado en el procesador de aplicaciones Intel, cuenta con varios puertos de entrada y salida, también con un puerto Ethernet, puerto USB y ranura micro SD. Galileo puede programarse en IDE el *software* tradicional de arduino, en la *Figura 15-1* se muestra la tarjeta Galileo.



**Figura 15-1** Tarjeta Intel Galileo  
Fuente: (INTEL, 2015)

#### 1.7.5.5 Comparativa entre tarjetas de desarrollo

En la *Tabla 1-5* se resume las principales características de las tarjetas de desarrollo anteriormente mencionadas donde Arduino tiene un voltaje de operación de 5V y no cuenta con entrada Ethernet.

**Tabla 5-1** Comparativa de tarjetas de desarrollo

Características	Arduino Uno	Raspberry Pi (Model B)	Waspote	Intel Galileo Gen1
Procesador	ATMega328	ARM11	ATMega1281	SoC Quark X100
Voltaje de Operación	5 V	3.3 V / 5V	0.5 – 3.8 V	3.3 V / 5V
Voltaje de entrada	7 – 12 V	5V	5 – 7 V	5 V
RAM	2 KB	512 MB	8 KB	512 KB
Memoria Flash	32 KB	-	128 KB	8 MB
USB	1	2	1	2
Velocidad de reloj	16 MHz	700 MHz	16 MHz	400 Mhz
E/S Analógicas	6	-	7	6
E/S Digitales	14	8	8	14
Ethernet	No	10/100	-	10/100
Precio aproximado	\$30	\$50	\$180	\$90

Fuente: (ALTAMIRANO BENALCÁZAR, y otros, 2016)

## CAPÍTULO II

### 2 DISEÑO DEL PROTOTIPO DEL SISTEMA INALÁMBRICO DE CONTROL AUTOMÁTICO DE RIEGO

En este capítulo se presenta el diseño del prototipo del sistema inalámbrico de control automático de riego (SICAR), los respectivos diagramas de bloques de cada uno de los nodos, además del detalle técnico de los componentes correspondientes a SICAR, y finalmente se muestra los esquemas de los nodos que conforman el sistema.

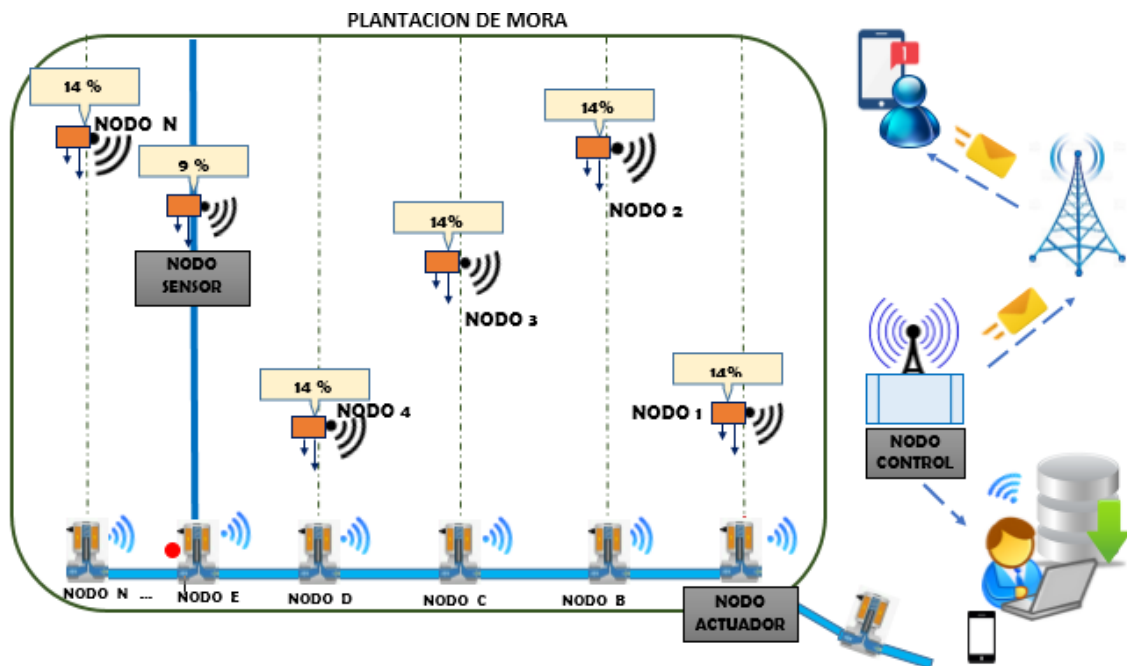
#### 2.1 Requerimientos del SICAR

Bajo el fundamento del estudio realizado en el capítulo anterior se puede definir los requerimientos de diseño del sistema inalámbrico de control automático de riego para el centro experimental de riego en la ESPOCH. Los requerimientos se enlistan a continuación:

- Implementar un sistema de bajo costo con equipos y materiales de fácil adquisición, que sea escalable independientemente del cultivo.
- Censar la humedad del suelo y la temperatura ambiente.
- Monitorear el cultivo por zonas específicas empleado nodos móviles e independientes con comunicación inalámbrica.
- Proveer información en tiempo real y permitir el almacenamiento periódico de los valores de humedad del suelo y temperatura ambiente.
- Enviar alertas a través de mensajes de texto utilizando la red GSM (*Global System for Mobile Communications* por sus siglas en inglés) cuando el nodo actuador esté en funcionamiento incluyendo las lecturas de los sensores.
- Visualizar la cantidad de agua en litros utilizada por el cultivo.

### 2.1.1 Concepción de la arquitectura general del sistema.

La propuesta para el SICAR se muestra en la *Figura 1-2* donde se aprecian los tres nodos principales que conforman el sistema: nodo sensor, nodo actuador y nodo control. La comunicación entre los nodos es a través de ZigBee.



**Figura 1-2** Arquitectura general del sistema inalámbrico  
Realizado por: PAREDES, Marjorie, 2017

**Nodo Sensor:** Tiene como objetivo la adquisición en tiempo real de niveles de humedad del suelo y la temperatura ambiente en una zona determinada del cultivo. Cada nodo sensor cuenta con alimentación independiente proveniente de un panel solar, los datos obtenidos por los sensores son visualizados en una pantalla LCD (*Liquid Cristal Display por sus siglas en inglés*) cuentan además con un módulo transmisor XBee que es el encargado de la transmisión inalámbrica de datos. Los sensores utilizados para la obtención de datos son: para la humedad un higrómetro FC-28 y para temperatura un DTH11, los sensores están conectados a la tarjeta de desarrollo la cual es encargada de recibir datos analógicos y transformar estos datos en una trama que serán enviados por XBee hacia el nodo control utilizando la tecnología de comunicación inalámbrica ZigBee. La cantidad de nodos dependerá de la extensión del terreno y necesidades del cultivo.

**Nodo Actuador:** Recibe la orden del nodo control a través del módulo XBee para empezar el riego activando una electroválvula, además cuenta con un medidor de flujo para llevar un registro de la cantidad de agua utilizada por el cultivo en un lapso determinado de tiempo.

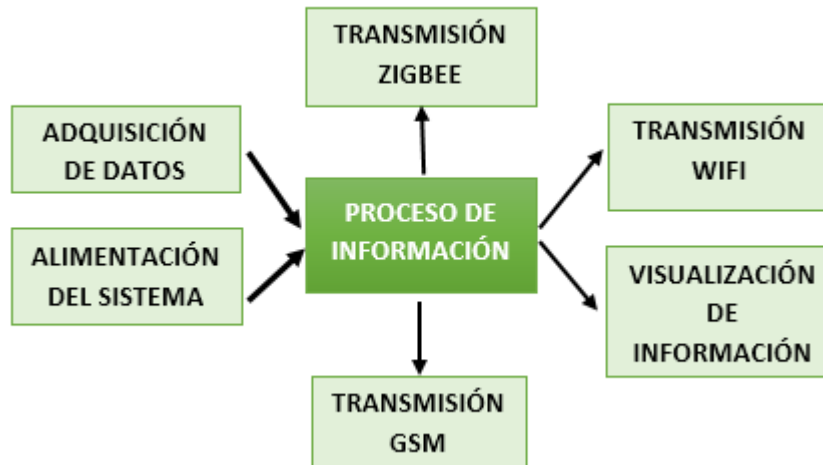
**Nodo Control:** Es el encargado de la recolección de información, actúa como transmisor y receptor, recibe información del nodo sensor y genera la orden para el nodo actuador, consta de una tarjeta de desarrollo encargada de desencapsular los datos recibidos por el módulo XBee.

### **2.1.2 *Diseño de la Arquitectura del SICAR***

Definidos los requerimientos, se muestra el diseño de cada uno de los nodos que conforman el sistema inalámbrico de control automático de riego.

#### **2.1.2.1 *Nodo Sensor***

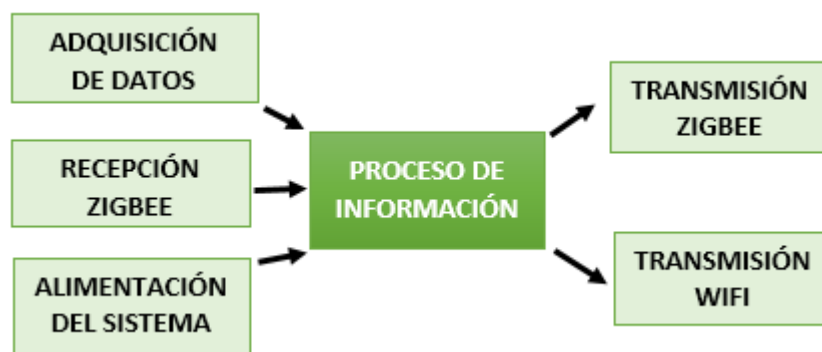
El la *Figura 2-2* se muestra el diagrama de bloques del nodo sensor, consta de cinco bloques: adquisición de datos que es el encargado de la lectura de valores entregados por los sensores de humedad y temperatura, visualización que haciendo el uso de un LCD muestra los valores de los sensores ambientales en tiempo real, esta información también es entregada al bloque de transmisión; este bloque utiliza la tecnología inalámbrica ZigBee para la comunicación con el nodo control. Además cuenta con un bloque de alimentación el cual utiliza un panel solar para evitar problemas con cables y así tener un nodo independiente. El bloque de proceso de información está basado en una tarjeta de desarrollo como es Arduino nano que posee entradas analógicas y digitales, los sensores de humedad y temperatura son datos analógicos, el bloque transmisión GSM cuenta con un módulo SIM900 que es el encargado de enviar alertas a través de mensajes de texto cuando la electroválvula se abra y empiece el riego acompañado de los valores de temperatura, humedad e identificador del nodo.



**Figura 2-2** Diagrama de bloques del nodo sensor de SICAR  
 Realizado por: PAREDES, Marjorie, 2017

### 2.1.2.2 *Nodo Actuador*

En la *Figura 3-2* se muestra el diagrama de bloques del nodo actuador, consta de cinco bloques, recepción zigbee que es el encargado de entregar al de procesamiento lo que recibe de los módulos xbee del nodo control, esta información será el porcentaje de humedad del suelo y la temperatura ambiente para poder activar la electroválvula y regar la parcela, el bloque de adquisición de datos recibe la información del medidor de flujo de agua, esta información es enviada por el bloque de transmisión zigbee hacia el nodo control.

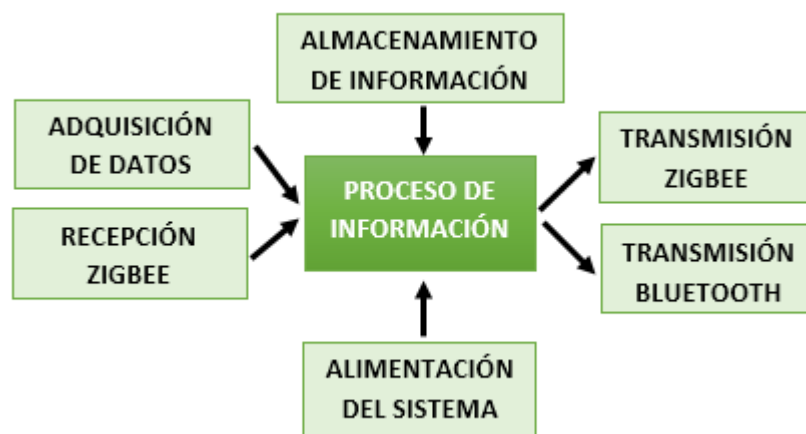


**Figura 3-2** Diagrama de bloques del nodo actuador de SICAR  
 Realizado por: PAREDES, Marjorie, 2017

El bloque de proceso de información está basado en una tarjeta de desarrollo como es Arduino nano que posee entradas analógicas y digitales, el sensor medidor de flujo y el bloque transmisión y recepción generan entradas digitales.

### 2.1.2.3 *Nodo Control*

En la *Figura 4-2* se muestra el diagrama de bloques del nodo control, consta de siete bloques, recepción zigbee que se encarga de la recolección de datos tanto del nodo sensor como del nodo actuador, transmisión zigbee es el encargado de enviar las ordenes de abrir o cerrar la electroválvula al nodo actuador, transmisión wifi está conformado por el módulo ESP8266 que transfiere información a un servidor en internet denominado *Thing Speak*, el bloque de transmisión Bluetooth envía datos de manera inalámbrica que será visualizado en tiempo real en un dispositivo móvil, el bloque de almacenamiento de información se divide en dos partes la primera hace uso de una memoria micro SD para almacenar la información, además la información estará disponible en la plataforma *Thing Speak* que es de *Software* libre, se podrá acceder a la información desde cualquier parte del mundo con tal solo un usuario y contraseña.



**Figura 4-2** Diagrama de bloques del nodo control de SICAR  
Realizado por: PAREDES, Marjorie, 2017

El bloque de proceso de información está basado en una tarjeta de desarrollo como es Arduino UNO que posee entradas analógicas y digitales, a esta tarjeta están

## 2.2 Selección de elementos que conforman el SICAR

A continuación se especifican los elementos utilizados para la implementación del prototipo del sistema inalámbrico de control automático de riego, sus hojas técnicas se detallan en el *Anexo 1*.

### 2.2.1 Tarjetas de desarrollo

Con el pasar de los años los dispositivos de procesamiento han tenido un gran cambio, empezando por el microcontrolador PIC que tiene su origen en 1975, siendo un pequeño dispositivo electrónico que cubrió grandes necesidades en su tiempo. Ahora se habla de otros dispositivos que están en su más grande apogeo denominados Arduino, son capaces de realizar el trabajo de control de procesos contando con una gran cantidad de entradas y salidas tanto analógicas como digitales (ALTAMIRANO BENALCÁZAR, y otros, 2016).

#### 2.2.1.1 Arduino UNO

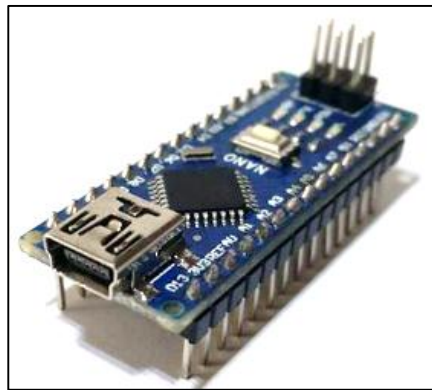
Es un microcontrolador basado en ATmega328P como se muestra en la *Figura 5-2* este dispositivo cuenta con una tensión de funcionamiento de 5 voltios y un voltaje de entrada de 7 a 12 voltios, corriente DC de 20 mA y corriente CC de 50 mA; además con 14 terminales digitales de entrada y salida, de estos; 6 proporcionan salida PWM, posee 6 entradas analógicas, un cristal de cuarzo de 16 MHz y una memoria flash de 32 KB. Utiliza software Arduino IDE para su programación (ARDUINO, 2017).



**Figura 5-2** Arduino UNO  
Realizado por: PAREDES, Marjorie, 2017

### 2.2.1.2 *Arduino NANO*

Es un microcontrolador pequeño como se muestra en la *Figura 6-2* basado en ATmega328 con una tensión de funcionamiento de 5 voltios y un voltaje de entrada de 7 a 12 voltios, consumo de energía de 19 mA y corriente CC de 40 mA; cuenta con 22 terminales digitales de entrada y salida, de estos; 6 proporcionan salida PWM, posee 8 entradas analógicas, un cristal de cuarzo de 16 MHz y una memoria flash de 32 KB. Utiliza software Arduino IDE para su programación (ARDUINO, 2017).



**Figura 5-2** Arduino NANO  
Realizado por: PAREDES, Marjorie, 2017

### 2.2.2 *Módulos XBee*

Los módulos XBee y XBee-PRO son dos integrantes de la familia Digi RF. Se trata de módulos integrados e inteligentes, mediante los cuales es posible operar en redes 802.15.4. Estos módulos se presentan como una interesante y atractiva alternativa, acortando el tiempo de desarrollo dado que ya poseen el stack correspondiente cargado, funcionando como modems configurables mediante el set de comandos AT y pudiendo operar en un modo transparente, lo que permite (por ejemplo) reemplazar un cable. Debido a que poseen un conversor analógico a digital y salidas PWM, es posible no solo transmitir datos serie y estado de entradas digitales sino también magnitudes analógicas, dado que es posible configurar a un módulo para que sus salida sigan a las entradas de otro. (Caprile, 2009)

Tanto la Serie 2 como la Serie 2 PRO están disponibles en dos potencias de transmisión, regular y PRO (Ver *Figura 6-2*), la versión regular se llama simplemente XBee, a diferencia de la segunda que es un poco más grande y más costosa denominada XBee-PRO.



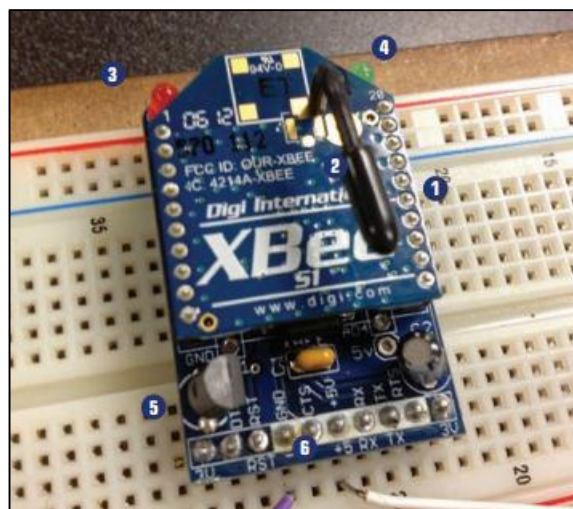


**Figura 6-2** Módulos XBee regular y módulo XBee-PRO  
**Fuente:** Faludi (2010)

Existen una gran variedad de módulos XBee, pero independientemente de la serie o modelo se resalta la ventaja que todos utilizan pines similares, la alimentación a tierra y terminales de comunicación se encuentran en el mismo lugar en todas las series.

#### 2.2.2.1 Anatomía de la XBee

Si observamos el módulo XBee que se muestra en la *Figura 7-2*, en esta se detalla sus partes principales. (Baichtal, 2014)



**Figura 7-2** XBee y sus elementos principales  
**Fuente:** (Baichtal, 2014)

**Terminales (1):** Son lo que reciben datos de arduino, cuenta con seis pines de 10 bits para entrada ADC y ocho pines Entrada/Salida digitales en el caso de la Serie 1.

**Antena (2):** Tiene múltiples opciones de antena dependiendo de la XBee, considerando la antena de cable como la mejor opción.

**LED de encendido (3):** Se activa cuando la placa se enciende cuando es energizada.

**LED de datos (4):** Este led parpadea para informar que los datos están siendo procesados por el módulo Xbee.

**Regulador de potencia (5):** El condensador y transistor controlan la potencia que entra al módulo, manteniendo la tensión uniforme para el correcto funcionamiento.

#### 2.2.2.2 Configuración módulo XBee

La configuración consiste en primera instancia que los dos módulos estén en la misma red con el mismo identificador (ID), el mismo canal (CH), además fijar el módulo destino y el canal por donde se llevará a cabo la transmisión (DH y DL) como se puede apreciar en la *Tabla 2-1* donde se tiene que el PAN id es de 1234.

**Tabla 1-2** Parámetros de configuración XBee

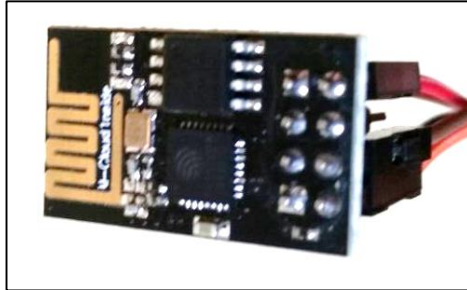
	<b>COORDINADOR</b>	<b>ROUTER</b>
<b>PAN ID</b>	1234	1234
<b>SL</b>	Predeterminado	Predeterminado
<b>SH</b>	Predeterminado	Predeterminado
<b>DL</b>	=SL(coordinador)	=SL(Router)
<b>DH</b>	=SH(Router)	=SH(coordinador)
<b>NI</b>	COR	ROU

Fuente: (SINTELWE, 2012)

#### 2.2.3 Módulo ESP8266 Wi-Fi

Es un SOC (*System on chip por sus siglas en inglés*), sistema en chip; es autónomo con pila de protocolos TCP/IP integrada que puede dar acceso a cualquier microcontrolador a su red WiFi. El ESP8266 que se muestra en la *Figura 9-2* es capaz de alojar una aplicación o descargar todas las funciones de la red Wi-Fi desde otro procesador de aplicaciones. Cada módulo ESP8266 viene preprogramado con un firmware de conjunto de comandos AT, lo que significa que

simplemente puede conectarse a un dispositivo Arduino y obtener casi la capacidad Wi-Fi que ofrece Wi-Shield. (Sparkfun, 2016)



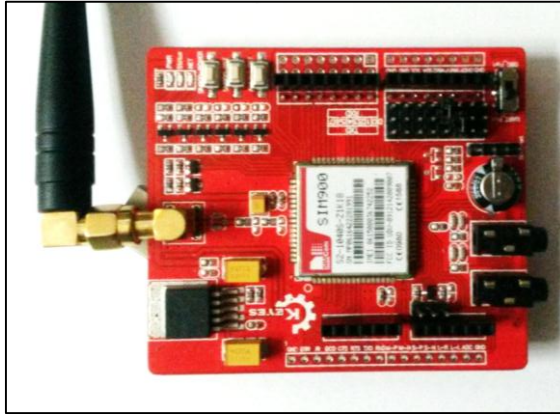
**Figura 8-2** Módulo Wi-Fi ESP8266  
Realizado por: PAREDES, Marjorie, 2017

La empresa (ESPRESSIF, 2015) facilita las especificaciones de este módulo.

- CPU Tensilica L106 32-bits
- Voltaje de operación: 3.3 V
- Corriente de operación: 80 mA
- Temperatura de operación: -40°C y 125°C
- Soporta IPV4 y protocolos TCP, UDP, HTTP, FTP
- Consumo de potencia de 0,5 uA (apagado) y 170 mA (en operación máxima)

#### **2.2.4** *Módulo SIM900*

Es una tarjeta GPRS (*General Packet Radio Service por sus siglas en inglés*), ultra compacta de comunicación inalámbrica. La tarjeta es compatible con todos los modelos de Arduino UNO, además se controla con otros microcontroladores. La tarjeta está basada en el módulo SIM900 GSM 4 como se muestra en la *Figura 10-2*. (Lara, 2015). Está configurada y controlada por vía UART usando comandos AT. ES ideal para sistemas remotos, comunicación recursiva, puntos de control y para enviar mensajes de texto a celulares.



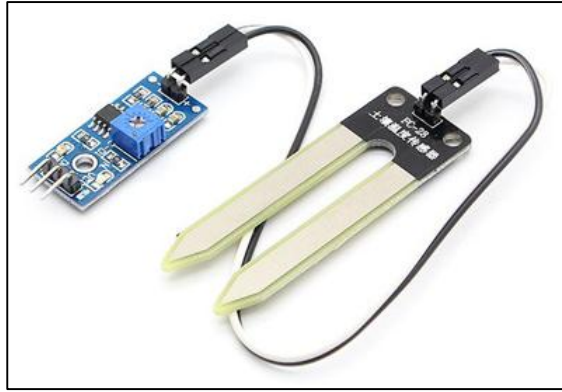
**Figura 9-2** Módulo SIM900 GSM/GPRS  
**Realizado por:** PAREDES, Marjorie, 2017

Según (Lara, 2015) las especificaciones de mayor relevancia son:

- Totalmente compatible con Arduino
- Voltaje de Alimentación: (VCC5 5V y VCC4 3.5-4.5 V)
- Interfaz: Serial TTL
- Consumo: 1.5 mA
- Conexión con el puerto serial
- Bandas de frecuencia: 850/900/ 1800/ 1900 Mhz
- Compatible con SIM: 2, 3 y 4 G
- Modos: SMS y GPRS
- Soporta el protocolo TCP/IP
- Soporta RTC (*Red telefónica conmutada*)
- Antena independiente
- Puerto conversor analógico a digital
- 12 puertos entrada/salida para (GPIO)

### **2.2.5** *Sensor de Humedad FC-28*

Mide la humedad de la tierra de forma indirecta calculando la conductividad (ElectroTools, 2017). El FC-28 como se muestra en la *Figura 11-2* se distribuye con una placa de medición estándar que permite obtener la medición como valor analógico o como una salida digital la cual se activa cuando la humedad supera un cierto umbral (Llamas, 2016).



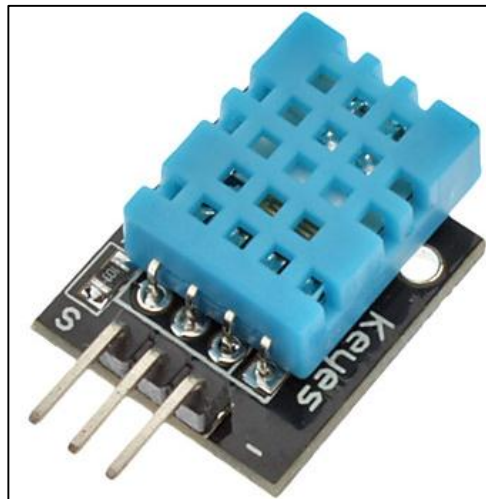
**Figura 11-2** Sensor de humedad FC-28  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas se detallan a continuación:

- Voltaje de entrada: 3.3 – 5 V
- Voltaje de salida: 0 – 4.2 V
- Corriente: 35 mA

#### 2.2.6 *Sensor de humedad DHT11*

Este sensor como se muestra en la *Figura 12-2* cuenta con una salida de señal digital calibrada, también con una medida de humedad tipo resistivo, además contiene un microcontrolador de 8 bits que ofrece una respuesta rápida y anti-interferencia.



**Figura 10-2** Sensor de humedad DHT11  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas son las siguientes:

- Rango de temperatura: 0 – 50 C°
- Humedad: 20 – 90 % de humedad relativa
- Tiempo de respuesta: 6 segundos (mínimo), 30 segundos (máximo)

### 2.2.7 Sensor de flujo de agua

En este sensor se incluyen tres cables como se muestra en la *Figura 13-2*, negro (GND), amarillo (salida de impulsos), rojo (potencia 5-24 VDC) cada pulso es aproximadamente 2,25 mililitros teniendo una previa calibración ya que la frecuencia de pulso varía dependiendo de la velocidad del flujo, orientación del sensor y presión del fluido.



**Figura 13-2** Sensor de flujo de agua  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas según (ELECTRONILAB, 2016) se detallan a continuación:

- Voltaje: 5-18 V CC
- Consumo máximo de corriente: 15mA a 5V
- Rango de trabajo: 1 a 30 litros/minuto
- Temperatura: -25°C -80°C
- Rango de Humedad: 35 % - 80% RH (Humedad Relativa)
- Pulsos por litro: 450

### 2.2.8 Módulo relé

Es un interruptor que se activa mediante una señal eléctrica, usado como control de cargas de potencia, para su activación contiene un transistor que permite controlar el encendido y apagado, además en la *Figura 14-2* se aprecia los tres pines: GND, VCC y la señal de entrada.



**Figura 14-2** Módulo relé  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas de detallan a continuación:

- Corriente de operación: 10 A
- Voltaje de bobina: 5 V
- Voltaje de funcionamiento: 24 V
- Consumo de corriente: 20 mA

### 2.2.9 Panel solar

Está elaborado de policristalino ultra eficiente, fue diseñado para proporcionar una solución de energía alternativa fiable, cuenta con 33 células de silicio cristalino, el marco está construido con aluminio anodizado resistente al viento y conveniente para el montaje, las células se laminan entre transmisividad, hierro bajo, vidrio templado de 3mm, una lámina de TPT y dos láminas de EVA para evitar la entrada de humedad. Estos paneles como se muestra en la *Figura 15-2* cuentan con 25 años de garantía.



**Figura 15-2** Panel solar

**Realizado por:** PAREDES, Marjorie, 2017

Las especificaciones técnicas según (CENTSYS, 2016) son las siguientes:

- Potencia máxima: 60W
- Tolerancia:  $\pm 3 \%$
- Voltaje de entrada: 22.2V
- Corriente: 3.5 A
- Voltaje máximo: 18.2V
- Corriente máxima: 3.3V
- Eficiencia del módulo: 14.22 %
- Eficiencia de celdas solares: 18 %
- Temperatura de operación:  $-40^{\circ}\text{C} - 85^{\circ}\text{C}$

#### **2.2.10** *Regulador de voltaje solar*

Es un controlador de voltaje que protege a la batería de sobrecargas y descargas por parte del panel solar, además regula el voltaje de entrada hacia la batería, cuenta con protección de polaridad inversa y descarga de regreso, en la *Figura 16-2* se muestra su estructura y luces led indicadoras del nivel de batería.





**Figura 16-2** Regulador de voltaje solar  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas se detallan a continuación:

- Voltaje DC: 12V/24V
- Consumo en reposo: 10mA
- Corriente de trabajo: 20 mA
- Voltaje máximo en circuito abierto: 25V/36V
- Temperatura de operación: -20° - 50°
- Sobre descarga
  - Protección: 10V/21V
  - Recuperación: 12V/24V
- Sobre carga
  - Protección: 14V/28V
  - Carga flotante: 13V/27V
  - Recuperación: 13V/26V

### 2.2.11 Batería Seca

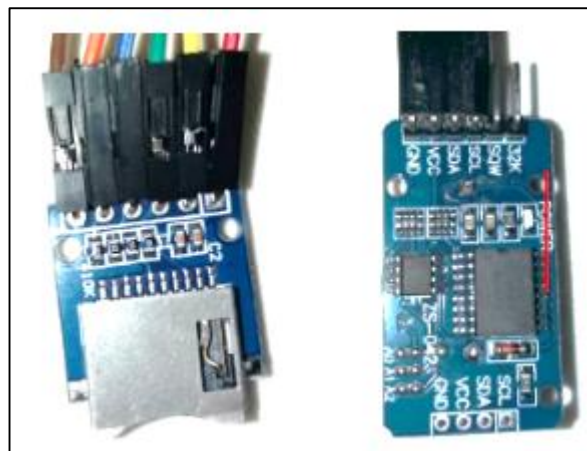
Es la agrupación de celdas galvánicas con electrolitos pastosos, almacena carga equivalente a 12V y una corriente de 4.5A, cada batería tiene especificaciones de almacenamiento de fábrica, en la *Figura 17-2* se aprecia la batería que cuenta con dos terminales positivo y negativo.



**Figura 17-2** Batería Seca  
Fuente: (Paredes, 2017)

### 2.2.12 Módulo micro SD

Permite la inserción de una memoria micro SD, cuenta con una alimentación de 3,3 V o 5 V usando los terminales respectivos, este módulo permite la lectura, escritura y edición de archivos dentro del dispositivo de almacenamiento SD, es compatible con todas las tarjetas de arduino disponibles en el mercado. La interfaz de comunicación es SPI, que es el encargado de la comunicación entre circuitos integrados (datos) a bits (ALTAMIRANO BENALCÁZAR, y otros, 2016). En la *Figura 18-2* se muestra el módulo micro SD.



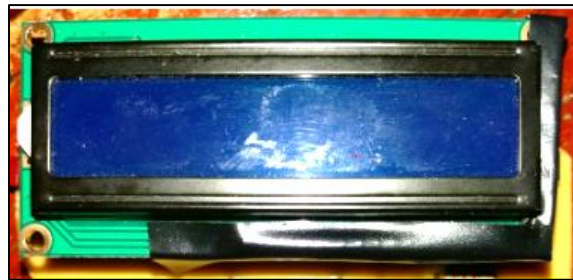
**Figura 18-2** Módulo micro SD  
Realizado por: PAREDES, Marjorie, 2017

Su distribución de terminales es la siguiente:

- CS: Activación por parte del Master al Slave
- MISO: Salida de datos de Slave y entrada al Master
- MOSI: Salida de datos desde el Master y entrada de datos al Slave
- VCC/GND : Alimentación
- SCK: Pulso que marca la sincronización, se envía un bit con cada pulso de reloj

### **2.2.13 Equipo de Visualización**

En la *Figura 19-2* se observa un LCD de dos filas y 16 columnas, que es un dispositivo electrónico de visualización de datos, donde un microcontrolador es el encargado del funcionamiento y necesita una alimentación de 5 V.



**Figura 19-2** Pantalla de visualización  
Realizado por: PAREDES, Marjorie, 2017

### **2.2.14 Electroválvula Rain Bird de 1''**

Las electroválvulas como se muestra en la *Figura 20-2* son las encargadas de la activación y suspensión del flujo de agua en un sistema de riego en particular, son un elemento fundamental en una red de sensores.



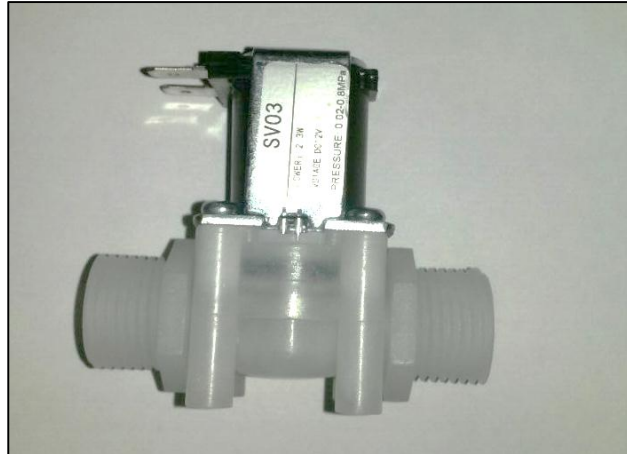
**Figura 20-2** Electroválvula de 1''  
Realizado por: PAREDES, Marjorie, 2017

Las principales características según (Deals, 2016) se muestran a continuación:

- Presión: 1 – 10,3 bares
- Caudal: 0,01 – 1,01,89 l/s
- Temperatura: hasta 43°C
- Corriente máxima de entrada: 0,290A a 60 Hz
- Corriente de retención: 0,091A a 60 Hz
- Resistencia de la bobina: 70 – 85 Ohmios (4°C – 43°C)

#### **2.2.15 Electroválvula Solenoide de ½''**

Cumple con las mismas funciones que la electroválvula de 1'' la diferencia es el caudal de agua que permite transportar y su tamaño como se muestra en la *Figura 21-2*.



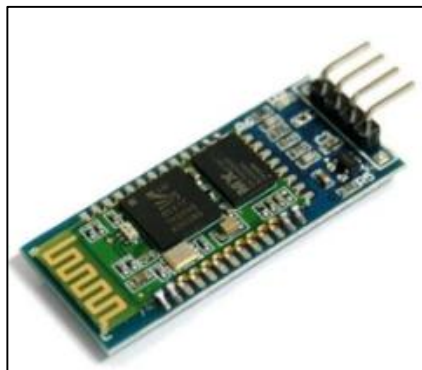
**Figura 21-2** Electrovalvula de ½ ”  
Realizado por: PAREDES, Marjorie, 2017

Las especificaciones técnicas se detallan a continuación:

- Voltaje de Alimentación 12 V
- Potencia 4.8 W
- Presión 0.8 MPa (máximo)

#### **2.2.16 Bluetooth HC-06**

En la *Figura 22-2* se muestra el módulo Bluetooth HC-06, cuenta con cuatro terminales de fácil utilización, interfaz serial, modo esclavo y con un alcance de 5 a 10 metros, este módulo utiliza 5V como voltaje de operación.



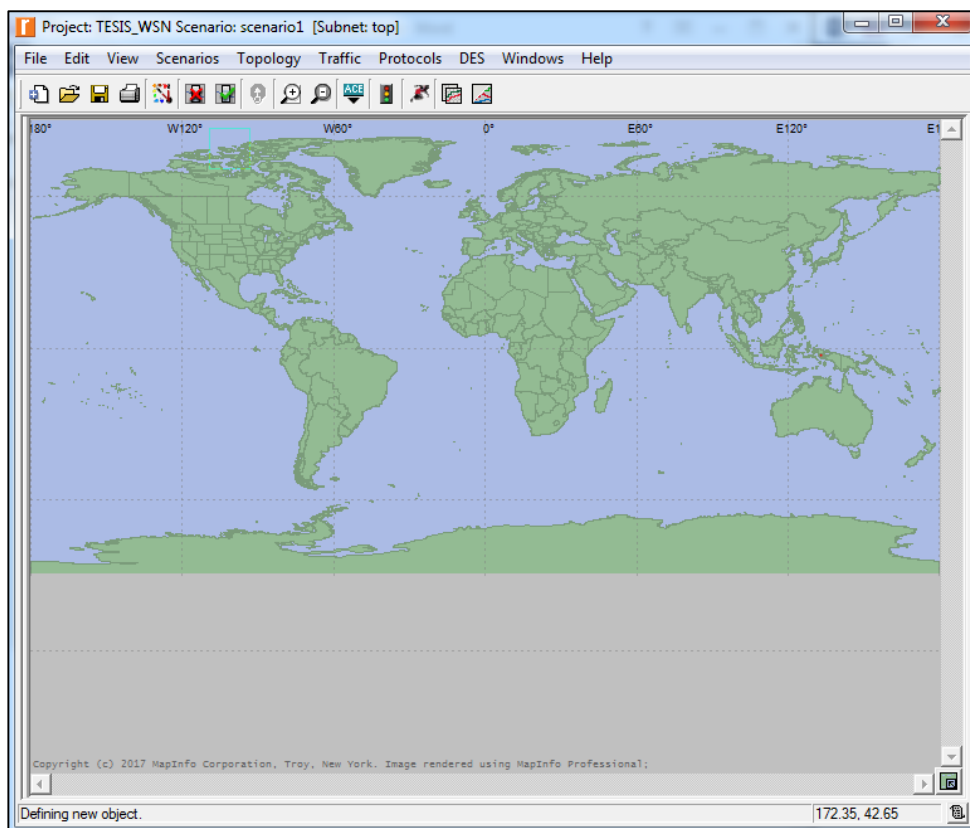
**Figura 22-2** Bluetooth HC-06  
Realizado por: PAREDES, Marjorie, 2017

## 2.3 Software para el estudio y análisis de topología de red

Para el análisis de topología de red adecuada para el SICAR se utiliza el software *Riverbed Modeler Academic* por sus siglas en inglés.

### 2.3.1 *Riverbed Modeler Academic*

Es un *software* de simulación de red con un rápido análisis y diseño, costa de un paquete de protocolos y tecnologías en un entorno sofisticado de desarrollo como se muestra en la *Figura 23-2*. En su modelado incluye varios tipos de red como: VoIP, TCP, OSPFv3, MPLS, IPv6 entre otras; Este software analiza redes, prueba y demuestra diseños tecnológicos antes de la producción. (Riverbed, 2017)



**Figura 23-2** Entorno de desarrollo de Riverbed Modeler Academic  
Realizado por: PAREDES, Marjorie, 2017

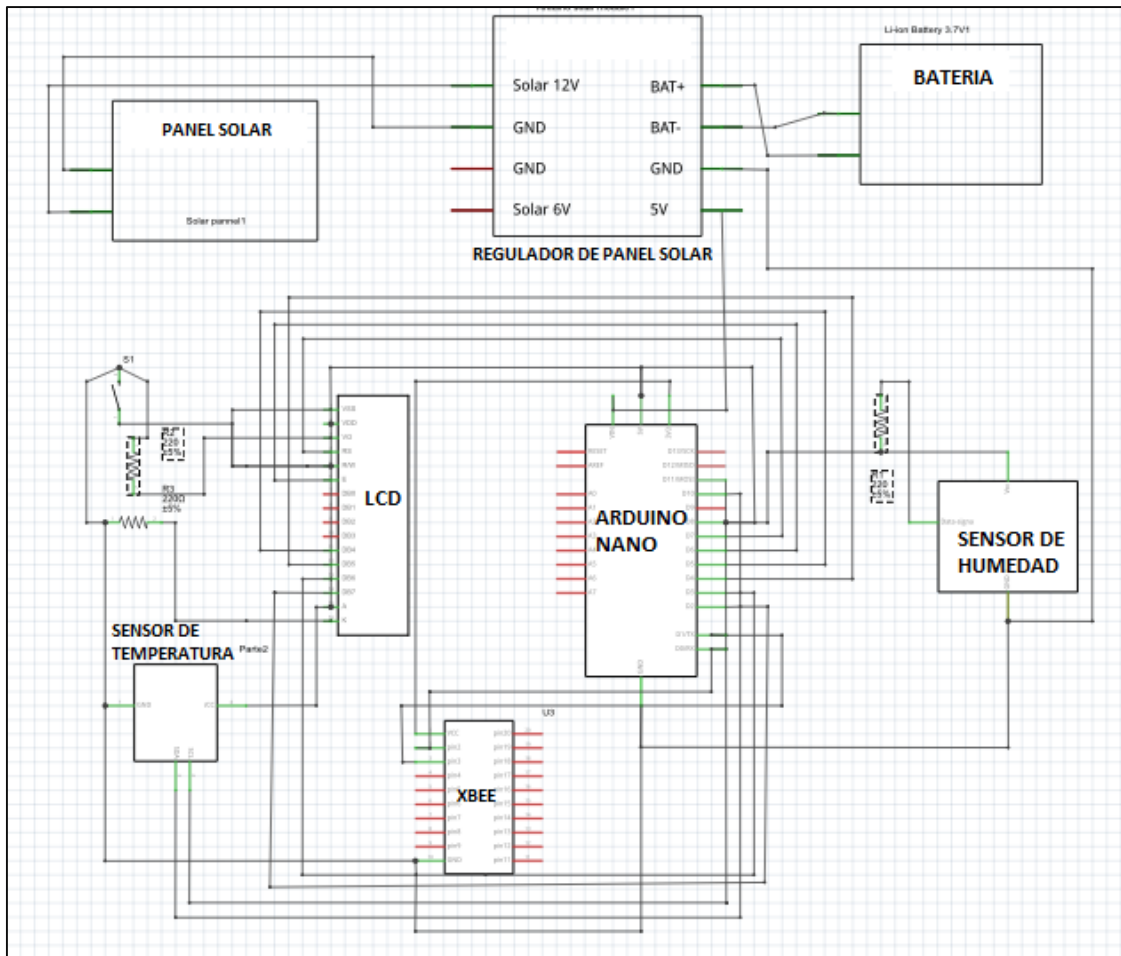
## 2.4 Esquema de conexión del Sistema Inalámbrico

A continuación del detalle de los elementos de los nodos que conforman el sistema inalámbrico de sensores se detallan las conexiones realizadas.

### 2.4.1 Diagrama de conexión del nodo sensor

Las conexiones del nodo sensor se muestran en la *Figura 24-2*. El equipo de procesamiento es el arduino NANO que tiene conectado sus terminales de la siguiente manera:

- El sensor de humedad higrómetro FC-28 tiene conectada su salida de datos D0 con el terminal A0 del Arduino, con alimentación de 5V y GND.
- El sensor de temperatura DHT11 tiene conectada su salida de datos D0 con el terminal A1 del Arduino, con alimentación de 5V y GND.
- El módulo XBee tiene una alimentación de 3.3V y GND, tiene conectada su salida al terminal 2 (Rx) y una entrada al terminal 3 (Tx) de arduino.
- El módulo SIM900 conecta su salida de Rx al terminal A2 y su salida Tx al terminal A3 de arduino, con alimentación de 5V y GND.
- Para la visualización de lectura de los nodos sensores se conectó una pantalla LCD al arduino por medio de los terminales RS(D9), E(D8), D4(D7), D5(D6), D6(D5) y D7(D4); la alimentación es de 5V.
- El micro switch se alimenta de 5 V y GND, su salida de datos está conectada al arduino por medio de los terminales 1(D10), 2(D11).
- Las tarjetas wifi MCU ESP-8266 tienen conectadas sus salidas A0 a los terminales A0 y A1 de arduino, además las salidas A0 de las tarjetas wifi están conectadas a una resistencia 220 Ohmios y estas a la alimentación GND.
- Dos LED son utilizados como luces de alerta, el led verde está conectado al terminal D12 del arduino y el rojo al terminal D13 y los terminales negativos de los leds están conectados a GND.



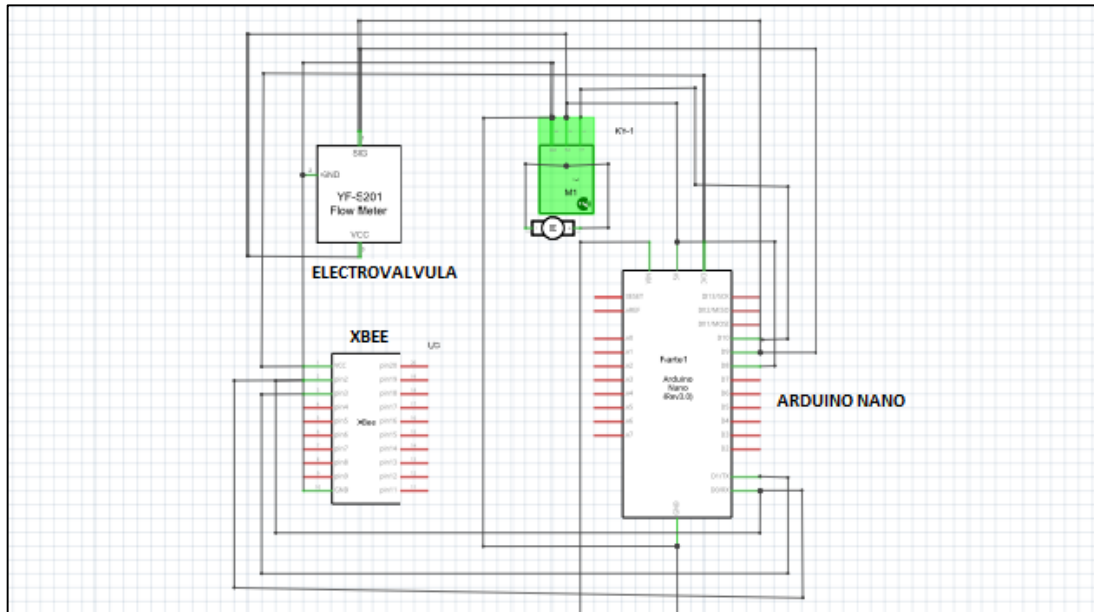
**Figura 24-2** Esquema de conexión del nodo sensor  
 Realizado por: PAREDES, Marjorie, 2017

### 2.4.2 Diagrama de conexión del nodo actuador

Trabaja con un arduino NANO como procesador, este es el encargado de interpretar la información recibida desde el nodo master, las conexiones del nodo actuador se muestran en la *Figura 25-2*, tiene conectado sus terminales de la siguiente manera:

- El módulo Xbee tiene conectada su salida TX al terminal D2 del arduino y la salida RX al terminal D3, la alimentación es de 3.3V y GND.
- El módulo medidor de flujo cuenta con tres salidas, dos de ellas alimentación de 5V y GND, tiene conectada su salida de datos al terminal D5 del arduino nano.
- El módulo relé tiene una alimentación de 5V y GND y su salida de datos está conectada al terminal D4 del arduino.



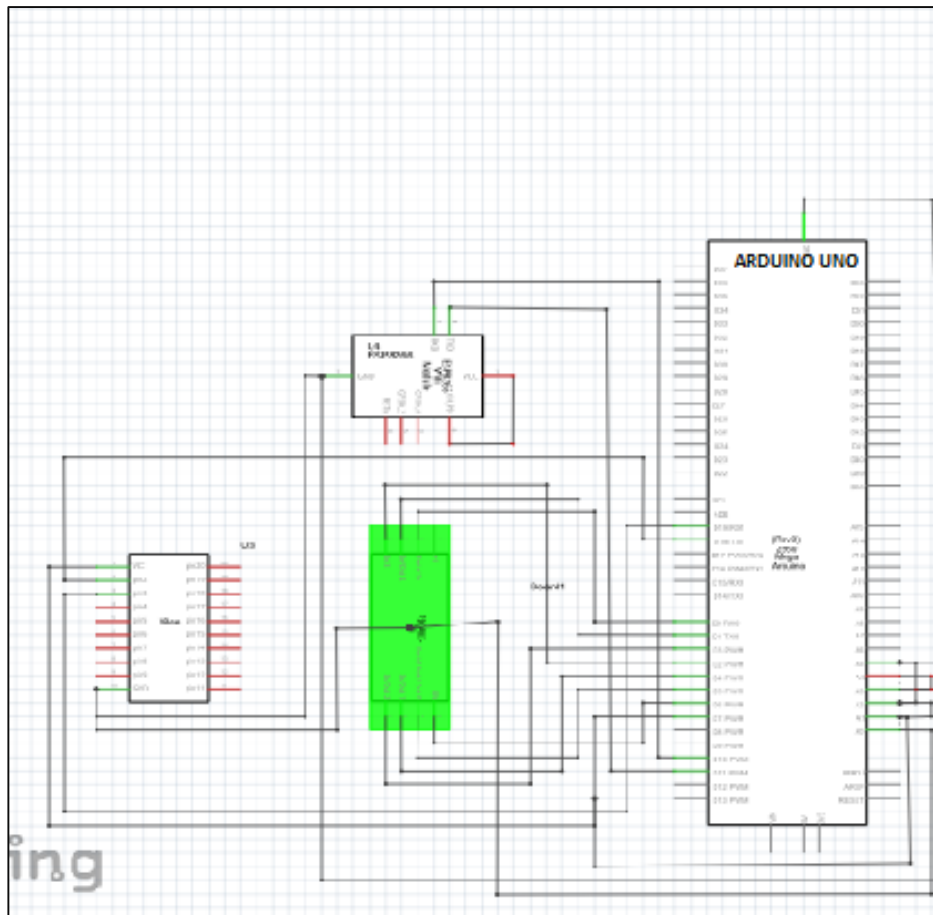


**Figura 25-2** Esquema de conexión del nodo actuador  
 Realizado por: PAREDES, Marjorie, 2017

### 2.4.3 Diagrama de conexión del nodo control

En la *Figura 26-2* se muestra el esquema del nodo control que cuenta con un arduino UNO para el procesamiento, contiene cuatro dispositivos: módulo transmisor y receptor XBee, módulo micro SD donde se almacenará la información, reloj del RTC y un módulo Bluetooth. Las conexiones de este nodo son los siguientes:

- Bluetooth tiene sus salidas de datos de TX y RX conectados a los terminales D0 y D1 del arduino UNO.
- La salida de TX de datos de la tarjeta Xbee está conectado al terminal D2 del arduino y el RX al terminal D3, su alimentación es de 3.3V y GND.
- El módulo micro SD tiene sus terminales conectados al arduino de la siguiente manera: D10 (CS), D11 (MOSI), D13 (SCK), D12 (MISO) y su alimentación de 5V y GND.
- El reloj permite la actualización de fecha y hora del módulo RTC, su salida de datos está conectado a los terminales de arduino de la siguiente manera SCL (D4) y SDA (D5).



**Figura 26-2** Esquema de conexión del nodo control  
 Realizado por: PAREDES, Marjorie, 2017

## 2.5 Requerimiento Software de la Red Inalámbrica de Sensores de Control Automático de Riego

En esta sección se detalla los requerimientos *software* del SICAR para el Centro Experimental del Riego de la Escuela Superior Politécnica de Chimborazo. Las necesidades de los diferentes nodos de detallan a continuación:

### 2.5.1 *Requerimientos software para el Nodo Sensor*

- Tomar las mediciones de los sensores en forma secuencial en tiempo real y convertirlas en señales digitales que van desde 0 a 1023 para luego convertirla en grados en el caso de la temperatura y porcentaje en la humedad.
- Programar para el cambio de identificación del nodo cuando el estado del micro switch externo cambie.

- Visualizar los datos de temperatura, humedad e identificación del nodo por medio del LCD.
- Enviar los valores de humedad y temperatura medidos por los sensores a la nube para su visualización en una página de dominio público utilizando la comunicación inalámbrica del módulo ESP8266 Wi-Fi.
- Enviar la cadena de información hacia el nodo control por medio de la comunicación ZegBee.
- Enviar alertas a través de mensajes de texto utilizando la red GSM, estas alertas serán enviadas cuando el nodo actuador notifique la apertura del riego, además se incluirá en el texto el identificador del nodo, la humedad y temperatura de ese instante.

### **2.5.2 *Requerimientos software para el Nodo Actuador***

- Recibir la trama de datos del nodo master de la lectura de los sensores de humedad y temperatura para activar la electroválvula y empezar el riego.
- Enviar al nodo master a través de la comunicación Zigbee el caudal de agua medido por el sensor de caudal.
- Enviar la información de la lectura del sensor de caudal a la página de *Thing Speak* a través de comunicación inalámbrica con la ayuda del módulo ESP8266 Wi-Fi.

### **2.5.3 *Requerimientos software para el Nodo Control***

- Receptar las tramas de datos que son enviadas del nodo sensor y actuador a través de la comunicación Zigbee.
- Almacenar la trama de datos en la memoria interna de la tarjeta de desarrollo para luego ser almacenada en un micro SD.

- Enviar la cadena de información hacia una aplicación disponible en un dispositivo móvil donde se visualizará los valores medidos por los sensores en tiempo real por medio de la comunicación Bluetooth.
- Suministrar información a la salida del módulo micro SD mediante un documento de texto donde costará la identificación del nodo, fecha, temperatura, humedad y caudal.

#### 2.5.4 *Software de desarrollo del sistema*

Maneja los diferentes nodos del SICAR se ejecutan por medio de funciones y programación estructurada. Para el desarrollo del SICAR se utilizaron tres herramientas que se detallan a continuación:

##### 2.5.4.1 *Herramienta Software XCTU*

*XBee Configuration and Test Utility* por sus siglas en inglés fue desarrollado por Digi y está disponible para Windows, MacOS y Linux es un *software* multiplataforma que permite la interacción con los módulos mediante un interfaz gráfico como se muestra en la que incluye herramientas utilizadas para configurar, inicializar, actualizar *firmware* y testear los módulos XBee (LIBELIUM, 2017). No requiere de programación, los módulos Xbee son conectados vía USB al computador donde se configuran los campos necesarios.

##### 2.5.4.2 *Herramienta Software Arduino IDE*

Fue diseñada con una interfaz amigable, está disponible para todos los sistemas operativos y es de código abierto, el entorno está escrito en Java, este software se usa en todas las placas disponibles de arduino. Contiene un editor de texto, un área de mensajes, una consola de texto, barra de herramientas y una serie de menús (ARDUINO, 2017). El lenguaje de programación utilizado por arduino es básicamente C y C++, los comandos utilizados por los programas de Arduino son implementados tomando referencia al lenguaje C++, conservando su sintaxis de declaración de variables, operadores, punteros y estructuras. (Pedrera, 2017 págs. 27,28). Entre sus opciones permite: seleccionar el tipo de tarjeta arduino, seleccionar librerías, compilar el *sketch* y descargar por USB además monitorear resultados. Un *sketch* está formado por dos funciones esenciales: *setup()* y *loop()*; la primera inicializa los requerimientos del sistema y solo

es ejecutada una vez, la segunda es una ejecución infinita es decir un bucle donde contiene todas las funciones y procesos del sistema. (GONZÁLEZ, 2015).

#### 2.5.4.3 *Herramienta Software ThingSpeak*

Es una plataforma abierta para el Internet de las Cosas que permite recopilar, almacenar, analizar, visualizar y actuar sobre la información recogida en sensores y dispositivos como aplicaciones móviles y redes sociales como twitter, es compatible con *hardware* de código abierto como Arduino, Raspberry Pi o BeagleBone, además con lenguajes computacionales como Matlab. Utiliza canales; los cuales se crean de acuerdo a la necesidad del usuario, estos canales contienen los campos de datos, ubicación y estado; en el canal creado se almacena la información enviada por los sensores donde posteriormente puede ser analizada y visualizada en gráficos, los datos obtenidos se almacenan en un archivo .xlsx que está disponible para descargar a un ordenador o dispositivo móvil (BBVA, 2016).

#### 2.5.4.4 *Herramienta Software App Inventor*

Creada por MIT (*Massachusetts Institute of Technology por sus siglas en inglés*) que posteriormente pasó a dominio de Google, es un entorno de desarrollo de aplicaciones para dispositivos Android, Es de desarrollo visual con facilidad de uso, para el desarrollo de aplicaciones se necesita un navegador web y un dispositivo Android (CÓDIGO21, 2016). En el SICAR se utiliza para crear una aplicación que se conecta con Bluetooth donde se visualiza la identificación del nodo, humedad, temperatura y caudal.

#### 2.5.5 *Programación de los nodos del sistema inalámbrico.*

Las líneas de código realizadas en el *software* de desarrollo IDE 1.8.2 para cada uno de los nodos se muestran en los Anexos 2, 3 y 4, se realiza con programación lineal por medio de funciones. A continuación se detalla las librerías, funciones, variables y diagramas de flujo de cada nodo

### 2.5.5.1 Diagrama de flujo del Nodo Sensor

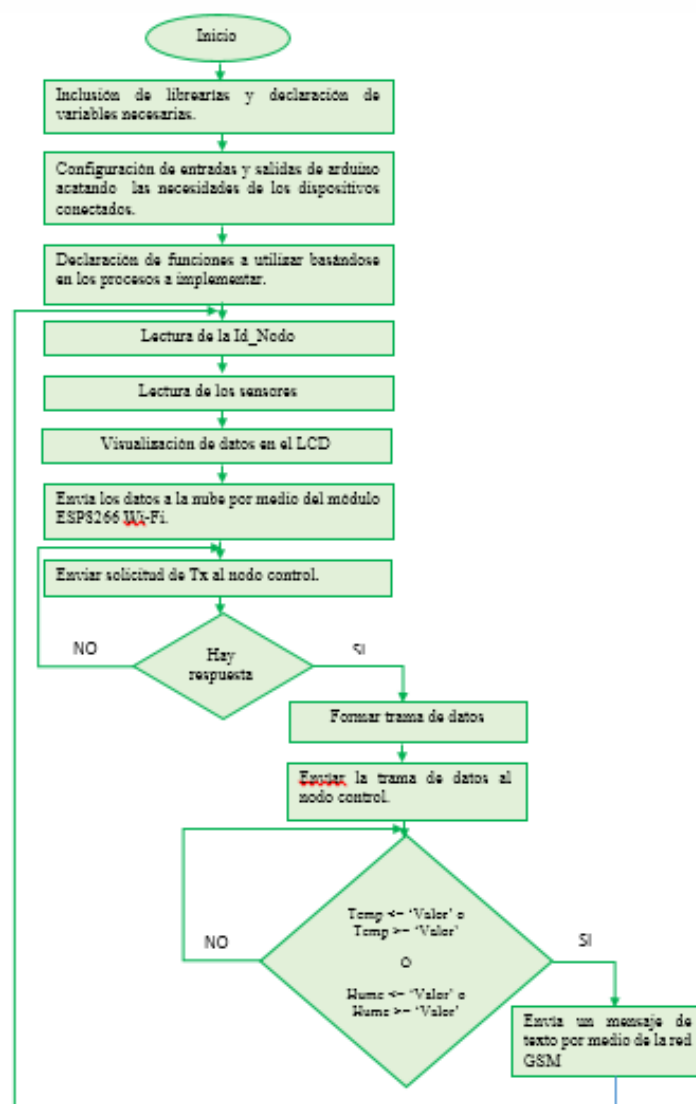
En la *Figura 27-2* se muestra el diagrama de flujo de la programación correspondiente al nodo sensor, el *sketch* de programación se realiza de la siguiente manera:

Declaración de variables e inicialización:

- Inclusión de librerías y declaración de variables necesarias.
- Configuración de entradas y salidas de arduino acatando las necesidades de los dispositivos conectados.
- Declaración de funciones a utilizar basándose en los procesos a implementar.
- Por medio de la función **TextosLCD()** se presenta mensajes en la pantalla gráfica del nodo.

Procesos repetitivos de manera indefinida:

- La identificación del nodo se obtiene con la función **obtener\_ID()**  
Al inicializar el nodo su número de identificador es 0 hasta que este sea ingresado manualmente con la ayuda del micro switch.
- Para la lectura de los sensores se utiliza la función **get\_tem()** en el caso del sensor de temperatura y **get\_hum()** para la humedad.
- Los valores de la lectura de sensores se visualizan en un lcd con la función **lcd.print()**.
- Posterior a la lectura de los sensores se crea una trama de datos que se transmite por ZigBee mediante la función **xbee.send()**.
- El módulo Xbee envía una solicitud de transmisión con la función **xbee.getResponse()**, a continuación busca el estado de entrega con la función **txStatus.getDeliveryStatus()**.



**Figura 27-2** Diagrama de flujo del programa para el Nodo Sensor  
Realizado por: PAREDES, Marjorie, 2017

### 2.5.5.2 Librerías usadas en la programación del Nodo Sensor

- **LiquidCrystal.h:** Permite a arduino controlar las pantallas LED mediante el uso de un *chipset* denominado Hitachi HD44780, esta librería funciona con 4 u 8 bits de velocidad (ARDUINO, 2017).
- **Xbee.h:** Esta librería permite la comunicación inalámbrica entre arduino y Zigbee en modo API, incluye un soporte para los diferentes paquetes: TX, RX, Remote AT y AT command entre otros.

- **SoftwareSerial.h:** Fue desarrollada con la finalidad de que exista comunicación en serie entre otros pines digitales de arduino, replicando su funcionalidad, estos puertos en serie pueden alcanzar una velocidad de hasta 115200 bps (ARDUINO, 2017).
- **DHT.h:** Permite el uso de los sensores de temperatura y humedad.

#### 2.5.5.3 *Funciones que están incluidas en el sketch del Nodo Sensor*

- **obtener\_ID():** Esta función es la encargada de agregar un número que sea el identificador del nodo.
- **get\_tem():** Esta función es la encargada de realizar la lectura de los valores enviados por el sensor de temperatura.
- **get\_hum():** Esta función es la encargada de realizar la lectura de los valores enviados por el sensor de humedad.
- **lcd.print():** Toma los valores de las lecturas de los sensores e identificación del nodo, almacena en una trama de datos y visualiza en una pantalla LCD de 16x2.
- **xbee.send():** Esta función permite enviar la trama de datos por medio de la tecnología Zigbee hacia los demás nodos.
- **xbee.getResponse():** Esta función es la encargada de enviar las solicitudes de transmisión a los otros nodos a través de Zigbee.
- **txStatus.getDeliveryStatus():** Después del envío de la solicitud se espera una notificación de aprobación, esta función se encarga de buscar el estado de entrega.

#### 2.5.5.4 *Diagrama de flujo del Nodo Actuador*

En la *Figura 28-2* se muestra el diagrama de flujo de la programación correspondiente al nodo actuador, el *sketch* de programación se realiza de la siguiente manera:

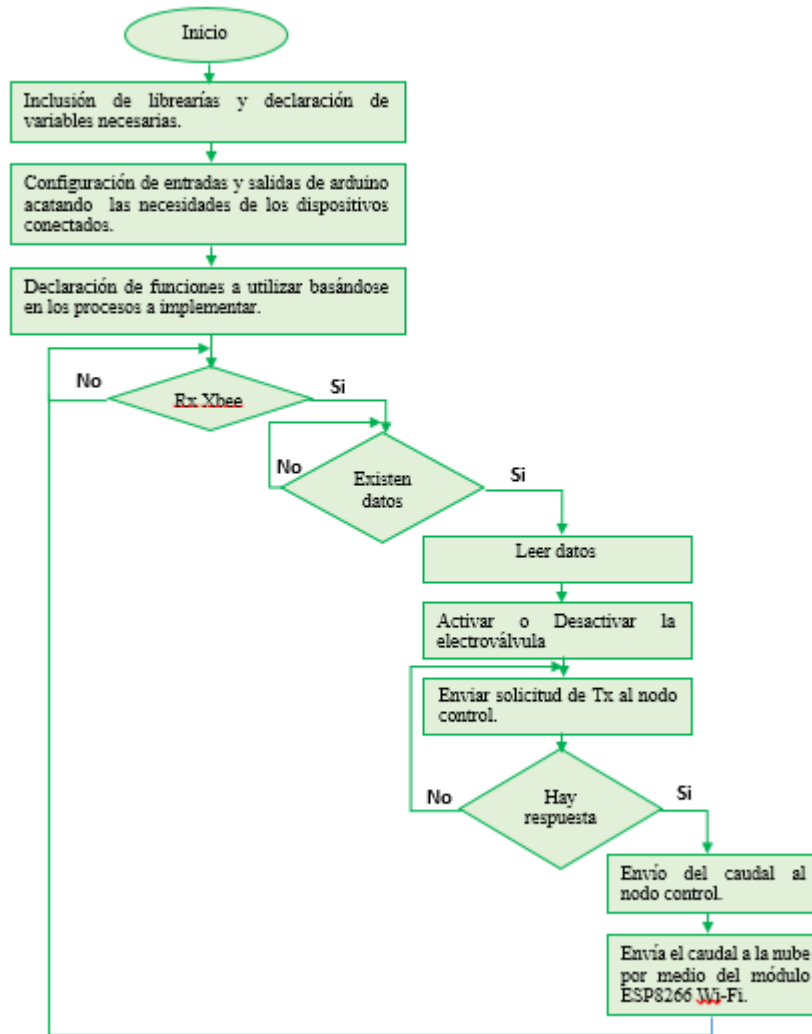


Declaración de variables e inicialización:

- Inclusión de librerías y declaración de variables necesarias.
- Configuración de entradas y salidas de arduino acatando las necesidades de los dispositivos conectados.
- Declaración de funciones a utilizar basándose en los procesos a implementar.

Procedimientos repetitivos de manera indefinida:

- Determinar la identificación del nodo que está enviando datos con la función **id\_nodo()**.
- Recibir datos (lectura de sensores) desde el nodo Control por medio de Zigbee.
- Leer la información recibida con la función **xbee.readResponse()**.
- Encender o apagar la electroválvula con la función **digital.Write(variable,HIGH/LOW)**.
- Medir el caudal de agua con la función **medir\_caudal()**.
- Incorporar a la trama de datos recibidos al caudal con la función **generacion\_trama()**.
- Enviamos la trama hacia el nodo control y hacia la nube con la función **xbee.send()** y comprobamos el envío con la función **comprobar\_envio()**



**Figura 28-2** Diagrama de flujo del programa para el Nodo Actuador  
Realizado por: PAREDES, Marjorie, 2017

#### 2.5.5.5 Librerías usadas en la programación del Nodo Actuador

Para el desarrollo de la programación de este nodo se utilizan las librerías **Xbee.h** y **SoftwareSerial.h** las cuales ya fueron descritas en la descripción del nodo sensor.

#### 2.5.5.6 Funciones que están incluidas en el sketch del Nodo Actuador

- **Id\_nodo()**: Esta función es la encargada de asignar un número de identificación de nodo.
- **Xbee.readResponse()**: Descrita en el nodo anterior.

- **digital.Write(variable,HIGH/LOW):** Es la función que genera la orden de encendido o apagado de la electroválvula.
- **medir\_caudal():** Es la encargada de realizar la medición del caudal de agua, calcula cada segundo; para realizar este cálculo acumula en una variable la cantidad de pulsos en minutos para después transformar el resultado a litros.
- **generacion\_trama():** Esta función tiene como propósito la creación de una nueva trama donde además de los valores de humedad y temperatura se añadirá el caudal de agua.
- **xbee.send():** Envía la nueva trama de datos creada hacia el nodo control por medio de la comunicación inalámbrica zigbee.
- **comprobar\_envio():** La función descrita es la encargada de enviar la solicitud de transmisión hacia el nodo control.

#### 2.5.5.7 Diagrama de flujo del Nodo Control

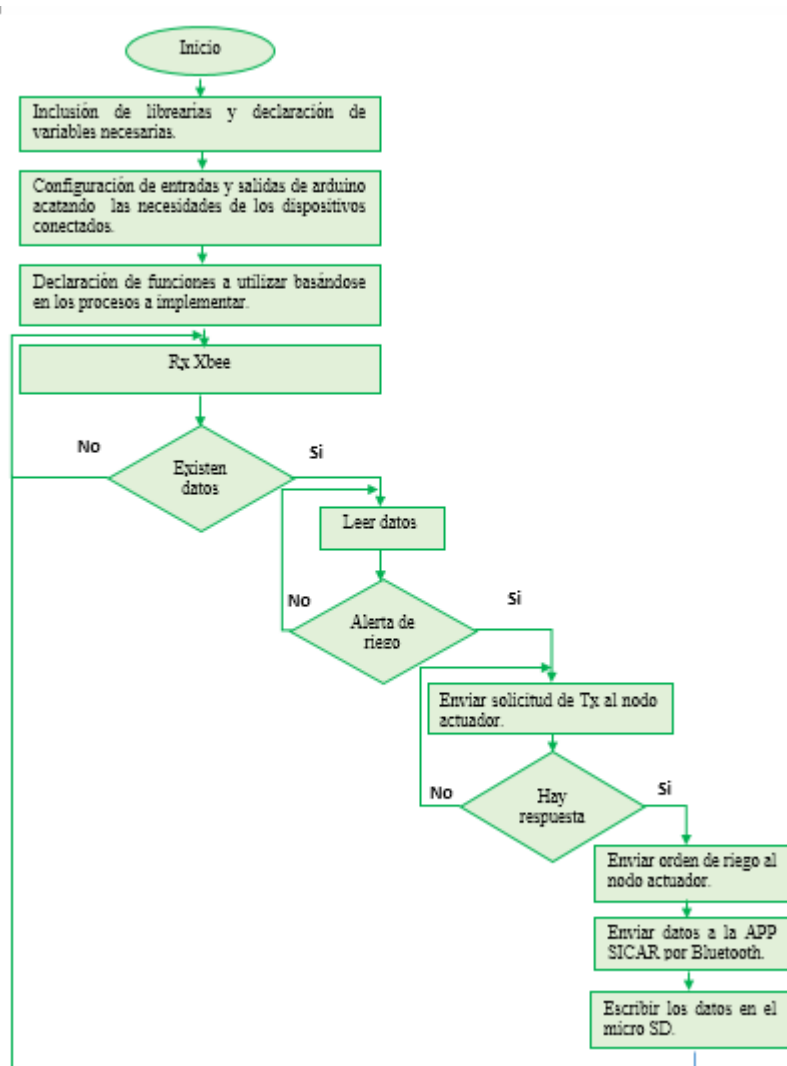
En la *Figura 29-2* se muestra el diagrama de flujo de la programación correspondiente al nodo control, el *sketch* de programación se realiza de la siguiente manera:

Declaración de variables e inicialización:

- Inclusión de librerías y declaración de variables necesarias.
- Configuración de entradas y salidas de arduino acatando las necesidades de los dispositivos conectados.
- Declaración de funciones a utilizar basándose en los procesos a implementar.
- Habilitar el puerto serial **serial.begin(9600)**.

Procesos repetitivos de manera indefinida:

- Crear un objeto Xbee con la función **Xbee()**.
- Lectura de tramas recibidas del nodo sensor y actuador se realiza con la función **xbee.readPacket()**.
- Crear una dirección de 64 bits para la trama que crea la electroválvula con la función **XbeeAddress64()**.
- Enviar la orden de encendido de la electroválvula se utiliza la función **encender\_electrovalvula()** y para el apagado se utiliza la función **apagar\_electrovalvula()**.
- Imprimir los datos en la aplicación móvil a través de la comunicación inalámbrica Bluetooth con la función **serial.print()**.
- Habilita los terminales digitales de arduino con la función **SD.begin()**.
- Abre un archivo en la tarjeta micro SD con la función **SD.open()**, si el archivo se abre se escribe el nuevo archivo creado.



**Figura 29-2** Diagrama de flujo del programa para el Nodo Control  
Realizado por: PAREDES, Marjorie, 2017

#### 2.5.5.8 Librerías usadas en la programación del Nodo Control

- **Xbee.h:** Descrita en el nodo anterior.
- **SoftwareSerial.h:** Descrita en el nodo anterior.
- **SD.h:** Permite leer y escribir en tarjetas SD, esta librería admite sistemas de archivos FAT16 y FAT32 en tarjetas SD estándar y tarjetas SDHC, utiliza nombres cortos para guardar los archivos, los nombres de los archivos pueden incluir rutas separadas por barras inclinadas: Directorio/NombredelArchivo.txt (ARDUINO, 2017).

- **Wire.h:** Esta librería permite la comunicación con dispositivos 12C/TWI. En las placas arduino con la disposición R3 (1.0 pin out), la SDA (línea de datos) y SCL (línea de reloj) están en los encabezados de los pines cerca del pin AREF (ARDUINO, 2017).
- **SPI.h:** Es un protocolo de datos en serie síncrono que utilizan los microcontroladores para una comunicación rápida y de cortas distancias de uno o más dispositivos periféricos. Permite comunicarse con dispositivos SPI con arduino como dispositivo maestro, con la comunicación SPI siempre hay un dispositivo maestro, generalmente hay tres líneas comunes en estos dispositivos (ARDUINO, 2017):
  - **MISO** (Master In Slave Out): Envía datos al maestro.
  - **MOSI** (Master Out Slave In): Línea del maestro para enviar datos a los periféricos.
  - **SCK** (Reloj serie): Sincronización del reloj para la transmisión de datos generada por el maestro.
- **RTClib.h:** Esta librería habilita a un arduino con arquitectura SAMD para tener el control y usar el RTC (*Real Time Clock por sus siglas en inglés*) interno. (ARDUINO, 2017)

#### 2.5.5.9 Funciones que están incluidas en el sketch del Nodo Sensor

**Serial.begin(9600):** Esta función es la encargada de habilitar el puerto serial para el envío de datos.

**Xbee():** Es la encargada de la creación de objetos xbee.

**Xbee.readPacket():** Esta función realiza la lectura de las tramas de datos recibidas desde los nodos sensor y actuador.

**XbeeAddress64():** Esta función crea una dirección de 64 bits para la trama creada por el nodo actuador.

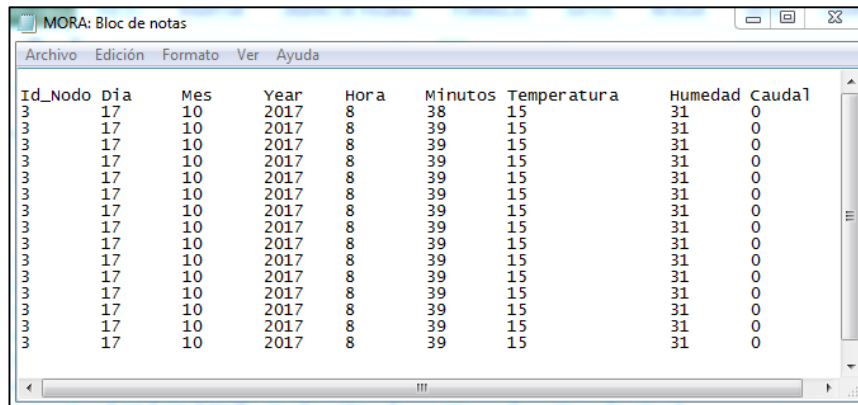
**encender\_electroválvula():** Cumple con la función de ejecutar una orden de encendido de la electroválvula.

**apagar\_electrovalvula():** Esta función es la encargada de enviar la orden de apagado de la electroválvula.

**Serial.print():** Esta función imprime los datos del terminal serial de arduino en la aplicación móvil.

**SD.begin():** Inicializa la tarjeta y librería SD.

**SD.open():** Abre el archivo en la tarjeta micro SD y permite la escritura como se muestra la *Figura 30-2*.



Id_Nodo	Dia	Mes	Year	Hora	Minutos	Temperatura	Humedad	Caudal
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0
3	17	10	2017	8	39	15	31	0

**Figura 30-2** Datos almacenados en un archivo  
Realizado por: PAREDES, Marjorie, 2017

## 2.5.6 Configuración de los módulos XBee

Estos módulos son configurados en el *software* XCTU mediante un cable USB en el computador.

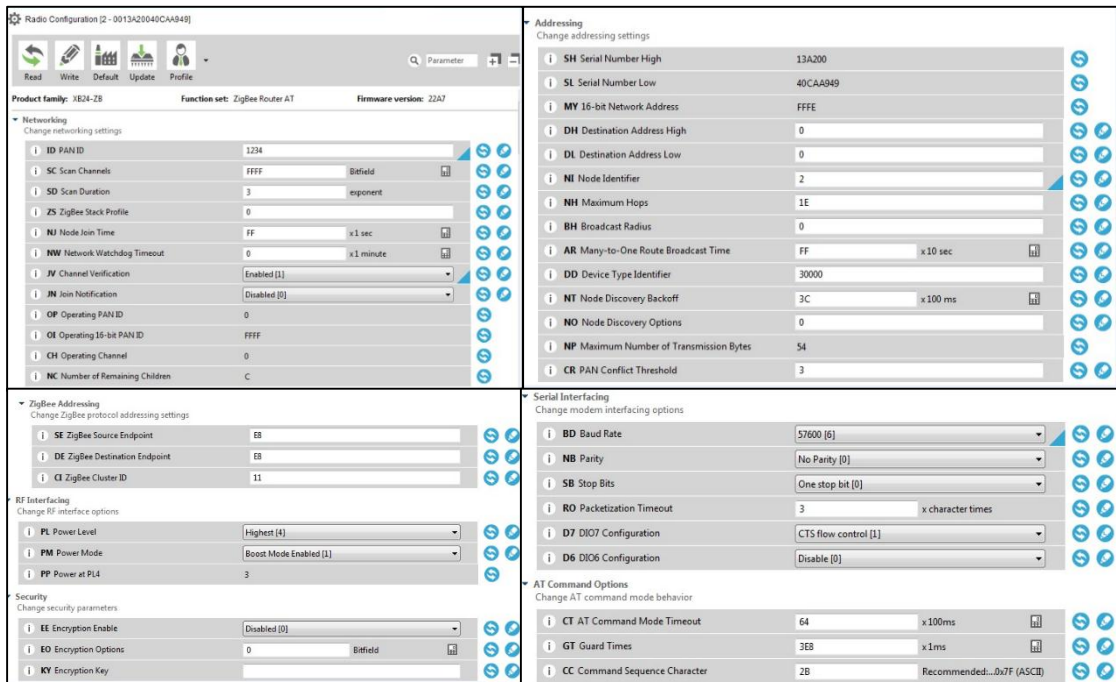
### 2.5.6.1 Configuración del módulo Control

Los dispositivos XBee se comunican de manera inalámbrica, para que se realice la transmisión los módulos debes estar en la misma frecuencia y en la misma red, el módulo es el encargado de crear la red, para esto es necesario tener en cuenta dos parámetros:

**Canal (CH):** es el canal usado para generar la red, en este sistema se usa el canal B.

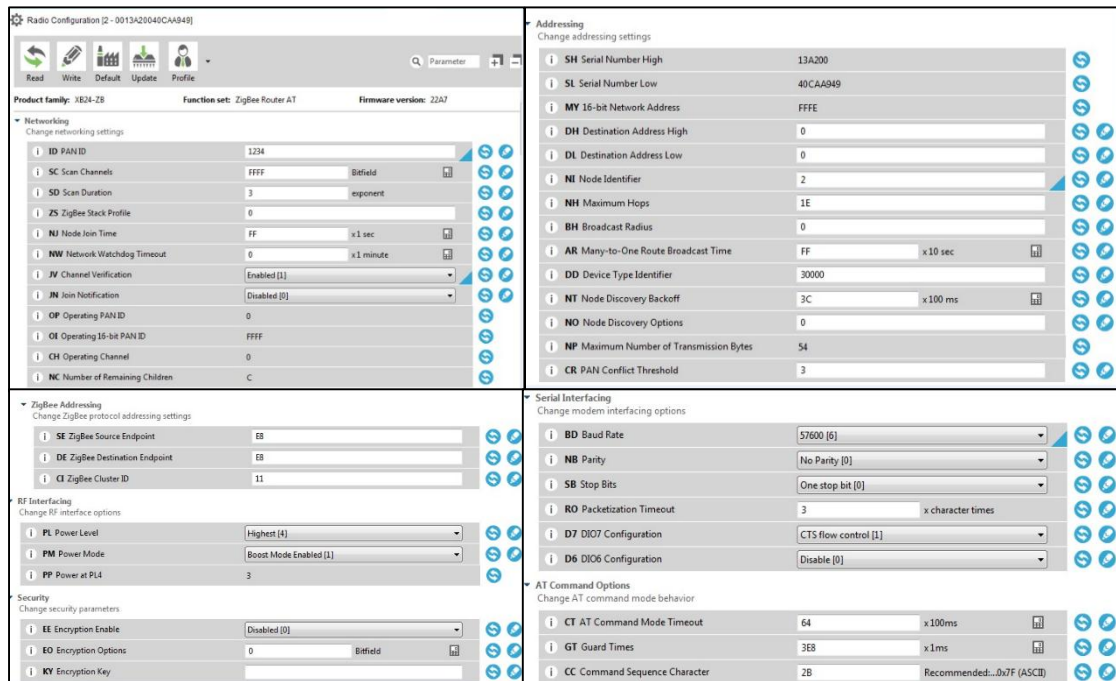
**Identificador (ID):** Es un número único que establece que los módulos están en la misma red, para esta red se utiliza el id: 1234

Un módulo Xbee solo transmite y recibe datos cuando tiene el mismo identificador y usa el mismo canal, la configuración del módulo control se muestra en la *Figura 31-2*, la del módulo actuador en la *Figura 32-2* y el módulo sensor en la *Figura 33-2*.



**Figura 31-2** Configuración del módulo control  
Realizado por: PAREDES, Marjorie, 2017

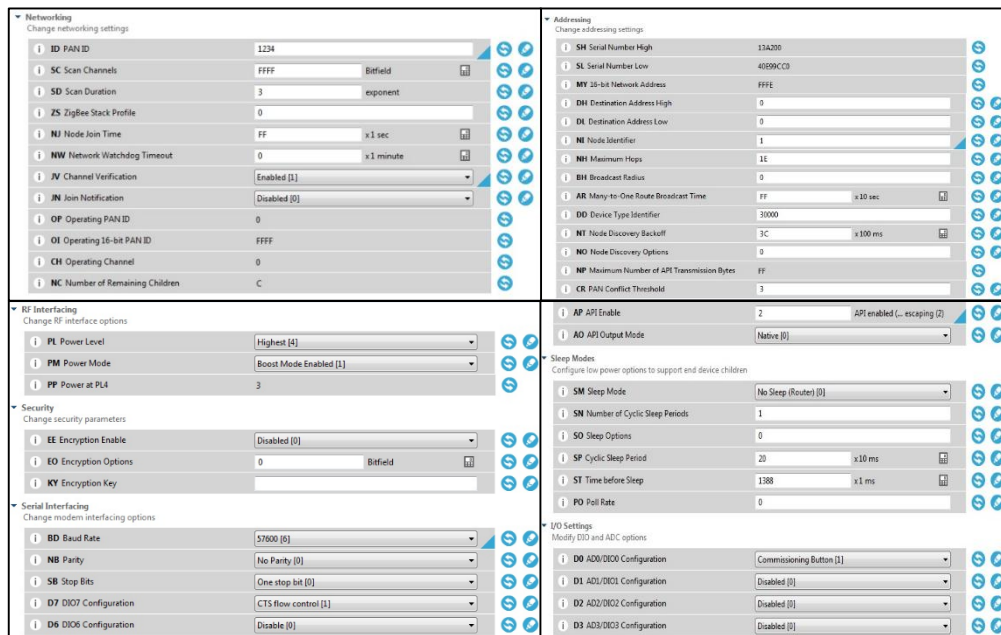
## Configuración del módulo Actuador



**Figura 32-2** Configuración del módulo actuador  
Realizado por: PAREDES, Marjorie, 2017



## Configuración del módulo Sensor



**Figura 33-2** Configuración del módulo sensor  
Realizado por: PAREDES, Marjorie, 2017

### 2.5.7 Software para la presentación y visualización de información del SICAR

SICAR cuenta con tres herramientas de visualización gráfica de datos, dos de ellas son plataformas web y una aplicación móvil.

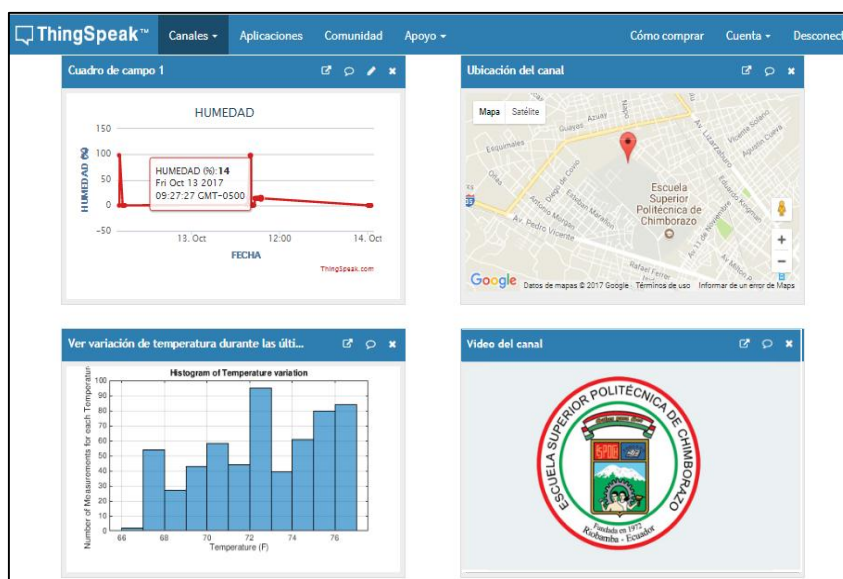
#### 2.5.7.1 Visualización en ThingSpeak

Esta herramienta de *software* libre permite crear canales, para el monitoreo del SICAR se crearon tres canales: **HUMEDAD**, **TEMPERATURA** y **CAUDAL DE AGUA** como se muestra en la *Figura 34-2*. *ThingSpeak* genera una ID del canal el cual es asignado a los módulos wifi EPS8266 para tener una comunicación única, para acceder a la información es necesario un usuario y contraseña.

Nombre	Creado	Actualizado en
🔒 HUMEDAD Private Public Configuración de uso compartido de API Keys Importar / Exportar datos	30-08-2017	2017-10-15 19:42
🔒 TEMPERATURA Private Public Configuración de uso compartido de API Keys Importar / Exportar datos	30-08-2017	2017-10-14 05:30
🔒 CAUDAL DE AGUA Private Public Configuración de uso compartido de API Keys Importar / Exportar datos	30-08-2017	2017-10-14 05:05

**Figura 34-2** Canales creados para el SICAR  
 Realizado por: PAREDES, Marjorie, 2017

Esta plataforma recibe la trama de datos enviados por el módulo ESP8266 y visualiza de manera gráfica en el canal correspondiente, en cada canal se puede agregar visualizaciones donde se incluye la gráfica de los datos enviados de cada nodo, además tiene la opción de incluir la ubicación exacta de la parcela con la ayuda de *Google maps*, también se grafican datos de comparaciones entre las variables de cada canal como se muestra en la *Figura 35-2*.



**Figura 35-2** Ventanas de visualización de datos en *ThingSpeak*  
 Realizado por: PAREDES, Marjorie, 2017

Esta plataforma da la opción de exportar los datos almacenados en cada canal, los datos están disponibles en formato csv que se visualizan en Microsoft Excel 2013.

### 2.5.7.2 Visualización en HTML

SICAR además de la plataforma de código abierto detallada anteriormente cuenta con una privada desarrollada en HTML (*HyperText Markup Language por sus siglas en inglés*) como se muestra en la *Figura 36-2* donde se visualiza los datos: caudal, temperatura y humedad. Cuenta con la opción de almacenar y exportar.



**Figura 36-2** Plataforma privada diseñada en HTML  
Realizado por: PAREDES, Marjorie, 2017

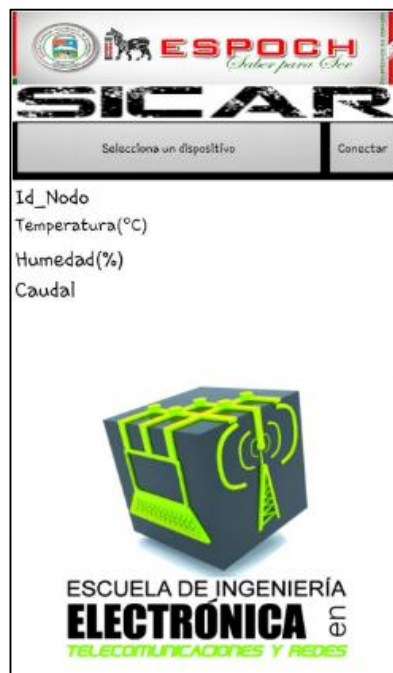
### 2.5.7.3 Visualización en App Inventor

SICAR cuenta con una aplicación móvil, su interfaz gráfica se muestra en la *Figura 37-2*. Fue desarrollada por la herramienta *App Inventor* y está disponible para todos los móviles con sistema operativo Android. La aplicación cuenta con dos botones:

**Selecciona un dispositivo:** Este botón selecciona el dispositivo *Bluetooth* instalado en el nodo control del SICAR.

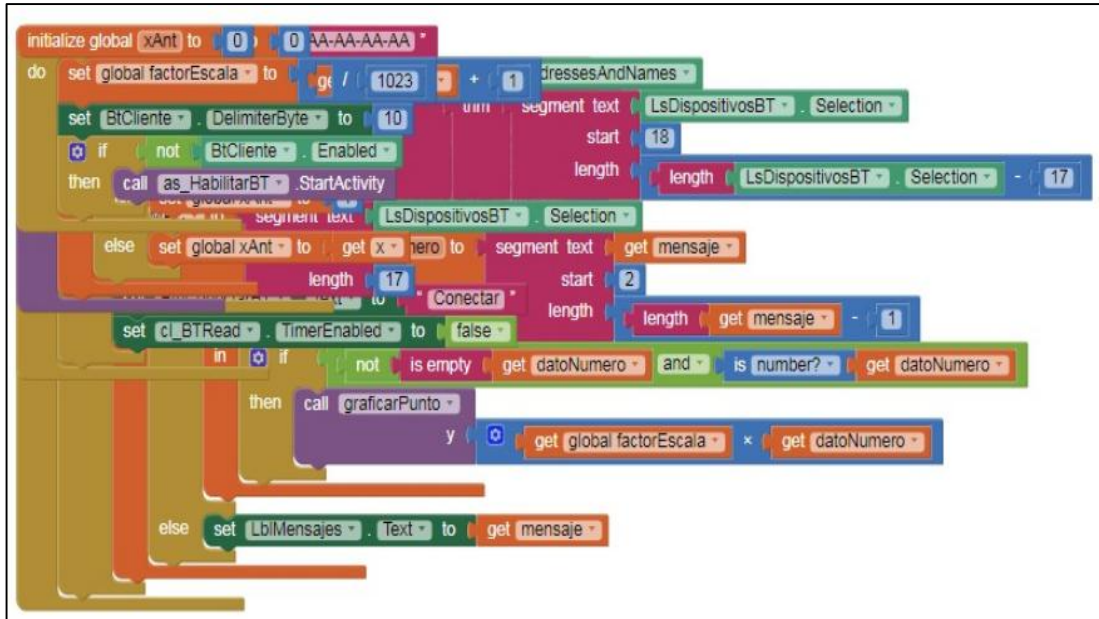
**Conectar:** Con este botón establecemos la conexión móvil - nodo control.

Si la conexión es realizada con éxito el nodo control envía las lecturas del nodo sensor para que sean mostradas en tiempo real en la pantalla de la app en el siguiente orden: Id\_Nodo, Temperatura, Humedad y Caudal.



**Figura 37-2** Interfaz de la aplicación en App Inventor.  
**Realizado por:** PAREDES, Marjorie, 2017

SICAR cuenta con esta aplicación como herramienta adicional a las plataformas mencionadas anteriormente, el usuario tendrá que conectarse con su dispositivo móvil y podrá visualizar la lectura de los sensores en tiempo real y así saber el estado de la plantación. El código del desarrollo de la aplicación se muestra en la *Figura 38-2*.



**Figura 38-2** Código de aplicación desarrollada en App Inventor  
 Realizado por: PAREDES, Marjorie, 2017

## CAPÍTULO III

### 3 ANÁLISIS Y RESULTADOS

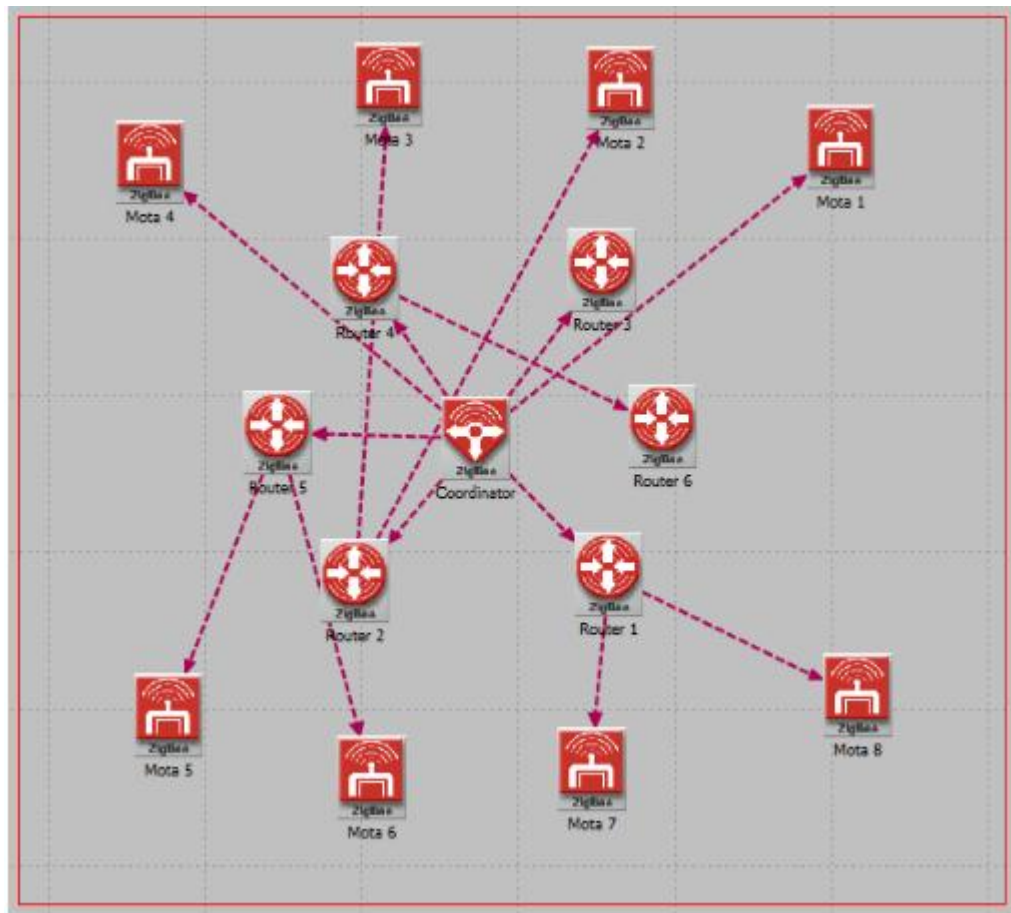
El presente capítulo se dedica al proceso de validación del sistema y resultados de la simulación de topologías de red, además resultados adquiridos en las pruebas realizadas al prototipo SICAR. La evaluación a cada uno de los nodos que conforman SICAR se ejecutó integrando el *software* y *hardware* diseñado. Tomando en cuenta los requerimientos del proyecto se realizaron pruebas de funcionamiento comparando resultados con equipos de mediciones similares. Se incluye un análisis de costos de los nodos que conforman SICAR. Las pruebas fueron ejecutadas en los predios de la Escuela Superior Politécnica de Chimborazo en la Facultad de Recursos Naturales conjuntamente con el Centro Experimental de Riego.

#### 3.1 Resultados de simulación en Riverbed Modeler

Para la elección de topología de red adecuada para el sistema se diseñó y evaluó dos topologías disponibles para la comunicación inalámbrica Zigbee como son topología de árbol y topología de malla. La simulación se llevó a cabo con un valor de 7 motas y un coordinador.

##### 3.1.1 *Análisis de la topología de red de árbol*

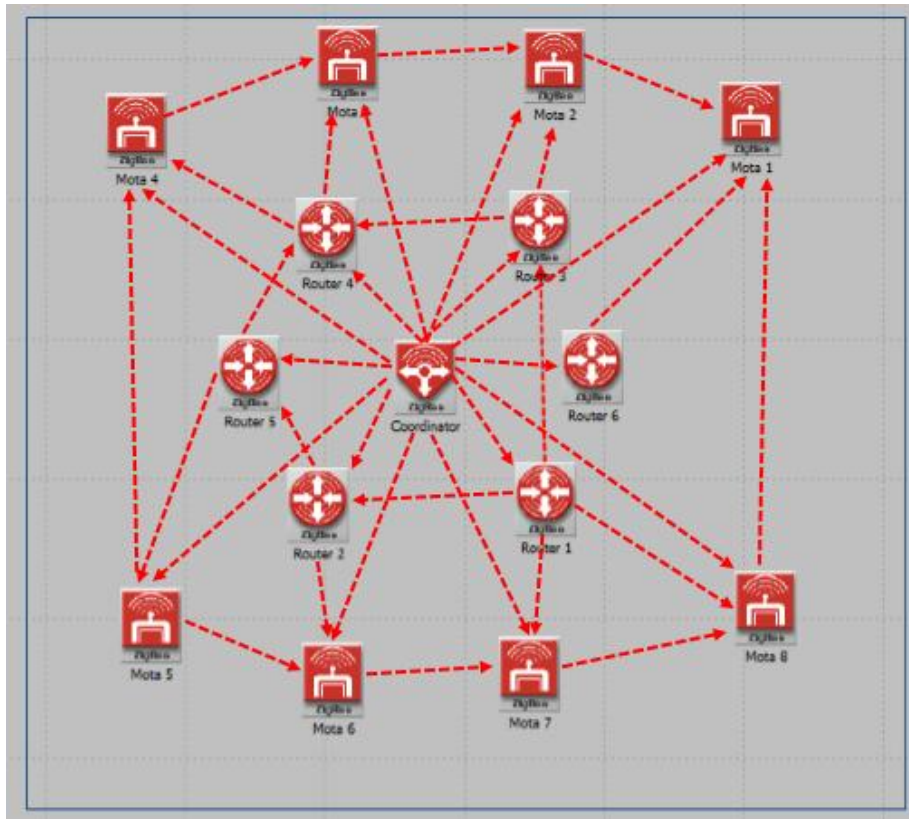
En la *Figura 1-3* se aprecia la simulación donde el coordinador se comunica con los routers y estos con las motas, esta secuencia de que la mota dependa del router y no haya comunicación directa entre coordinador y mota hace que haya retrasos en envío y recepción de paquetes.



**Figura 1-3** Simulación de la topología de red de árbol en Riverbed Modeler  
 Realizado por: PAREDES, Marjorie, 2017

### 3.1.2 *Análisis de la topología de red de malla*

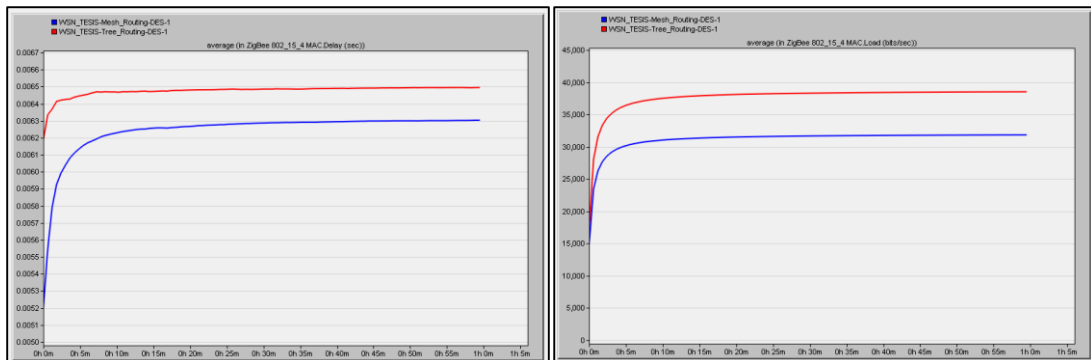
Esta topología tiene conexión directa entre coordinador - router y coordinador – mota, haciendo una comunicación directa sin dependencia de ningún router como se muestra en la *Figura 2-3*.



**Figura 2-3** Simulación de la topología de red de malla en Riverbed Modeler  
 Realizado por: PAREDES, Marjorie, 2017

### 3.1.3 Comparación de topologías

En la *Figura 3-3* se muestra la carga y retrasos promedios de las topologías realizadas en la simulación donde se aprecia que la topología de red de malla tiene un retraso de 0.0052 segundos y la topología de árbol un retraso de 0.0063 segundos, además la topología de malla tiene una velocidad de carga de 25 Kbps. Después de este análisis se escoge a la topología de red de malla para la ejecución del SICAR, siendo esta la topología que cumple con las necesidades del sistema a ser implementado.



**Figura 3-3** Retraso promedio de datos (izquierda) y carga promedio de datos (derecha)  
 Realizado por: PAREDES, Marjorie, 2017



### 3.2 Pruebas del nodo sensor

Para el correcto funcionamiento del nodo se realizaron pruebas de lectura de sensores, alcance, comunicación y visualización.

#### 3.2.1 Pruebas de mediciones realizadas por el nodo sensor

En la *Figura 4-3* se muestra el prototipo ensamblado del nodo sensor del SICAR.



**Figura 4-3** Prototipo del Nodo Sensor (izquierda), alimentación solar del nodo (derecha)  
Realizado por: PAREDES, Marjorie, 2017

#### Equipos utilizados para las mediciones

**Tensiómetro:** Es un equipo de mediciones de humedad del suelo, este equipo tiene un margen de error de en la primera hora de lectura, al entrar en estabilidad se comparó con el sensor de humedad implementado en el nodo sensor del SICAR. En la *Figura 5-3* se aprecia las lecturas de los dos equipos visualizando una humedad de 19 centibares.



**Figura 5-3** Tensiómetro (izquierda), nodo sensor (derecha)  
Realizado por: PAREDES, Marjorie, 2017

**Estación meteorológica:** El CER cuenta con este equipo como se muestra en la *Figura 6-3* donde se visualizan variables ambientales como la temperatura, velocidad del viento, humedad ambiental entre otras, se toma a la variable temperatura para la toma de muestras para el estudio de margen de error.



**Figura 6-3** Estación meteorológica CER  
Realizado por: PAREDES, Marjorie, 2017

En la tabla *Tabla 3-1* se muestran los resultados de la comparación de los equipos, las muestras fueron tomadas en un lapso de 7 días, de la información recolectada se toma una muestra para el análisis respectivo, la finalidad de la comparación de resultados de los equipos que pone a disposición el CER y SICAR es ver el margen de error del nodo sensor y así determinar si satisface los requerimientos establecidos. Del análisis se determina que el sensor de humedad incorpora un error  $\pm 2$  centibares y el de temperatura incorpora un error  $\pm 1\%$ , con estos resultados se determina que satisface los requerimientos del diseño previsto.

**Tabla 1-3** Mediciones realizadas con el sensor FC-28 del SICAR y tensiómetro y sensor DHT-11 con la central meteorológica del CER.

Número de Muestra	HUMEDAD		Error absoluto de la humedad	TEMPERATURA		Error absoluto de la temperatura
	Tensiómetro	Sensor FC-28		Sensor DHT-11	Central Meteorológica CER	
1	22	20	2	19	19	0
2	27	28	-1	20	20	0
3	34	34	0	19	18	1
4	37	35	2	18	18	0
5	36	36	0	17	17	0
6	35	35	0	20	21	-1
7	50	49	1	16	16	0
8	53	55	-2	15	15	0
9	49	50	-1	18	18	0
10	57	59	-2	19	19	0
11	55	56	-1	19	19	0
12	49	47	2	19	19	0
13	40	41	-1	21	21	0
14	46	45	1	18	17	-1
15	37	38	-1	14	14	0
16	57	57	0	16	16	0
17	39	37	2	19	19	0
18	60	62	-2	18	17	1
19	27	27	0	21	22	-1

Realizado por: PAREDES, Marjorie, 2017

### 3.2.2 Repetitividad del Nodo Sensor

Es la capacidad de un dispositivo sensor de reproducir una lectura con una precisión dada. (Maria, 2015). Este análisis se basa en la toma de muestras durante un minuto, los datos obtenidos se almacenan en la tarjeta micro SD y con la ayuda de Microsoft Excel 2013 se calcula la media, la desviación estándar y el coeficiente de variación de los datos.

En la *Tabla 3-2* se muestra las mediciones de los sensores donde se recalca que las lecturas del sensor de temperatura fueron constantes teniendo un coeficiente de variación de 0 %, a diferencia del sensor de humedad que tuvo valores variantes dando como resultado un coeficiente de variación de 1 %, con estos resultados se determina que el SICAR es un sistema estable.

**Tabla 2-3** Repetitividad de los valores leídos por los sensores del Nodo Sensor

<b>Número de Muestra</b>	<b>Hora</b>	<b>Valor Actual de la Humedad</b>	<b>Valor actual de la Temperatura</b>
1	16:44:28	70	28
2	16:44:30	70	28
3	16:44:32	70	28
4	16:44:34	70	28
5	16:44:36	71	28
6	16:44:40	70	28
7	16:44:44	70	28
8	16:44:48	71	28
9	16:44:50	71	28
10	16:44:52	70	28
11	16:44:53	70	28
12	16:44:54	70	28
13	16:44:55	70	28
14	16:44:57	71	28
15	16:44:59	70	28
16	16:45:02	71	28
17	16:45:05	70	28
18	16:45:07	70	28
19	16:45:09	70	28
20	16:45:10	70	28
<b>Media</b>		<b>70,28</b>	<b>28,00</b>
<b>Desviación Estándar(<math>\sigma</math>)</b>		<b>0,64</b>	<b>0,00</b>
<b>Coefficiente de Variación</b>		<b>1%</b>	<b>0,0%</b>

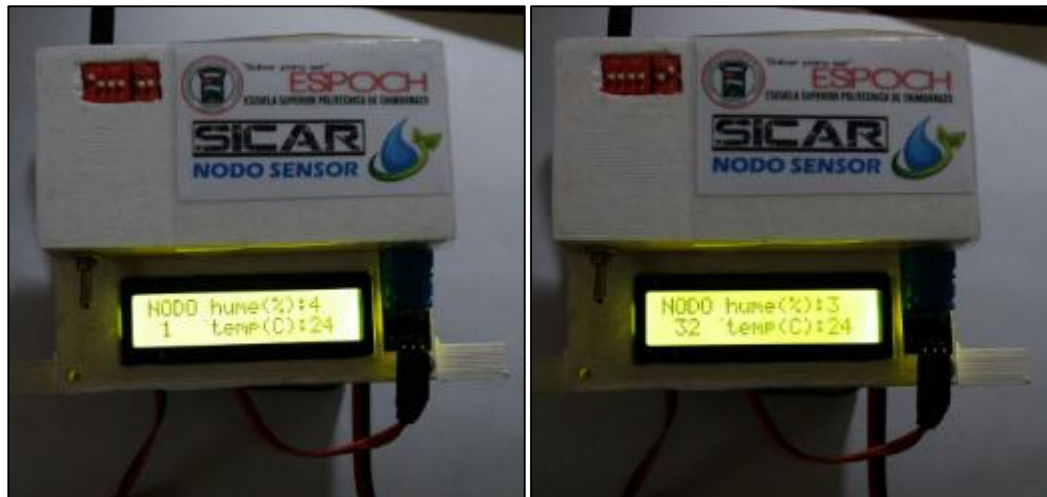
Realizado por: PAREDES, Marjorie, 2017

### 3.2.3 Características de las comunicaciones entre dispositivos del Nodo Sensor

Para la comunicación entre dispositivos que componen el nodo sensor se realiza una evaluación, por lo que se lleva a cabo pruebas de comunicación.

### Comunicación entre la tarjeta Arduino y micro switch

Consiste en ingresar manualmente la identificación del nodo con la combinación binaria del micro switch, en la *Figura 7-3* se muestra dicha combinación y visualización. La combinación del micro switch da la opción de ingresar hasta 64 identificadores.

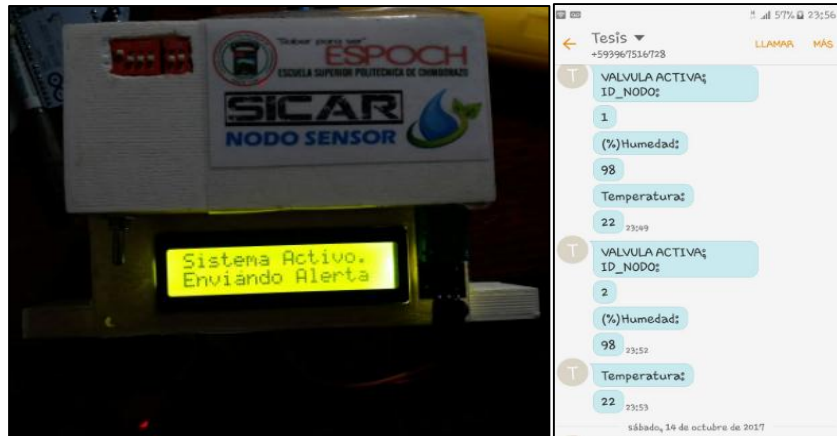


**Figura 7-3** Ingreso del identificador del nodo, combinación del micro switch para 1 (izquierda), y combinación del micro switch para 32 (derecha).

Realizado por: PAREDES, Marjorie, 2017

### Comunicación entre tarjeta Arduino y módulo SIM900 – GSM

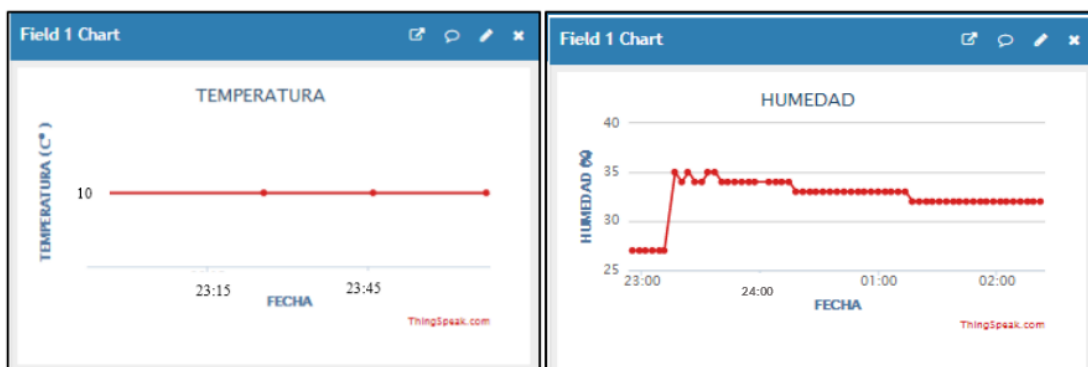
El nodo sensor además tiene incorporado entre sus dispositivos un módulo SIM900 que es el encargado de enviar notificaciones a través de la red GSM cuando empieza el riego como se muestra en la *Figura 8-3*, en este mensaje incluye el aviso de la válvula activa, el identificador del nodo, la humedad y temperatura del momento en el cual es enviada dicha notificación. Esta notificación llega al dispositivo móvil después de 8 segundos que la tarjeta arduino envía la orden.



**Figura 8-3** Envío de notificaciones a través de la red GSM (izquierda) y recepción de mensaje de texto en el móvil (derecha).  
**Realizado por:** PAREDES, Marjorie, 2017

### Comunicación entre tarjeta Arduino y módulo ESP8266

El nodo sensor cuenta también con la comunicación inalámbrica del módulo ESP8266, este dispositivo es el encargado de enviar las lecturas de los sensores de humedad y temperatura a la nube a la plataforma libre *ThingSpeak* donde la información se almacena y se muestra en un gráfico dependiendo de los datos que lleguen como se muestra en la *Figura 9-3*.



**Figura 9-3** Visualización de datos de temperatura (izquierda) y humedad (derecha) en la plataforma ThingSpeak.  
**Realizado por:** PAREDES, Marjorie, 2017

Esta información almacenada es exportada a Microsoft Excel 2013 como se observa en la *Figura 10-3*.

TEMPERATURA - Excel																																																																																																																																															
ARCHIVO	INICIO	INSERTAR	DISEÑO DE PÁGINA		FÓRMULAS																																																																																																																																										
<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> </tr> </thead> <tbody> <tr><td>1</td><td>Día</td><td>Mes</td><td>Year</td><td>Hora</td><td>Minutos</td><td>Segundos</td><td>Temperatura</td></tr> <tr><td>2</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>10</td><td>14</td><td>10</td></tr> <tr><td>3</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>12</td><td>10</td><td>10</td></tr> <tr><td>4</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>15</td><td>60</td><td>10</td></tr> <tr><td>5</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>20</td><td>39</td><td>10</td></tr> <tr><td>6</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>23</td><td>42</td><td>10</td></tr> <tr><td>7</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>27</td><td>17</td><td>10</td></tr> <tr><td>8</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>30</td><td>54</td><td>10</td></tr> <tr><td>9</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>32</td><td>38</td><td>10</td></tr> <tr><td>10</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>34</td><td>29</td><td>10</td></tr> <tr><td>11</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>36</td><td>51</td><td>10</td></tr> <tr><td>12</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>38</td><td>15</td><td>10</td></tr> <tr><td>13</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>40</td><td>19</td><td>10</td></tr> <tr><td>14</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>42</td><td>33</td><td>10</td></tr> <tr><td>15</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>45</td><td>50</td><td>10</td></tr> <tr><td>16</td><td>17</td><td>10</td><td>2017</td><td>23</td><td>47</td><td>28</td><td>10</td></tr> </tbody> </table>									A	B	C	D	E	F	G	1	Día	Mes	Year	Hora	Minutos	Segundos	Temperatura	2	17	10	2017	23	10	14	10	3	17	10	2017	23	12	10	10	4	17	10	2017	23	15	60	10	5	17	10	2017	23	20	39	10	6	17	10	2017	23	23	42	10	7	17	10	2017	23	27	17	10	8	17	10	2017	23	30	54	10	9	17	10	2017	23	32	38	10	10	17	10	2017	23	34	29	10	11	17	10	2017	23	36	51	10	12	17	10	2017	23	38	15	10	13	17	10	2017	23	40	19	10	14	17	10	2017	23	42	33	10	15	17	10	2017	23	45	50	10	16	17	10	2017	23	47	28	10
	A	B	C	D	E	F	G																																																																																																																																								
1	Día	Mes	Year	Hora	Minutos	Segundos	Temperatura																																																																																																																																								
2	17	10	2017	23	10	14	10																																																																																																																																								
3	17	10	2017	23	12	10	10																																																																																																																																								
4	17	10	2017	23	15	60	10																																																																																																																																								
5	17	10	2017	23	20	39	10																																																																																																																																								
6	17	10	2017	23	23	42	10																																																																																																																																								
7	17	10	2017	23	27	17	10																																																																																																																																								
8	17	10	2017	23	30	54	10																																																																																																																																								
9	17	10	2017	23	32	38	10																																																																																																																																								
10	17	10	2017	23	34	29	10																																																																																																																																								
11	17	10	2017	23	36	51	10																																																																																																																																								
12	17	10	2017	23	38	15	10																																																																																																																																								
13	17	10	2017	23	40	19	10																																																																																																																																								
14	17	10	2017	23	42	33	10																																																																																																																																								
15	17	10	2017	23	45	50	10																																																																																																																																								
16	17	10	2017	23	47	28	10																																																																																																																																								

HUMEDAD - Excel																																																																																																																																															
ARCHIVO	INICIO	INSERTAR	DISEÑO DE PÁGINA		FÓRMULAS																																																																																																																																										
<table border="1"> <thead> <tr> <th></th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> <th>G</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Día</td> <td>Mes</td> <td>Year</td> <td>Hora</td> <td>Minutos</td> <td>Segundos</td> <td>Humedad</td> </tr> <tr> <td>2</td> <td>17</td> <td>10</td> <td>2017</td> <td>23</td> <td>0</td> <td>0</td> <td>35</td> </tr> <tr> <td>3</td> <td>17</td> <td>10</td> <td>2017</td> <td>23</td> <td>12</td> <td>10</td> <td>34</td> </tr> <tr> <td>4</td> <td>17</td> <td>10</td> <td>2017</td> <td>23</td> <td>30</td> <td>19</td> <td>34</td> </tr> <tr> <td>5</td> <td>18</td> <td>10</td> <td>2017</td> <td>24</td> <td>10</td> <td>39</td> <td>34</td> </tr> <tr> <td>6</td> <td>18</td> <td>10</td> <td>2017</td> <td>24</td> <td>38</td> <td>18</td> <td>33</td> </tr> <tr> <td>7</td> <td>18</td> <td>10</td> <td>2017</td> <td>24</td> <td>46</td> <td>16</td> <td>33</td> </tr> <tr> <td>8</td> <td>18</td> <td>10</td> <td>2017</td> <td>24</td> <td>51</td> <td>30</td> <td>33</td> </tr> <tr> <td>9</td> <td>18</td> <td>10</td> <td>2017</td> <td>1</td> <td>19</td> <td>24</td> <td>33</td> </tr> <tr> <td>10</td> <td>18</td> <td>10</td> <td>2017</td> <td>1</td> <td>23</td> <td>60</td> <td>32</td> </tr> <tr> <td>11</td> <td>18</td> <td>10</td> <td>2017</td> <td>1</td> <td>40</td> <td>13</td> <td>32</td> </tr> <tr> <td>12</td> <td>18</td> <td>10</td> <td>2017</td> <td>1</td> <td>55</td> <td>45</td> <td>32</td> </tr> <tr> <td>13</td> <td>18</td> <td>10</td> <td>2017</td> <td>2</td> <td>20</td> <td>19</td> <td>32</td> </tr> <tr> <td>14</td> <td>18</td> <td>10</td> <td>2017</td> <td>2</td> <td>42</td> <td>35</td> <td>31</td> </tr> <tr> <td>15</td> <td>18</td> <td>10</td> <td>2017</td> <td>2</td> <td>51</td> <td>11</td> <td>31</td> </tr> <tr> <td>16</td> <td>18</td> <td>10</td> <td>2017</td> <td>2</td> <td>58</td> <td>41</td> <td>31</td> </tr> </tbody> </table>									A	B	C	D	E	F	G	1	Día	Mes	Year	Hora	Minutos	Segundos	Humedad	2	17	10	2017	23	0	0	35	3	17	10	2017	23	12	10	34	4	17	10	2017	23	30	19	34	5	18	10	2017	24	10	39	34	6	18	10	2017	24	38	18	33	7	18	10	2017	24	46	16	33	8	18	10	2017	24	51	30	33	9	18	10	2017	1	19	24	33	10	18	10	2017	1	23	60	32	11	18	10	2017	1	40	13	32	12	18	10	2017	1	55	45	32	13	18	10	2017	2	20	19	32	14	18	10	2017	2	42	35	31	15	18	10	2017	2	51	11	31	16	18	10	2017	2	58	41	31
	A	B	C	D	E	F	G																																																																																																																																								
1	Día	Mes	Year	Hora	Minutos	Segundos	Humedad																																																																																																																																								
2	17	10	2017	23	0	0	35																																																																																																																																								
3	17	10	2017	23	12	10	34																																																																																																																																								
4	17	10	2017	23	30	19	34																																																																																																																																								
5	18	10	2017	24	10	39	34																																																																																																																																								
6	18	10	2017	24	38	18	33																																																																																																																																								
7	18	10	2017	24	46	16	33																																																																																																																																								
8	18	10	2017	24	51	30	33																																																																																																																																								
9	18	10	2017	1	19	24	33																																																																																																																																								
10	18	10	2017	1	23	60	32																																																																																																																																								
11	18	10	2017	1	40	13	32																																																																																																																																								
12	18	10	2017	1	55	45	32																																																																																																																																								
13	18	10	2017	2	20	19	32																																																																																																																																								
14	18	10	2017	2	42	35	31																																																																																																																																								
15	18	10	2017	2	51	11	31																																																																																																																																								
16	18	10	2017	2	58	41	31																																																																																																																																								

**Figura 10-3** Datos de temperatura (izquierda) y humedad (derecha) en Microsoft Excel 2013.

Realizado por: PAREDES, Marjorie, 2017

### 3.3 Pruebas del nodo actuador

En la *Figura 11-3* se muestra el prototipo ensamblado del nodo actuador, este es el encargado de abrir o cerrar la electroválvula para el riego, además cuenta con un sensor medidor de caudal. Este nodo recibe la orden del nodo control para llevar a cabo el riego.



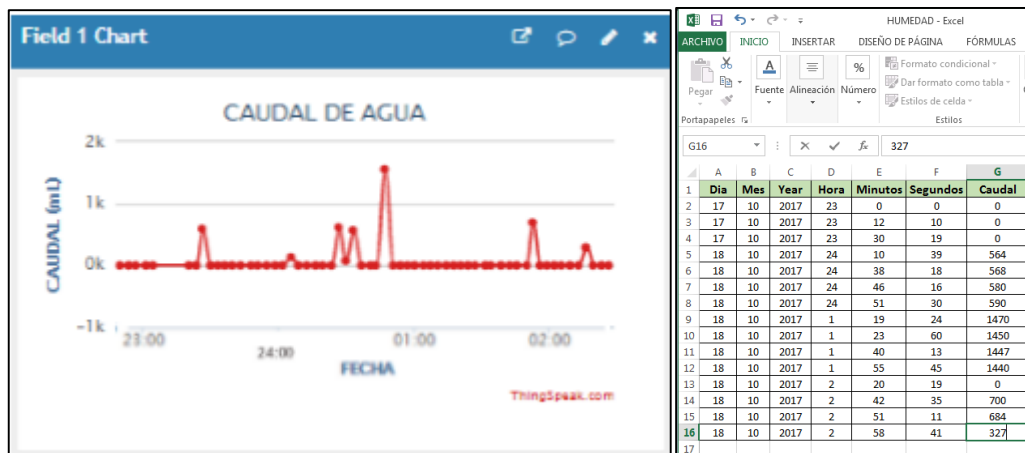
**Figura 11-3** Prototipo del Nodo Actuador, vista externa (izquierda) y vista interna (derecha)

Realizado por: PAREDES, Marjorie, 2017

## Comunicación entre tarjeta Arduino y el módulo ESP8266

Este nodo cuenta con un módulo wifi ESP8266 que transmite por medio de comunicación inalámbrica hacia la nube donde se almacena y se muestra de manera gráfica los datos leídos en la plataforma *ThingSpeak* como se muestra en la *Figura 12-3*.

El valor del caudal se va sumando con el pasar de las horas de riego, determinando así la cantidad de agua utilizada en un determinado tiempo. En la gráfica se muestra los resultados de tres horas de riego.



**Figura 12-3** Visualización del caudal en la plataforma ThingSpeak (izquierda), almacenamiento de la plataforma en Microsoft Excel 2013 (derecha).

Realizado por: PAREDES, Marjorie, 2017

La parcela de mora ubicada en la Facultad de Recursos Naturales cuenta con 140 plantas de mora, el terreno tiene una dimensión de 32x52 metros como se muestra en la *Figura 13-3*. Cada planta cuenta con cuatro goteros haciendo un total de 560 goteros en toda la parcela.





**Figura 13-3** Fotografía aérea de la parcela de mora  
Realizado por: PAREDES, Marjorie, 2017

En la *Tabla 3-3* se detalla la cantidad de agua utilizada en la parcela durante el monitoreo en tres horas, esta parcela utiliza goteros de 16 mm.

**Tabla 3-1** Consumo de agua en la parcela y por cada planta.

<b>GOTEROS DE 16mm - 0,30 L/seg</b>			
<b>AGUA CONSUMIDA POR TODA LA PARCELA</b>			
<b>Total –Goteros</b>	<b>Caudal Agua/ min</b>	<b>Minutos de Riego</b>	<b>Total Consumo (L)</b>
560	1,8	30	30,24
560	1,8	60	60,48
560	1,8	120	120,96
560	1,8	180	180,44
<b>AGUA CONSUMIDA POR CADA PLANTA</b>			
<b>Goteros por planta</b>	<b>Caudal Agua/ min</b>	<b>Minutos de Riego</b>	<b>Total Consumo (L)</b>
4	1,8	30	0,21
4	1,8	60	0,43
4	1,8	120	0,87
4	1,8	180	1,2

Realizado por: PAREDES, Marjorie, 2017

La parcela que cuenta con 140 plantas de mora utiliza 180,44 litros de agua en tres horas de riego que equivale a 1,2 litros por planta, SICAR determinó que tres horas de riego fueron suficientes para llegar al valor óptimo de hidratación de las plantas a diferencia de las cuatro horas que el CER riega, teniendo un ahorro considerable de 60,48 litros de agua.

### 3.4 Pruebas del Nodo Control

En la *Figura 14-3* se aprecia el prototipo ensamblado del nodo control, este nodo actúa como coordinador, es el encargado del funcionamiento de todo el sistema, el módulo xbee de este nodo es el que crea la red y agrega a los demás módulos del nodo sensor y actuador.



**Figura 14-3** Prototipo del Nodo Control, vista externa (izquierda) y vista interna (derecha)  
Realizado por: PAREDES, Marjorie, 2017

#### 3.4.1 Pruebas de almacenamiento

Este nodo cuenta con una tarjeta micro SD que almacena los datos enviados por los nodos sensor y actuador, los valores almacenados están disponibles para ser visualizados en un bloc de notas o en Microsoft Excel como se muestra en la *Figura 15-3*.

Id_Nodo	Día	Mes	Year	Hora	Minutos	Temperatura	Humedad	Caudal
6	20	10	2017	12	20	22	23	0
6	20	10	2017	12	21	22	23	0
6	20	10	2017	12	22	22	23	0
6	20	10	2017	12	23	22	23	0
6	20	10	2017	12	24	22	23	0
6	20	10	2017	12	25	22	23	0
6	20	10	2017	12	26	22	23	0
6	20	10	2017	12	27	22	23	0
6	20	10	2017	12	28	22	23	0
6	20	10	2017	12	29	22	24	0
6	20	10	2017	12	30	22	24	0
6	20	10	2017	12	31	22	24	0
6	20	10	2017	12	32	22	23	0
6	20	10	2017	12	33	22	24	0
6	20	10	2017	12	33	22	24	0
6	20	10	2017	12	34	22	24	0
6	20	10	2017	12	35	22	24	0
6	20	10	2017	12	36	22	24	0
6	20	10	2017	12	37	22	23	0
6	20	10	2017	12	38	22	23	0
6	20	10	2017	12	39	22	23	0

**Figura 15-3** Prueba de apertura del archivo en Microsoft Excel 2013.

Realizado por: PAREDES, Marjorie, 2017

## Comunicación Bluetooth y aplicación en el dispositivo móvil

El nodo control cuenta con un módulo Bluetooth que es el encargado de enviar datos en tiempo real a la aplicación móvil como se muestra en la *Figura 16-3*. El nodo sensor y actuador envían datos constantemente al nodo control y este los visualiza, el alcance del monitoreo es de 10 metros sin retrasos.

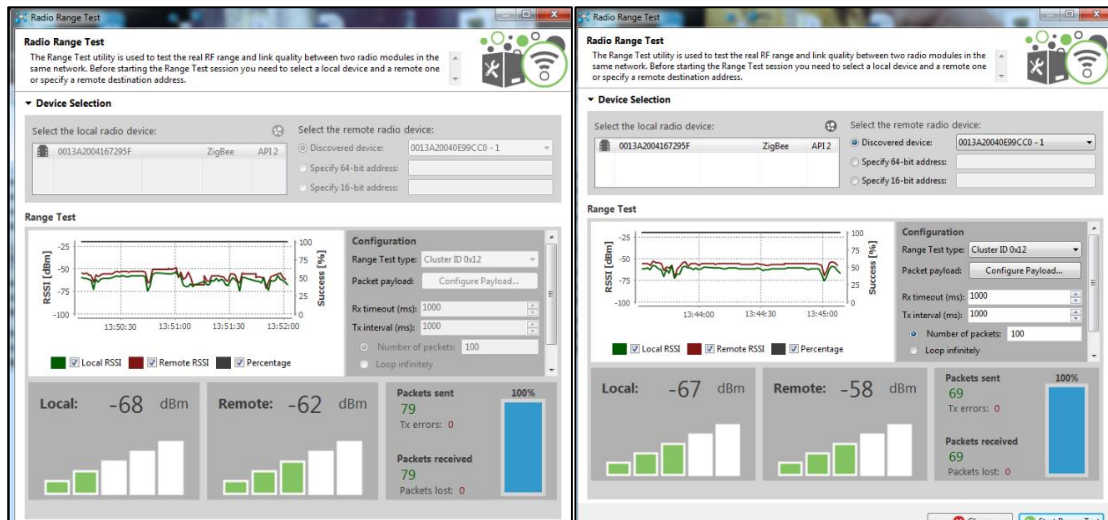


**Figura 16-3** Monitoreo en la aplicación móvil (izquierda), y visualización en el nodo sensor (derecha).

Realizado por: PAREDES, Marjorie, 2017

## Pruebas de comunicación entre nodos en el *software* XCTU

Se realiza las pruebas utilizando las herramientas disponibles en XCTU como se muestra en la *Figura 17-3*, teniendo una comunicación estable y sin pérdida de paquetes hasta 300 metros en línea de vista directa, a partir de esa distancia existe un retraso de dos a cinco segundos en la recepción, pero no pérdida de datos, para este estudio se consideraron 300 metros ya que esta distancia cumple con los requerimientos del sistema.



**Figura 17-3** Comunicación nodo control-sensor (izquierda) y comunicación nodo control-actuador (derecha).

Realizado por: PAREDES, Marjorie, 2017

La recepción del nodo actuador es mejor con -58 dBm a diferencia del nodo sensor que tiene una recepción de -62 dBm, además analizando la gráfica se nota que el envío de información es constante en la mayoría de tiempo, mientras que en el nodo sensor hay variación mas no pérdida de paquetes.

### 3.5 Funcionamiento general SICAR desarrollado para el CER – ESPOCH

Después de las pruebas respectivas de cada nodo que conforman el sistema inalámbrico y ser probados en los predios de la Facultad de Recursos Naturales de la ESPOCH se verificó el correcto funcionamiento de la comunicación entre todos sus elementos y la integridad de sus datos transmitidos. En la *Figura 18-3* se muestra el sistema inalámbrico funcional.



**Figura 18-3** SICAR en funcionamiento en la parcela de mora.  
Realizado por: PAREDES, Marjorie, 2017

### 3.6 Análisis económico de la implementación del SICAR

En este apartado se presenta la parte económica, gastos y costos de los dispositivos electrónicos utilizados en el prototipo implementado. En la *Tabla 3-3* se muestra el presupuesto detallado de los nodos que conforman el sistema inalámbrico.

**Tabla 4-3** Presupuesto detallado de cada elemento que conforman los nodos del SICAR

NODO	DISPOSITIVO	CANT.	COSTO UNITARIO (\$)	COSTO TOTAL (\$)
<b>SENSOR</b>	Tarjeta Arduino NANO	1	8,00	8,00
	Sensor de Humedad FC-28	1	6,00	6,00
	Sensor de Temperatura DH11	1	11,00	11,00
	Pantalla LCD 16x2	1	9,00	9,00
	Micro Switch	1	0,60	0,60
	Módulo GSM SIM900	1	55,00	55,00
	Módulo wifi MCU ESP8266	2	12,00	24,00
	Módulo Xbee S2	1	45,00	45,00
	Regulador de voltaje	1	8,00	8,00
	Regulador de voltaje solar	1	25,00	25,00
	Batería seca de 12 V- 6A	1	45,00	45,00
	Panel solar industrial	1	150,00	150,00
	Pantalla LCD 16x2	1	8,00	8,00
	Cable	5	0,10	0,50
	Led	2	0,05	0,10
	Resistencias	4	0,05	0,20
	Ensamblaje	1	40,00	40,00
<b>TOTAL NODO SENSOR</b>				<b>435,40</b>
<b>ACTUADOR</b>	Tarjeta Arduino NANO	1	8,00	8,00
	Módulo XBee S2	1	45,00	45,00
	Sensor de fuido de agua	1	12,00	12,00
	Módulo wifi MCU ESP8266	1	12,00	12,00
	Módulo relé	1	18,00	18,00
	Electroválvula	1	25,00	25,00
	Led	3	0,10	0,30
	Resistencias	3	0,05	0,15
	Cable	5	0,10	0,50
	Ensamblaje	1	10,00	10,00
<b>TOTAL NODO ACTAUDOR</b>				<b>130,95</b>
<b>CONTROL</b>	Tarjeta Arduino UNO	1	15,00	15,00
	Módulo Xbee S2	1	45,00	45,00
	Transformador 12V	1	5,00	5,00
	Módulo micro SD	1	5,00	5,00
	Tarjeta SD	1	5,00	5,00
	Bluetooth	1	10,00	10,00
	RTC (Reloj en tiempo real)	1	5,00	5,00
	Módulo relé	1	2,00	2,00
<b>TOTAL NODO CONTROL</b>				<b>92,00</b>
<b>COSTO TOTAL DEL PROTOTIPO SICAR</b>				<b>\$ 658,35</b>

Realizado por: PAREDES, Marjorie, 2017

De los valores obtenidos en la tabla anterior se puede apreciar que el módulo sensor es el más costoso con \$435,40 por el valor del panel solar que es de \$150 y el nodo más económico es el actuador con \$92,00 y el costo total del prototipo es de \$658,35.

A comparación de otros sistemas implementados en la ESPOCH este sistema automatizado tiene un 40% de ahorro siendo un factor considerable haciendo de SICAR un sistema de bajo costo.

## CONCLUSIONES

- Se implementó un prototipo de WSN con nodos inteligentes para el sistema de riego aplicado a la agricultura de precisión para el CER – ESPOCH, haciendo uso de la topología de red malla, utilizando tres tipos de comunicación inalámbrica: Zigbee, wifi y Bluetooth además hace uso de la red GSM.
- Se diseñó un sistema inalámbrico de sensores de supervisión y control de riego cuya operatividad se basa en tres nodos: sensor, actuador y control que cumplen funciones específicas. Se recurrió a la comunicación ZigBee por su alcance y velocidad de transmisión, teniendo un alcance de 300 metros en línea de vista directa sin pérdida de paquetes con una velocidad de transferencia de datos de hasta 250Kbps, los módulos que conforman SICAR pueden comunicarse simultáneamente con el nodo control.
- Se estableció de las pruebas realizadas de la comunicación Bluetooth que existe una comunicación sin pérdida hasta 10 metros, estas lecturas son visualizadas en la aplicación móvil desarrollada en app inventor, las variables que se visualizan son: identificador del nodo, humedad, temperatura y caudal, el módulo ESP8266 envía las lecturas de los sensores a la plataforma *ThingSpeak* donde se almacena y grafica los valores guardados, además estos datos se pueden exportar como un archivo .xlsx y ser almacenada en un ordenador para futuros estudios, también estos datos son almacenados en la tarjeta micro SD disponible en el nodo control
- Se determinó mediante la comparación de lecturas de los equipos que el sensor de humedad posee un error absoluto de  $\pm 5$  centibares y el de temperatura un error absoluto de  $\pm 1^{\circ}\text{C}$  y que el diseño no incorpora error a los sensores.
- Se implementó un identificador de nodo con la combinación del micro switch que da hasta 64 opciones, para esta implementación fue necesario un solo nodo ya que la humedad del suelo es cambiante a 100 metros a la redonda de la planta y la parcela que fue considerada para este estudio es de 50x32 metros siendo innecesario la implementación de más nodos sensores
- Se diseñó un nodo sensor móvil gracias a la alimentación a través de un panel solar y una batería de 12V – 4,5A esta batería almacena voltaje y amperaje que es utilizada



para el funcionamiento en la noche, las pruebas realizadas para determinar la duración de la batería arrojaron 19 horas de duración.

- Se determinó que las 140 plantas de mora utilizaron 180,44 litros de agua en tres horas de riego que equivale a 1,2 litros por planta, SICAR comprobó que tres horas de riego fueron suficientes para llegar al valor óptimo de hidratación de las plantas a diferencia de las cuatro horas que el CER riega, teniendo un ahorro considerable de 60,48 litros de agua por riego, teniendo un riego semanal de 3 veces tendríamos un ahorro de 725,76 litros de agua al mes y 8709,12 litros al año.

## **RECOMENDACIONES**

- Implementar alimentación a través de paneles solares a todos los nodos para tener un sistema completamente móvil e independiente.
- SICAR es un sistema escalable, lo que se recomienda la creación de más nodos sensores y actuadores para tener un mayor cubrimiento de cultivos.
- Incorporar nodos que permitan el análisis de variantes del suelo con el uso de nuevos sensores como: humedad ambiental, pH, nutrientes, salinidad, etc.
- Gestionar la inserción de la plataforma virtual al dominio privado de la Escuela Superior Politécnica de Chimborazo.

## **BIBLIOGRAFÍA**

**AGROUNICA.** *Capacidad de Campo.*[En línea] 2011.

[Citado el: 29 de Junio de 2017.]

<https://es.slideshare.net/AgrounicaBlogspot/sap-p2>.

**ALTAMIRANO BENALCÁZAR, Hector Rodrigo y PUENTE MORENO, Amanda Madelaine.** *Diseño e implementación de una red inalámbrica de sensores inteligentes para el registro de la contaminación acústica en la epoch basado en tarjetas de desarrollo (Tesis de pregrado).* [En línea] Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería en Electrónica, Telecomunicaciones y Redes, Riobamba, Ecuador.2016,pp. 15-73

[Citado el: 21 de 08 de 2017.]

<http://dspace.esPOCH.edu.ec/handle/123456789/6378>.

**ARDUINO.** *¿Qué es Arduino?* [En línea] 2017.

[Citado el: 17 de Mayo de 2017.]

<https://www.arduino.cc>.

**BAICHTAL, JHON.** ARDUINO FOR BEGINNERS: Essencial skills maker needs. USA : Pearson Education., 2014. pp. 92-93.

**BBVA.** *Internet de las Cosas.* BBVA API\_MARKET. [En línea] 2016.

[Citado el: 14 de Octubre de 2017.]

<https://bbvaopen4u.com/es/actualidad/apis-para-el-internet-de-las-cosas-thingspeak-pachube-y-fitbit>.

**CAPRILE, SERGIO R.** Equisbí: Desarrollo de aplicaciones con comunicación remota basadas en módulos ZigBee y 802.15.4. Buenos Aires : GAE, 2009. p. 209.

**CARBALLAR, JOSE ANTONIO.** Wi-fi: LO QUE NECESITA CONOCER. Madrid : RC Libros, 2010.

**CEDEÑO VERA, GISSELA ESTEFANIA .** *Implementación de sistema de control de riego mediante red inalámbrica de sensores de humedad (Tesis de pregrado).* [En línea] Universidad

Técnica Estatal de Quevedo, Facultad de Ciencias de la Ingeniería, Carrera de Ingeniería en Sistemas, Quevedo, Ecuador. 2014.

[Citado el: 02 de Junio de 2017.]

<http://repositorio.uteq.edu.ec/handle/43000/125>.

**CENTSYS.** *Panel Solar*. [En línea] 2016.

[Citado el: 11 de Octubre de 2017.]

[http://www.centsys.co.za/upload/CENTSYS%20Documentation/0\\_07\\_B\\_0133%2060W%20Solar%20Panel%20Specifications%20sheet-29062015-NG.pdf](http://www.centsys.co.za/upload/CENTSYS%20Documentation/0_07_B_0133%2060W%20Solar%20Panel%20Specifications%20sheet-29062015-NG.pdf).

**CÉSAR, JULIO.** *Familia de Arduino*. NODEBOTS. [En línea] 2014.

[Citado el: 20 de 08 de 2017.]

<http://slides.com/jurasec/nodebots--2#/3>.

**CÓDIGO21.** *Primeros pasos App Inventor*. [En línea] 2016.

[Citado el: 14 de Octubre de 2017.]

<http://codigo21.educacion.navarra.es/autoaprendizaje/primeros-pasos-con-app-inventor-2/>.

**DEALS, IRRIGATION.** *Electroválvula*. [En línea] 2016.

[Citado el: 20 de Junio de 2017.]

<http://www.irrigationdeals.com/es/electrovalvulas-de-riego-rain-bird/72-rain-bird-hv-f-1-24v-electrovalvula-con-control-de-caudal-hembra-100-hvf.html>.

**El Telégrafo.** *El agua siembra, riega y cosecha desarrollo*. [En línea] 2014.

[Citado el: 02 de Junio de 2017.]

<http://tinyurl.com/ogdgkpj>.

**ELECTRONILAB.** *Sensor de flujo de agua*. [En línea] 2016.

[Citado el: 23 de Agosto de 2017.]

<https://electronilab.co/tienda/sensor-de-flujo-de-agua-g12-1-30lmin/>.

**ELECTROTOOLS.** *Medir humedad del suelo*. [En línea] 2017.

[Citado el: 17 de Mayo de 2017.]

<http://www.electrontools.com/Home/WP/2016/05/17/medir-la-humedad-del-suelo-con-arduino/>.

**ESPRESSIF.** *EPS8266*. [En línea] 2015.

[Citado el: 16 de Mayo de 2017.]

<https://espressif.com/>.

**E-TOOLS.** *Humedad*. [En línea] 2016.

[Citado el: 18 de Mayo de 2017.]

<http://www.electrontools.com/Home/WP/2016/05/17/medir-la-humedad->.

**FALUDI, ROBERT.** *Wireless Sensor Networks*. Estados Unidos de América : O'Really Madia, 2010.

**FORSYTHE, WARREN.** *Física de Suelos*. San José, Costa Rica : IICA, 1985. p. 19.

**GÁLVEZ MOROCHO, JESSICA PAULINA y CAZARES ALEGRÍA, HIPATIA CECIBEL.** *Diseño e implementación de un sistemas de riego inteligente mediante lógica difusa para el Centro Experimental de Riego de la Facultad de Recursos Naturales (Tesis de pregrado)*. Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica, Control y Redes Industriales, Riobamba, Ecuador. 2016.

**GIBBSS, MARK.** *Raspberry Pi*, NETWORK WORLD. [En línea] 2016.

[Citado el: 20 de 08 de 2017.]

<http://www.networkworld.com/article/3111816/computers/the-discerning-nerds-guide-to-raspberry-pi-hardware-2016-mid-year-edition.html>.

**GONZÁLEZ, ANTONY GARCÍA.** El setud y el loop en Arduino. [En línea] 2015.

[Citado el: 14 de Octubre de 2017.]

<http://panamahitek.com/el-setup-y-el-loop-en-arduino/>.

**GUERRA SALAZAR, José.** *Diseño e implementación de un sistema inalámbrico para el*

*monitoreo en línea del microclima de invernaderos artesanales (Tesis). (Maestría).* Instituto Superior Politécnico "José Antonio Echeverría", Facultad de Ingeniería Eléctrica, Centro de Investigaciones en Microelectrónica, La Habana, Cuba. 2015. pp. 15-72.

**INTEL. IOT.** [En línea] 2015.

[Citado el: 20 de 08 de 2017.]

<https://software.intel.com/en-us/iot/hardware/galileo>.

**LARA, EDUARDO. SIM900 GSM GPRS .** [En línea] 2015.

[Citado el: 17 de Mayo de 2017.]

<https://www.hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>.

**LEITÓN, JUAN SANTIAGO. 1985. Riego y Drenaje.** Costa Rica : EUNED, 1985.

**LIBELIUM. Waspote.** [En línea] 2017.

[Citado el: 20 de Agosto de 2017.]

<http://www.libelium.com/products/waspote/overview/>.

**LIBELIUM. Xbee XCTU.** [En línea] 2017.

[Citado el: 26 de Mayo de 2017.]

<http://www.libelium.com/development/waspote/documentation/x-ctu-tutorial/>.

**LLAMAS, LUIS. Ingeniería, informática y diseño.** [En línea] 2016.

[Citado el: 15 de Mayo de 2017.]

<https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>.

**LLAMAS, LUIS. Humedad del suelo.** [En línea] 2016.

[Citado el: 17 de Mayo de 2017.]

<https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>.

**MANAGEMENT, FARM. WSN.** [En línea] 2017.

[Citado el: 17 de 08 de 2017.]

<http://www.farmmanagement.pro/the-use-of-wireless-sensor-networks-in-precision-agriculture/>.

**MEDINA, JUVENAL Y ZÚÑIGA VELANDO, CARLOS.** *Manual de Operación y Mantenimiento de un Sistema de Riego por Goteo.* Lima : PREDES, 2005.

**MFBARCELL.** Red de Sensores Inalámbricos. [En línea] 2016.

[Citado el: 28 de Mayo de 2017.]

<http://www.mfbarcell.es/conferencias/wsn.pdf>.

**EL MUNDO,** *Red de sensores que vigilan el medio ambiente en tiempo real.* [En línea] 2009.

[Citado el: 21 de 05 de 2017.]

<http://www.elmundo.es/elmundo/2009/12/22/baleares/1261469973.html>.

**NAVARRO, VICTOR HUGO CADENA.** *HABLEMOS DE RIEGO CON LOS AGRICULTORES.* Ecuador : s.n., 2014.

**PEDRERA, ANTONIO CAICEDO.** *Arduino para Principiantes.* 2ª Edición. España : IT Campus Academy, 2017.

**MIS REDES,** *Red de Sensores Inalámbricos.* [En línea] 2012.

[Citado el: 23 de 06 de 2017.]

<http://realmadriale.blogspot.com/>.

**RIVERBED.** *Riverbed Modeler.* [En línea] 2017.

[Citado el: 14 de 06 de 2017.]

<https://www.riverbed.com/mx/products/steelcentral/steelcentral-riverbed-modeler.html>.

**RUIZ, ANTONIO Y MOLINA, JOSÉ.** *Autoamtización y telecontrol de sistemas de riego.* Barcelona : Marcombo, 2010.

**SINTELWE.** *Módulos Xbee.* [En línea] 2012.

[Citado el: 21 de 08 de 2017.]

<http://sintelweb.blogspot.com/2012/11/configuracion-de-dos-modulos-xbee.html>.

**SPARKFUN.** *Módulo wifi ESP8266.* [En línea] 2016.

[Citado el: 16 de Mayo de 2017.]

<https://www.sparkfun.com/products/13678>.

**AGENCIA TI.** *Plataforma E-Health.* [En línea] 2015.

[Citado el: 18 de 05 de 2017.]

<http://smart.agenciati.com/Productos/health.php>.



# ANEXOS

# Anexo 1: Hojas técnicas de los elementos que conforman el SICAR

## Arduino Uno



**Arduino Uno R3 Front**      **Arduino Uno R3 Back**  
**Arduino Uno R3 Front**      **Arduino Uno SMD**      **Arduino Uno Front**      **Arduino Uno Back**

### Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 5 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega16U2 (ATmega8U2 up to version R2) programmed as a USB-to-serial converter.

**Revision 2** of the Uno board has a resistor pulling the BU2 HWB line to ground, making it easier to put into **DFU mode**.

**Revision 3** of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- ATmega 16U2 replace the BU2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

### Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V

### Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

### Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference and tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega16U2 (or BU2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/BU2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the BU2.
- On Rev2 or later boards: there is a resistor that pulling the BU2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

### Automatic (Software) Reset

Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 Mhz

### Schematic & Reference Design

EAGLE files: [arduino-uno-rev3-reference-design.zip](#) (NOTE: works with Eagle 6.0 and newer)  
Schematic: [arduino-uno-rev3-schematic.pdf](#)

**Note:** The Arduino reference design can use an ATmega8, 168, or 328. Current models use an ATmega328, but an ATmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

### Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

### Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

### Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

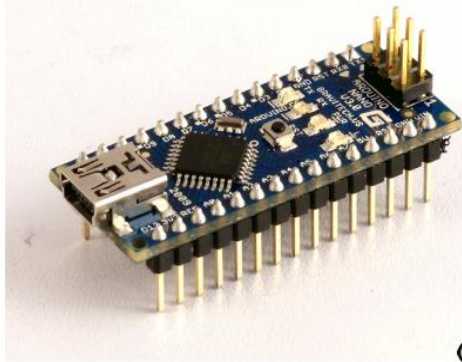
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

### USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

### Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



## Product Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

## Index

## Technical Specifications

Page 2

## How to use Arduino Programming Environment, Basic Tutorials

Page 6

## Terms &amp; Conditions

Page 7



radiospares RADIONICS



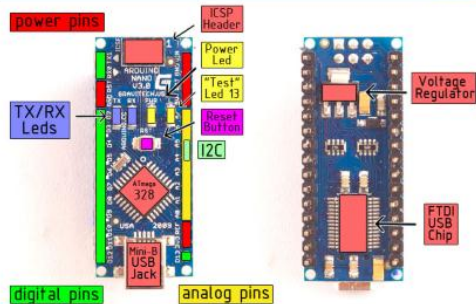
Arduino Nano 3.0 (ATmega328): [schematic](#), [Eagle files](#).

Arduino Nano 2.3 (ATmega168): [manual](#) (pdf), [Eagle files](#). Note: since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

## Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

## the board



radiospares RADIONICS



## Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

## Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the [EEPROM library](#)); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

## Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- PC: 4 (SDA) and 5 (SCL).** Support PC (TWI) communication using the [Wire library](#) (documentation on the Wiring website).

There are a couple of other pins on the board:

- AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega168 ports](#).



radiospares RADIONICS



## Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the [FTDI library](#) (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

## Programming

The Arduino Nano can be programmed with the Arduino software ([download](#)). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header file](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.



radiospares RADIONICS



### VANNES SÉRIE HV

Des performances exceptionnelles. Une durabilité inégalée.

- Conception compacte, rayon de rotation de 6,5 cm pour les installations en espace étroit
- Outils de votre choix pour ouvrir la vanne (clé, tournevis cruciforme, tournevis plat)
- Membrane asymétrique pour une fermeture lente, évitant les coups de bélier

#### CARACTÉRISTIQUES

##### Facilité de maintenance

- Vis capotés multi-usages pour faciliter la maintenance
- Action rapide à la membrane avec seulement quatre vis
- Détachement de membrane pour une maintenance facile

##### Fiabilité

- Corps résistant en polypropylène renforcé de fibre de verre
- Conception à débit inverse, normalement fermée
- Maintenance facile, peu de pièces détachées
- Membrane Buna N avec filtres auto-rotatifs de 200 microns et ressort en acier inoxydable

##### Polyvalence

- Fonctionne dans les applications à faible débit et "Xerigation" quand le filtre RBV est installé en amont
- Purge manuelle externe permettant d'évacuer les saletés et débris lors de l'installation ou du démarrage du système
- Ouverture manuelle sans fuite d'eau

#### SPÉCIFICATIONS

Pression: 1,0 à 10,3 bar  
Débit: 0,05 à 6,62 m<sup>3</sup>/h ; pour les débits inférieurs à 0,08 m<sup>3</sup>/h ; ou toute application "Xerigation", utiliser le filtre RBV-100-200MX installé en amont  
Température: température de l'eau jusqu'à 43 °C ; température ambiante jusqu'à 52 °C

#### SPÉCIFICATIONS ÉLECTRIQUES

Solénóide 24VCA 50/60Hz  
Courant d'appel maximum: 0,220 A @ 60Hz  
Courant de maintien: 0,143 A @ 60Hz  
Résistance de bobine: 52 à 55 Ohms

#### Partie de pression de vanne (bar)

1" HV (psi)	1" HV-MB (psi)
0,20	0,11
0,25	0,14
0,30	0,16
0,35	0,19
0,40	0,23
0,45	0,27
0,50	0,31
0,55	0,36

#### Comment commander

100 - HV - MM

100 - HV-MB

100 - HV-SS

100 - HVF

### Serie HV

Valvula de alto valor. Alto rendimiento. Grandes ahorros.

#### Características

- Diseño excéntrico Buna N patentado de presión equilibrada con filtro de agua piloto autolimpiante hasta 30 (200 microns) y resorte caudate de acero inoxidable - El diseño excéntrico ofrece un cierre más suave, con menos golpe de ariete
- Solo cuatro tornillos caudate de bronce durables que salen con la mitad del número de vueltas para un servicio rápido y sencillo - por lo menos dos veces más rápido que la competencia
- Cuerpo de polipropileno con fibra de vidrio para mayor resistencia (modelo talón a talón con cuerpo de PVC)

Todos los configuraciones populares de modelos disponibles:

- Diseño compacto, radio de giro de 2,54" (6,45 cm) para instalaciones estrechas
- Diseño de caudal inverso, normalmente cerrado
- Purga externa para limpiar el sistema manualmente y quitar las partículas de suciedad durante la instalación y puesta en marcha
- Purga interna para una operación manual en seco
- Funciona en aplicaciones de baja caudal y "Xerigation" cuando el filtro malla 200 se instala en contracorriente

#### Dimensiones

- Altura: 4.42" (11,2 cm)
- Altura (MB): 4.50" (11,4 cm)
- Longitud: 4.4" (11,2 cm)
- Longitud (MB): 5.08" (12,8 cm)
- Ancho: 3.1" (7,9 cm)

#### Pérdida de presión de la válvula HV (psi)

Caudal (gpm)	1" HV (psi)	1" HV-MB (psi)
1	1,57	1,73
3	2,07	2,03
5	2,38	2,25
10	3,33	2,80
20	4,59	4,45
30	6,14	7,85
40	8,23	13,68

#### Pérdida de presión de la válvula HV (psi) SIST. METRICO

Caudal (m <sup>3</sup> /h)	Caudal (lit/s)	1" HV (bares)	1" HV-MB (bares)
0,25	0,06	0,11	0,12
0,75	0,21	0,14	0,14
1,00	0,28	0,16	0,16
2,00	0,56	0,23	0,19
5,00	1,39	0,32	0,31
7,50	2,08	0,42	0,54
9,10	2,52	0,57	0,94

\* Este índice recomienda que los índices de caudal en la línea de suministro no superen los 2,5 pie/seg. (0,3 m/s) para reducir los efectos del golpe de ariete.

#### Modelos

- 100HV-NPT: 1" (26/34) NPT hembra x hembra\*
- 100HV-SS: 1" (26/34) talón x talón
- 100HV-MB: 1" (26/34) macho x conector de punta
- 100HV-F: 1" (26/34) NPT hembra x hembra\*
- 100HV-SS: 1" (26/34) talón x talón

\* Disponible con rosas BSP

#### Recomendaciones

- Este índice recomienda que los índices de caudal que producen velocidad de desajuste en la línea de suministro no superen los 7,3 pie/s (2,1 m/s) para reducir los efectos del golpe de ariete.
- Los valores de uso residencial de este índice se pueden utilizar con módulos reguladores de presión PWS.
- No se recomienda su uso con sistemas de alto caudal.

#### Cómo especificar

100 - HV - SS

Configuración estándar: 1" (26/34) NPT hembra x hembra

100 - HV - SS: 1" (26/34) talón x talón

100 - HV - MB: 1" (26/34) macho x conector de punta

100 - HV - F: 1" (26/34) NPT hembra x hembra

100 - HV - SS: 1" (26/34) talón x talón

RELAY ISO9002

**SRD**

### 1. MAIN FEATURES

- Switching capacity available by 10A in spite of small size design for highdensity P.C. board mounting technique.
- UL, CUL, TUV recognized.
- Selection of plastic material for high temperature and better chemical solution performance.
- Sealed types available.
- Simple relay magnetic circuit to meet low cost of mass production.

### 2. APPLICATIONS

- Domestic appliance, office machine, audio, equipment, automobile, etc.
- (Remote control TV receiver, monitor display, audio equipment high rushing current use application.)

### 3. ORDERING INFORMATION

SRD	XX VDC	S	L	C
Model of relay	Nominal coil voltage	Structure	Coil sensitivity	Contact form
SRD	03, 05, 06, 09, 12, 24, 48VDC	S-Sealed type	L:0.36W	A:1 form A B:1 form B C:1 form C
		F:Flux free type	D:0.45W	

### 4. RATING

CCC	FILE NUMBER:CH0052885-2000	7A/240VDC
CCC	FILE NUMBER:CH0036746-99	10A/250VDC
UL /CUL	FILE NUMBER: E167996	10A/125VAC 28VDC
TUV	FILE NUMBER: R9933789	10A/240VAC 28VDC

### 5. DIMENSION (unit:mm) DRILLING (unit:mm) WIRING DIAGRAM

### 6. COIL DATA CHART (AT20°C)

Coil Sensitivity	Coil Voltage Code	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω) ±10%	Power Consumption (W)	Pull-In Voltage (VDC)	Drop-Out Voltage (VDC)	Max.-Allowable Voltage (VDC)
SRD (High Sensitivity)	03	03	120	25	abt. 0.36W	75%Max.	10% Min.	120%
	05	05	71.4	70				
	06	06	60	100				
	09	09	40	225				
	12	12	30	400				
	24	24	15	1600				
SRD (Standard)	03	03	150	20	abt. 0.45W	75% Max.	10% Min.	110%
	05	05	85.3	55				
	06	06	75	80				
	09	09	50	180				
	12	12	37.5	320				
	24	24	18.7	1280				

### 7. CONTACT RATING

Item	Type	SRD FORM C	SRD FORM A
Contact Capacity		7A 28VDC	10A 28VDC
Resistive Load (cosφ=1)		10A 125VAC	10A 240VAC
Inductive Load (cosφ=0.4 LFI=7msec)		3A 120VAC	5A 120VAC
		3A 28VDC	5A 28VDC
Max. Allowable Voltage		250VAC/110VDC	250VAC/110VDC
Max. Allowable Power Force		800VAC/240W	1200VAC/300W
Contact Material		AgCdO	AgCdO

### 8. PERFORMANCE (at initial value)

Item	Type	SRD
Contact Resistance		100mΩ Max.
Operation Time		10msec Max.
Release Time		5msec Max.
Dielectric Strength		
Between coil & contact		1500VAC 50/60HZ (1 minute)
Between contacts		1000VAC 50/60HZ (1 minute)
Insulation Resistance		100 MΩ Min. (500VDC)
Max. ON/OFF Switching		
Mechanically		300 operation/min
Electrically		30 operation/min
Ambient Temperature		-25°C to +70°C
Operating Humidity		45 to 85% RH
Vibration		
Endurance		
Error Operation		10 to 55Hz Double Amplitude 1.5mm
Shock		
Endurance		100G Min.
Error Operation		10G Min.
Life Expectancy		
Mechanically		10 <sup>7</sup> operations. Min. (no load)
Electrically		10 <sup>6</sup> operations. Min. (at rated coil voltage)
Weight		abt. 10gms.

### 9. REFERENCE DATA



## XBEE® S1 802.15.4 RF MODULES

Easy-to-deploy modules provide critical end-point connectivity to devices and sensors

XBee RF modules provide OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/Mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the XBee can substitute one XBee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-to-market.

XBee 802.15.4 RF modules are ideal for applications requiring low latency and predictable communication timing. Providing quick, robust communication in point-to-point, peer-to-peer, and multipoint/star configurations, XBee 802.15.4 products

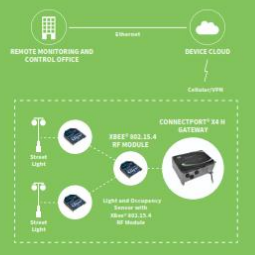
enable robust end-point connectivity with ease. Whether deployed as a pure cable replacement for simple serial communication, or as part of a more complex hub-and-spoke network of sensors, XBee 802.15.4 RF modules maximize performance and ease of development.

XBee 802.15.4 modules seamlessly interface with compatible gateways, device adapters and range extenders, providing developers with true beyond-the-horizon connectivity.

### BENEFITS

- Simple, out-of-the-box RF communications, no configuration needed
- Point-to-multipoint network topology
- 2.4 GHz for worldwide deployment
- Common XBee footprint for a variety of RF modules
- Low-power sleep modes
- Multiple antenna options

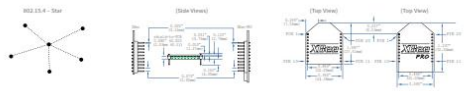
### APPLICATION EXAMPLE



### RELATED PRODUCTS



SPECIFICATIONS	Legacy XBee® S1 802.15.4	Legacy XBee-PRO® S1 802.15.4
<b>PERFORMANCE</b>		
RF DATA RATE	250 kbps	250 kbps
MODULUS RANGE	300 ft (90 m)	300 ft (90 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	2000 ft (600 m)	2000 ft (600 m)
TRANSMIT POWER	1 mW (+10 dBm)	10 mW (+10 dBm)
RECEIVER SENSITIVITY (2% FER)	-92 dBm	-100 dBm
DIGI HARDWARE	52	52
TRANSCIVER CHIPSET	Proximax MCR1312L	Proximax MCR1312L
<b>FEATURES</b>		
SERIAL DATA INTERFACE	0.7V CMOS UART	0.7V CMOS UART
CONFIGURATION METHOD	API or AT Commands, local or over the air	API or AT Commands, local or over the air
FREQUENCY BAND	2.4 GHz	2.4 GHz
INTERFERENCE IMMUNITY	0000 (Spread Spectrum Spreading)	0000 (Spread Spectrum Spreading)
SERIAL DATA RATE	2500 bps - 250 kbps	2500 bps - 250 kbps
ADC INPUTS	80 10-bit ADC inputs	80 10-bit ADC inputs
DIGITAL I/O	8	8
ANTENNA OPTIONS	Chip, Wire Whip, U.F.L. & RPMAA	Chip, Wire Whip, U.F.L. & RPMAA
<b>NETWORKING &amp; SECURITY</b>		
ENCRYPTION	128-bit AES	128-bit AES
RELIABLE PACKET DELIVERY	Medium/Noncontiguous	Medium/Noncontiguous
ADDRESS CHANNELS	16Kb/s, 56 and 800 KAC, 16 Channels	16Kb/s, 56 and 800 KAC, 16 Channels
<b>POWER REQUIREMENTS</b>		
SUPPLY VOLTAGE	2.8 - 3.6VDC	2.8 - 3.6VDC
TRANSMIT CURRENT	40 mA @ 3.3VDC	110 mA @ 3.3VDC
RECEIVE CURRENT	10 mA @ 3.3VDC	10 mA @ 3.3VDC
POWER-DOWN CURRENT	<100 µA @ 27° C	<100 µA @ 27° C
<b>REGULATORY APPROVALS</b>		
FCC (USA)	OUR XBEE	OUR XBEEPRO
IC (CANADA)	4234A XBEE	4234A XBEEPRO
ETSI (EUROPE)	Yes	Yes - Max 10.0 dB
C.TICK AUSTRALIA	Yes	Yes
TELEC (JAPAN)	Yes	Yes



WWW.DIGI.COM DIGI

PART NUMBERS	DESCRIPTION
<b>KITS</b>	
XBEE-AT-900C	Wireless Connectivity Kit w/ Legacy XBee 802.15.4
XBEE-AT-900D	XBee / Arduino Compatible Coding Platform w/ Legacy XBee 802.15.4
<b>MODULES</b>	
XBEE-800-001	XBee S1 802.15.4 low power module w/ wire antenna
XBEE-800-002	XBee S1 802.15.4 low power module w/ PCB antenna
XBEE-800-003	XBee S1 802.15.4 low power module w/ U.F.L. connector
XBEE-800-004	XBee S1 802.15.4 low power module w/ RPMAA connector
XBEE-800-005	XBee PRO S1 802.15.4 extended range module w/ wire antenna
XBEE-800-006	XBee PRO S1 802.15.4 extended range module w/ wire antenna (International)
XBEE-800-007	XBee PRO S1 802.15.4 extended range module w/ U.F.L. connector
XBEE-800-008	XBee PRO S1 802.15.4 extended range module w/ U.F.L. connector (International)
XBEE-800-009	XBee PRO S1 802.15.4 extended range module w/ U.F.L. connector
XBEE-800-010	XBee PRO S1 802.15.4 extended range module w/ RPMAA connector
XBEE-800-011	XBee PRO S1 802.15.4 extended range module w/ RPMAA connector (International)
XBEE-800-012	XBee PRO S1 802.15.4 extended range module w/ PCB antenna
XBEE-800-013	XBee PRO S1 802.15.4 extended range module w/ PCB antenna (International)

FOR MORE INFORMATION  
PLEASE VISIT [WWW.DIGI.COM](http://WWW.DIGI.COM)

DIGI SERVICE AND SUPPORT / You can purchase with confidence knowing that Digi is always available to serve you with expert technical support and our industry leading warranty. For detailed information visit [www.digi.com/support](http://www.digi.com/support).  
© 1996-2016 Digi International Inc. All rights reserved.  
All trademarks are the property of their respective owners.

DIGI INTERNATIONAL WORLDWIDE HQ  
877-912-3444 / 952-912-3444 / [www.digi.com](http://www.digi.com)

DIGI INTERNATIONAL FRANCE  
+33 2 52 52 98 98 / [www.digi.fr](http://www.digi.fr)

DIGI INTERNATIONAL JAPAN  
+81 3 5429 0261 / [www.digi-int.co.jp](http://www.digi-int.co.jp)

DIGI INTERNATIONAL SINGAPORE  
+65 6213 5380

DIGI INTERNATIONAL CHINA  
+86 22 59622281 / [www.digi.com.cn](http://www.digi.com.cn)

31051412  
C/416

1. General Overview

1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

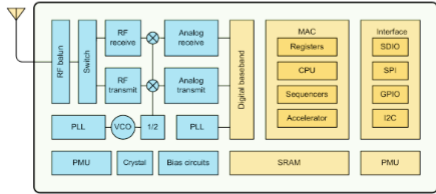


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).

Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

1.3. Parameters

Table 1 Parameters

6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup: GPIO16
9	MTMS	I/O	GPIO14; HSPL_CLK
10	MTDI	I/O	GPIO12; HSPL_MISO
11	VDDPST	P	Digital/I/O Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPL_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPL_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPL_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/I/O Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Q); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Q); SPIWP; HSPWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Q); SPL_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Q); SPL_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Q); SPL_MISO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Q); SPL_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPL_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V~3.6V
30	VDDA	P	Analog Power 3.0V~3.6V
31	RES12K	I	Serial connection with a 12 kQ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)

Note: GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode.

2.2. Electrical Characteristics

Table 3 ESP8266EX Electrical Characteristics

Parameters	Conditions	Min	Typical	Max	Unit
Storage Temperature Range		-40		125	°C
Maximum Soldering Temperature	IPC/JEDEC J-STD-020			260	°C
Working Voltage Value		3.0	3.3	3.6	V
I/O	V <sub>L</sub> /V <sub>H</sub>	-0.3/0.75V <sub>IO</sub>		0.25V <sub>IO</sub> /3.6	V
	V <sub>OL</sub> /V <sub>OH</sub>	N/A		0.1V <sub>IO</sub> /N	V
	I <sub>OH</sub>			12	mA
Electrostatic Discharge (HBM)	TAMB=-25°C			2	KV
Electrostatic Discharge (CDM)	TAMB=-25°C			0.5	KV

2.3. Power Consumption

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

Table 4 Description on Power Consumption

Parameters	Min	Typical	Max	Unit
Tx 802.11b, CCK 11Mbps, P <sub>OUT</sub> =+17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P <sub>OUT</sub> =+15dBm		140		mA
Tx 802.11n, MCS7, P <sub>OUT</sub> =+13dBm		120		mA
Rx 802.11b, 1024 bytes packet length, -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA

## GSM Shield SIM900

### Get Starting



This GPRS Shield is compatible with all boards which have the same form factor (and pinout) as a standard Arduino Board. **GPRS module** delivers GSM/GPRS 850/900/1800/1900MHz performance for voice, SMS, Data, and Fax in a small form factor.  
 The GPRS Shield is configured and controlled via its UART using simple **AT commands**. You can use the **2 Jumper block** to connect the SIM900 UART post to any pins within D0-D3 (for Hardware/Software serial port).  
 ECom not only can use the S\_PWR button for power on, but also can use the digital pin (D6) of Arduino to power on and reset (D5) the SIM900 module.

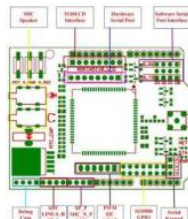
### Feature

- Fully compatible with Arduino / Uno and Mega.
- Free serial port connecting - you can select Hardware Serial port (D0/D1) control or Software Serial port (D2/D3) control it.
- SIM900 all pins breakout. Not just the UART port and debug port be layout, but also all pins on SIM900 be layout to the 2.54 standard pitch.
- Super capacitor power supply for the RTC.
- ECom not only can use the button for power on, but also can use the digital pin of Arduino to power on and reset the SIM900 module.
- Quad-Band 850/ 900/ 1800/ 1900 MHz
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Control via AT commands (GSM 07.07 ,07.05 and ECOM enhanced AT Commands)
- SIM application toolkit
- Supply voltage range : 3.1 ... 4.8V
- Low power consumption: 1.5mA(sleep mode)
- Operation temperature: -40°C to +85 °C
- Dimension:68.33x53.09mm(Same dimension of Arduino main board)

### Cautions

- **Make sure add using 9V Charger for power supply for your Arduino board and ECom. The 9V Charger we will provide to you with ECom Shield. Because of the power supply range of SIM900 is from 3.2V to 4.8V. The transmitting burst will cause voltage drop and the power supply must be able to provide sufficient current up to 2A. The USB port cannot supply such a large current.**
- **Make sure your SIM card is unlocked.**
- **The product is provided as is without an insulating enclosure. Please observe ESD precautions specially in dry (low humidity) weather.**
- **The factory default setting for the GPRS Shield UART is 19200 bps 8-N-1. (Can be changed using AT commands).**

### Hardware



Top-view



Super RTC Cap and Line In



LCD5100 interface and Software SerialPort Jumper



SIM Card Connector

### Light Status

LED	State	Function
Status	Off	Power Off
Status	On	Power On
Netlight	Off	SIM900 is not working
	64ms On/800ms Off	SIM900 does not find the network
	64ms On/3000ms Off	SIM900 find the network
	64ms On/300ms Off	GPRS communication

### Getting Started

We will use two methods

1. Using AT commands.
2. Using Arduino code (Uno - Mega) compatible.

Now we will prepare the Arduino board for communicating with the PC by UART protocol. Emulate a second serial port (UART) using software on the digital pins D2 and D3 and patch through all the communication between this second software serial port and the actual hardware serial port.  
 By doing this, all the data coming from the computer (connected to the actual hardware UART) would be relayed as is to the GPRS Shield (connected to software UART).

- Run Arduino IDE 1.0 or later
- Open new sketch and write a below code.

```
SoftwareSerial mySerial(2, 3);

void setup()
{
  mySerial.begin(19200);           // the GPRS baud rate
  Serial.begin(19200);           // the GPRS baud rate
}
```

## DHT 11 Humidity & Temperature Sensor

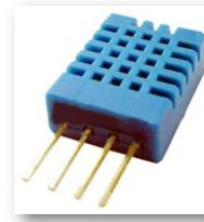


### 1. Introduction

This DFRobot DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

### Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
<b>Humidity</b>				
Resolution		1%RH	1%RH	1%RH
Repeatability			8 Bit	
Accuracy	25°C		± 1%RH	
	0-50°C		± 4%RH	
Interchangeability	Fully interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/(63%)25°C	6.5	10.5	15.5
	1m/s Air			
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
<b>Temperature</b>				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/(63%)	6.5		30.5



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

### 2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50°C	± 5%RH	± 2°C	1	4 Pin Single Row

### 3. Typical Application (Figure 1)

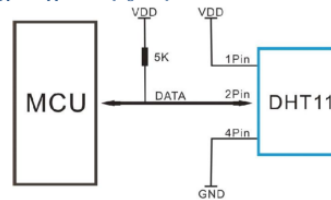


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

### 4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

### 5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.

**Data format:** 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".





## 60W Solar panel specifications

Our range of solar panels are constructed from ultra-efficient polycrystalline and have been designed to provide a reliable and cost-effective alternative energy solution for applications where mains power is either not available, or not desirable.

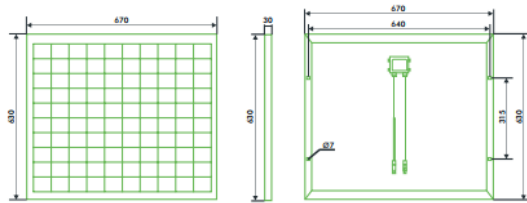
### Main features:

- Clever engineering makes for minimal power losses during prolonged periods in shade
- Fully weather-resistant - durable white tempered glass, EVA resin and weather-resistant film enclosed in anodised aluminium frame

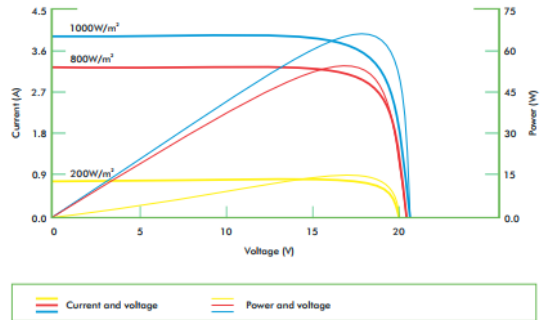
### Technical specifications

Type Of Module	60W
Maximum Power	60W
Tolerance	± 3%
Open Circuit Voltage	22.2V
Short Circuit Current	3.56A
Maximum Power Voltage	18.2V
Maximum Power Current	3.3A
Module Efficiency	14.22%
Solar Cell Efficiency	17.26%
Series Fuse Rating	15A
Terminal Box	IP65
Maximum system voltage	1000V DC
Operating Temperature	-40°C - 85°C
Dimensions	630mm x 670mm x 30mm
Weight	5.1kg

### Physical dimensions



### Current-voltage and power-voltage characteristics - various irradiance levels

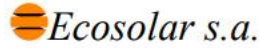


### Electrical characteristics

#### Electric performance typical performance characteristics

Current Temperature Coefficient %/°C	+0.06
Voltage Temperature Coefficient %/°C	-0.34
Power Temperature Coefficient %/°C	-0.47

Performance Warranty: 90% output, 12 year  
80% output, 25 years



## Manual del Usuario e Instalador del Regulador de Carga Solar Solsum

Ecosolar es marca registrada de Ecosolar s.a.

Monroe 2630 piso 6º B  
C1428BLR - Buenos Aires  
República Argentina  
Teléfono: ++054 - 11 - 4545-4679  
Email: ecosolar@ecosolarsa.com

Si el producto no puede ser reparado o reemplazado, o esto no ocurre en un período de tiempo razonable, se deberá reemplazar o proveer de las funciones que no puede realizar el producto; si esto no fuese suficiente para el interés del cliente, el contrato se cancelará.

No se asume responsabilidades por daños emergentes o consecuentes, daños causados por el mal uso o instalación de cualquier elemento no autorizado.

Asimismo no será aceptada responsabilidad alguna por los inconvenientes, daños o perjuicios que pudiera producir el uso del equipo al comprador o a terceras personas bajo cualquier circunstancia.

No se aceptará reclamos por lucro cesante, daños y perjuicios por la no disposición del equipo antes, durante y después del período necesario para realizar las reparaciones al mismo.

Información adicional que no figura en el original del Manual Solsum (que están en inglés y en alemán).

### Consejos para la instalación.

El regulador se debe instalar lo más cercano posible a las baterías y protegido contra las inclemencias del tiempo.

Asegúrese de instalar las baterías y regulador en un sitio ventilado.

Debe instalar el regulador con las borneras hacia abajo.

Asegúrese que la corriente proveniente de los paneles solares no sobrepase el máximo amperaje admisible por el regulador.

A los efectos que realice todas las funciones (en especial de la protección por descarga profunda), debe conectarse a los paneles, baterías y consumo.

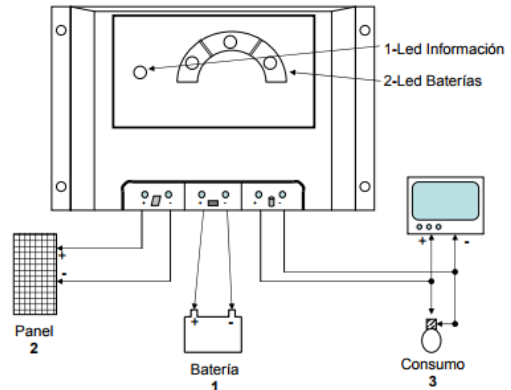
Todos los componentes del sistema (paneles fotovoltaicos, baterías, regulador de carga y consumo) deben tener la misma tensión. Verifique esto antes de comenzar la instalación. En caso de duda consulte con su vendedor.

Preste mucha atención a la polaridad de cada componente.

En las baterías se almacena una gran cantidad de energía. En el caso de un corto circuito, esta energía puede liberarse en un instante y puede ocasionar fuego por la cantidad de calor originada en el corto circuito.

Consejos útiles a tener en cuenta.

Evite a toda costa que se produzcan corto circuitos: peligro de fuego.



### Paso 1: Conecte las baterías.

Conecte los cables con la polaridad correcta al par medio de conectores del regulador (con el símbolo de baterías).

Si hay algún fusible externo quítelo momentáneamente. Conecte el borne positivo (+) con el cable correspondiente al positivo de baterías, luego conecte el borne negativo (-) con el cable correspondiente al negativo de baterías. Si hay fusible externo, vuelva a conectarlo.

Si la polaridad es la correcta el Led Información se prende en verde.

### Paso 2: Conecte los paneles fotovoltaicos.

Asegúrese que los paneles están protegidos contra una posible exposición a la luz (cúbralos o espere a la noche)

Asegúrese que los paneles no exceden el máximo de corriente permitida para el regulador.

Equipamiento que no debe ser desconectado por batería baja (por ejemplo, en un barco, equipamiento de navegación), conéctelo directamente a las baterías (en este caso coloque un fusible).

En instalaciones de corriente continua, bajo determinadas circunstancias o cuando se conecta o desconecta algún equipo, pueden producirse chispas. Teniendo esto presente, no coloque reguladores y baterías en ambientes donde existan materiales inflamables o explosivos (por ejemplo emanaciones de garrafas de gas, pinturas, solventes, etc.); así como también tome los cuidados necesarios en la instalación, no teniendo este tipo de materiales cerca. En caso de duda: consulte con un experto.

### 11. Datos técnicos.

Regulador a 25° C	Solsum 5.0	Solsum 8.0	Solsum 6.6	Solsum 8.8	Solsum10.10
Voltaje	12 V		12 V (24 V)		
Amp Máx de panel	5 A	8 A	6 A	8 A	10 A
Amp Máx carga	5 A*1	8 A*1	6 A	8 A	10 A
Función LVD	no	no	si	si	si
Máx voltaje panel	47 V				
Máx Autoconsumo	< 4 mA				
Tensión final	13,9 V (27,4 V)				
Tens. equalización	14,4 V (28,8 V)				
Tensión de reconexión (LVR)*2	Sin LVR		12,4 ~ 12,7 V (24,8 ~ 25,4 V)		
Corte descarga profunda LVD*2	Sin LVD		11,2 ~ 11,6 V (22,1 ~ 23,2 V)		
Temperatura Oper	-25°C ... +50°C				
Terminal Ø	6 mm²				
Protección	IP32				
Peso	Aprox. 150 g				
Tamaño	145 x 97,15 x 23,83 mm				

\*1) = No tiene LVD (corte por descarga profunda), el consumo conéctelo directamente a la batería.

\*2) = El valor menor para la corriente nominal, el mayor para la menor corriente.

## Anexo 2: Software del Nodo Sensor

```
//Añadimos Libreria XBee
#include<LiquidCrystal.h>
#include <XBee.h>
#include <SoftwareSerial.h>
#include "DHT.h" //cargamos la librería DHT
////////////////////////////////////

#include <stdlib.h>
#include "SIM900.h"
#include "sms.h"
MSGSM sms;

#define B0 9
#define B1 8
#define B2 12
#define B3 13
#define B4 A4
#define B5 A5

int dipPins[] = {B0, B1, B2, B3, B4, B5};

boolean started=false;
////////////////////////////////////

////////////////////////////////////

int valor_dip_switch = 0;
int value=0;
int value1=0;
int value2=0;
int pin=1;
char value_str[200];
char value_str1[200];
char value_str2[200];
////////////////////////////////////

SoftwareSerial mySerial1(2, 3);///2,3
////////////////////////////////////
////////////////////////////////////

//#define temperatura A0
#define sensor A1
#define DHTPIN A0 //Seleccionamos el pin en el
que se //conectará el sensor
#define DHTTYPE DHT11 //Se selecciona el DHT11
(hay //otros DHT)

DHT dht(DHTPIN, DHTTYPE); //Se inicia una
variable que será usada por Arduino para
comunicarse con el sensor
//LiquidCrystal lcd(11, 10, 7, 6, 5, 4);
LiquidCrystal lcd(11, 10, 4, 5, 6, 7);
////////////////////////////////////

//Definimos DH_Coordinador
#define DH_COORDINADOR 0x0013A200
//Definimos DL_Coordinador
#define DL_COORDINADOR 0x4167295F

//Declaración de constantes generales

const unsigned int BAUD_RATE = 57600;
const float ALIMENTACION = 5.0;

//Declaracion de estructuras de datos

//Temperatura en 4 bytes
union u_tag1
{
    byte t[4];
    float ftemp;
} datos1;

//Humedad en 4 bytes
union u_tag2
{
    byte h[4];
    float fhum;
} datos2;

//Posible tercer dato en 4 bytes
union u_tag3
{
    byte d3[4];
    float dd3;
} datos3;

//Declaracion de constantes para ejecucion de
programa
int i = 0;
```



```

void loop()
{
    //Obtenemos temperatura y la adjuntamos a la
    estructura
    datos1.ftemp = get_temp();
    //Obtenemos humedad y la adjuntamos a la
    estructura
    datos2.fhum = get_hum();
    //Obtenemos valor de dip_switch y la adjuntamos a
    la estructura
    datos3.dd3 = get_dipswitch();
    //Comenzamos a rellenar la trama de datos
    datos[0] = id;
    //Asignamos valor de nodo padre (en este caso
    nodo principal)
    datos[1] = 0;
    //Asignamos valor de temperatura a trama de datos
    for (i = 0; i < 4; i++)
    {
        datos[i + 2] = datos1.t[i];
    }
    //Asignamos valor de humedad a trama de datos
    for (i = 0; i < 4; i++)
    {
        datos[i + 6] = datos2.h[i];
    }
    //Asignamos valor de dip_switch a trama de datos
    for (i = 0; i < 4; i++)
    {
        datos[i + 10] = datos3.d3[i];
    }
    //Serial.println(datos2.h[i]);
    //Serial.println(valor_dip_switch);
    //Serial.println(datos3.d3[i]);
    //Serial.println(datos3.d3[i]);
    //Enviamos trama de datos
    xbee.send(zbTx);

    //Encendemos led de envio de datos
    //parpadeo_led(LED_SENALIZADOR, 1, 100);

    // Luego de enviar solicitud Tx, esperamos una
    respuesta de estado de transmisión

    // esperamos hasta medio segundo por la respuesta
    de estado
    if (xbee.readPacket(500))
    {
        // Se obtuvo una respuesta!
        //Serial.println("Enviado");
        // Debe haber un valor de znet tx status
        if (xbee.getResponse().getApiId() ==
        ZB_TX_STATUS_RESPONSE)
        {
            xbee.getResponse().getZBTxStatusResponse(txStatus
            );

            // Buscamos el estado de la entrega, el quinto
            byte
            if (txStatus.getDeliveryStatus() == SUCCESS)
            {
                // Envio correcto.
                // parpadeo_led(LED_SENALIZADOR, 5, 50);
                //Serial.println("Recibido");
            } else
            {
                // La XBee de destino no recibió el dato. ¿Se
                encuentra encendida?
                // parpadeo_led(LED_ERROR, 3, 500);
            }
        }
        else
        {
            //Serial.println("No se envió correctamente");
        }
        //////////////////////////////////////
        int val1 = datos1.ftemp;
        int val = datos2.fhum;
        int val2 = datos3.dd3;

        value2 = val2;
        value = val;
        value1 = val1;

        //////////////////////////////////////
        if (datos2.fhum > 20 || datos1.ftemp < 5)
        {

```

```

    SendMessage();
}
//lcd.clear();
//delay(1000);
}

//Función que lee el sensor de temperatura y
devuelve el dato en grados.
const float get_temp()
{
    ////////////////////////////////////////////////////
    int t = dht.readTemperature(); //Lee la temperatura
    ////////////////////////////////////////////////////
    //sensors.requestTemperatures(); //Se envía el
comando para leer la temperatura
    //float temp = sensors.getTempCByIndex(0); //Se
obtiene la temperatura en °C
    ////////////////////////////////////////////////////
    lcd.setCursor(13, 1);
    lcd.print(t);
    //Serial.println(t);
    return (t);
}

//Función que lee el sensor de humedad
const float get_hum()
{
    // float h = analogRead(humedad);
    int valorHumedad = map(analogRead(sensor), 0,
1023, 100, 0);
    int h = valorHumedad;
    lcd.setCursor(13, 0);
    lcd.print(h);
    //Serial.println(h);
    return (h);
}

//Funcion que lee el dipswitch
const float get_dipswitch()
{
    unsigned int l;
    unsigned int base=1;
    int valor_dip_switch=0;
    int lectura=0;
    for (l=0;l<6;l++)
    {
        // valor_dip_switch +=
digitalRead(dipPins[l])*base;
        lectura=digitalRead(dipPins[l]);
        if(lectura==HIGH)
        {
            valor_dip_switch += base;
        }
        base = base*2;
    }
    ////////////////////////////////////////////////////
    lcd.setCursor(1,1);
    lcd.print(valor_dip_switch);
    //Serial.println(valor_dip_switch);
    return (valor_dip_switch);
}

//Función que controla el parpadeo de led
void parpadeo_led(int pin, int times, int wait)
{
    for (int i = 0; i < times; i++)
    {
        digitalWrite(pin, HIGH);
        delay(wait); digitalWrite(pin, LOW);
        if (i + 1 < times)
        {
            delay(wait);
        }
    }
}

//Función para obtener id de nodo
void obtener_ID()
{
    byte comando_NI[] = {'N', 'I'};
    AtCommandRequest atRequest =
AtCommandRequest(comando_NI);
    AtCommandResponse atResponse =
AtCommandResponse();

    //Se obtiene el ID del nodo mediante el comando
AT NI
    xbee.send(atRequest);
}

```

```

//Esperamos hasta 5 segundos para la respuesta
del estado
if (xbee.readPacket(5000))
{
  //Hay una respuesta
  //Deberia ser una respuesta al comando AT
  if (xbee.getResponse().getApiId() ==
AT_COMMAND_RESPONSE)
  {
xbee.getResponse().getAtCommandResponse(atResp
onse);
  //Si el valor de la respuesta at es Ok
  if (atResponse.isOk())
  {
    if (atResponse.getValueLength() > 0)
    {
      //Calculamos el valor de id del nodo
      id = atResponse.getValue()[0] - '0';
    }
  }
}
}
}
}

```

```

////////////////////funciones
void SendTextMessage()
{
  lcd.setCursor(0, 0);
  lcd.print("Sistema Activo");
  lcd.setCursor(0, 1);
  lcd.print("Enviando Alerta");

  sms.SendSMS("+593990318469", "VALVULA
ACTIVA; ID_NODO:");
  itoa(value2,value_str2,10);
  sms.SendSMS("+593990318469",value_str2);
  //////////////////////////////////////
  sms.SendSMS("+593990318469", "(%)Humedad:");
  itoa(value,value_str,10);
  sms.SendSMS("+593990318469",value_str);
  //////////////////////////////////////
  sms.SendSMS("+593990318469", "Temperatura:");
  itoa(value1,value_str1,10);
  sms.SendSMS("+593990318469",value_str1);
  //if (sms.SendSMS("+593990318469", "val"))
  // Serial.println("\nSMS sent OK");
}

```

### Anexo 3: Software del Nodo Actuador

```
/Añadimos Libreria XBee
#include <XBee.h>
#include <SoftwareSerial.h>

////////////////////////////////////
//Definimos DH_Coordinador
#define DH_COORDINADOR 0x0013A200

//Definimos DL_Coordinador
#define DL_COORDINADOR 0x4167295F

//Declaración de constantes generales para
terminales
#define bomba      3

//Definiciones RED
#define NUM_NODOS 3
#define NUM_ROUTERS 2

//Declaración de constantes generales para
funcionamiento de programa
const unsigned int LED_SENALIZADOR = 13;
const unsigned int LED_ERROR = 12;
const unsigned int BAUD_RATE = 57600;
const float ALIMENTACION = 5.0;

//Declaración de variables generales para
funcionamiento de programa

int i = 0;
int origen = 0;
volatile int pulsos; // Cantidad de pulsos del sensor.
Como se usa dentro de una interrupcion debe ser
volatile
unsigned int litrosPorHora; //Calculated
litros/hora
unsigned int litrosPorMinuto; //Calculamos
litros/minuto
unsigned char sensorDeFlujo = 2; // Pin al que esta
conectado el sensor
unsigned long tiempoAnterior; // Para calcular el
tiempo

unsigned long pulsosAcumulados; // Pulsos
acumulados
float litros; // Litros acumulados
//Byte para detectar id de nodo actual
byte id = 0xFF;

//Declaracion de estructuras de datos
//Temperatura en 4 bytes
union u_tag1
{
    byte f1[4];
    float ff1;
} datos1;

//Humedad en 4 bytes
union u_tag2
{
    byte f2[4];
    float ff2;
} datos2;

//Posible tercer dato en 4 bytes
union u_tag3
{
    byte d3[4];
    float dd3;
} datos3;

//Estructura de datos para información de nodo
typedef struct
{
    byte padre;
    byte timeout;
    float grupo1;
    float grupo2;
    float grupo3;
} Nodo;

Nodo nodo[NUM_NODOS];

//Estructura de payload (datos a recibir)
```



```

// Payload conformado por:
// Byte id nodo, Byte padre nodo, Bytes temperatura,
// Bytes humedad, Bytes grupo 3 (posible).
// En total 14 bytes.
byte buffer[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//Estructura de payload (datos a enviar)
// Payload conformado por:
// Byte id nodo, Byte padre nodo, Bytes temperatura,
// Bytes humedad, Bytes grupo 3 (posible).
// En total 14 bytes.
byte datos[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//Inicialización parámetros XBee
XBee xbee = XBee();
//Creación objeto Xbee
XBeeAddress64 direccion64 = XBeeAddress64(
  DH_COORDINADOR, DL_COORDINADOR); //
Dirección XBee 64bits
XBeeResponse response = XBeeResponse();
//Respuesta de XBee
ZBRxResponse rx = ZBRxResponse();
//Respuesta de XBee en ZNet
ModemStatusResponse msr =
ModemStatusResponse(); //Estado
de Respuesta de Módem
ZBTxRequest zbTx = ZBTxRequest(direccion64,
  datos, sizeof(datos)); //Requerimiento de
transmisión XBee
ZBTxStatusResponse txStatus =
ZBTxStatusResponse(); //Dato de
estado de solicitud de transmisión XBee

//Puerto serie para XBee si no tenemos Serial1
SoftwareSerial XBeeSerial(4, 5); // RX, TX

//RX ARDUINO --- DOUT XBEE (PIN 2)
//TX ARDUINO --- DIN XBEE (PIN 3)

void flujo () // Funcion de interrupcion
{
  pulsos++; // Simplemente sumar el numero de
  pulsos
}

void setup()
{
  pinMode(sensorDeFlujo, INPUT);
  attachInterrupt(0, flujo, RISING); // Setup Interrupt
  interrupts(); // Habilitar interrupciones
  tiempoAnterior = millis();
  pinMode(13, OUTPUT);
  pinMode(bomba, OUTPUT);
  pinMode(LED_SENALIZADOR, OUTPUT);
  pinMode(LED_ERROR, OUTPUT);
  Serial.begin(9600);
  //Serial 1 para XBee
  //Serial1.begin(BAUD_RATE);
  XBeeSerial.begin(BAUD_RATE);
  //Establecemos XBee
  //xbee.setSerial(Serial1);
  xbee.setSerial(XBeeSerial);
  //Obtenemos ID del Nodo
  obtener_ID();
  //Serial.print("Id Nodo: ");
  //Serial.print(id);
  //Serial.println();
}

void loop()
{
  //Medimos caudal
  medir_caudal();
  //Esperamos a recibir datos desde el Coordinador
  recibir_datos();
  //Generamos trama de datos
  generacion_trama();
  //Enviamos trama de datos
  xbee.send(zbTx);
  //Comprobamos envio de datos generados
  anteriormente
  comprobar_envio();
  //delay(1000);
}

//Función que controla el parpadeo de led

```

```

void parpadeo_led(int pin, int times, int wait)
{
  for (int i = 0; i < times; i++)
  {
    digitalWrite(pin, HIGH);
    delay(wait); digitalWrite(pin, LOW);
    if (i + 1 < times) {
      delay(wait);
    }
  }
}

//Función para obtener id de nodo
void obtener_ID()
{
  byte comando_NI[] = {'N', 'I'};
  AtCommandRequest atRequest =
  AtCommandRequest(comando_NI);
  AtCommandResponse atResponse =
  AtCommandResponse();
  //Se obtiene el ID del nodo mediante el comando
  AT NI
  xbee.send(atRequest);
  //Esperamos hasta 5 segundos para la respuesta
  del estado
  if (xbee.readPacket(5000))
  {
    //Hay una respuesta
    //Deberia ser una respuesta al comando AT
    if (xbee.getResponse().getApiId() ==
  AT_COMMAND_RESPONSE)
    {
      xbee.getResponse().getAtCommandResponse(atResp
onse);
      //Si el valor de la respuesta at es Ok
      if (atResponse.isOk())
      {
        if (atResponse.getValueLength() > 0)
        {
          //Calculamos el valor de id del nodo
          id = atResponse.getValue()[0] - '0';
        }
      }
    }
  }
}

}

void generacion_trama()
{
  //Obtenemos temperatura y la adjuntamos a la
estructura
  datos1.ff1 = litrosPorMinuto;
  //Obtenemos humedad y la adjuntamos a la
estructura
  datos2.ff2 = litrosPorMinuto;
  //Comenzamos a rellenar la trama de datos
  datos[0] = id;
  //Asignamos valor de nodo padre (en este caso
nodo principal)
  datos[1] = 0;
  //Asignamos valor de temperatura a trama de datos
  for (i = 0; i < 4; i++)
  {
    datos[i + 2] = datos1.f1[i];
  }
  //Asignamos valor de humedad a trama de datos
  for (i = 0; i < 4; i++)
  {
    datos[i + 6] = datos2.f2[i];
  }
  //Rellenamos tercera sección de datos.
  for (i = 0; i < 4; i++)
  {
    datos[i + 10] = datos3.d3[i];
  }
}

}

void comprobar_envio()
{
  //Encendemos led de envio de datos
  parpadeo_led(LED_SENALIZADOR, 1, 100);

  // Luego de enviar solicitud Tx, esperamos una
respuesta de estado de transmisión

```

```

// esperamos hasta medio segundo por la respuesta
de estado
if (xbee.readPacket(500))
{
// Se obtuvo una respuesta!
//Serial.println("Enviado");
// Debe haber un valor de znet tx status
if (xbee.getResponse().getApiId() ==
ZB_TX_STATUS_RESPONSE)
{
xbee.getResponse().getZBTxStatusResponse(txStatus
);

// Buscamos el estado de la entrega, el quinto
byte
if (txStatus.getDeliveryStatus() == SUCCESS)
{
// Envio correcto.
parpadeo_led(LED_SENALIZADOR, 5, 50);
//Serial.println("Recibido");
}
else
{
// La XBee de destino no recibió el dato. ¿Se
encuentra encendida?
parpadeo_led(LED_ERROR, 3, 500);
}
}
else
{
//Serial.println("No se envió correctamente");
}
}

void recibir_datos()
{
//Esperamos hasta 500 milisegundos para
determinar si se ha recibido
//un mensaje en el XBee Router
xbee.readPacket(500);
if (xbee.getResponse().isAvailable())
{

```

```

//Se obtuvo una respuesta
//Serial.println("Recibido");
//Debe haber un valor de znet en rx status
if (xbee.getResponse().getApiId() ==
ZB_RX_RESPONSE)
{
xbee.getResponse().getZBRxResponse(rx);
if (rx.getOption() ==
ZB_PACKET_ACKNOWLEDGED)
{
//El nodo que envía datos recibió un ACK (fue
reconocido)
parpadeo_led(LED_SENALIZADOR, 10, 10);
}
else
{
//El nodo que envía datos no recibió un ACK
(no fue reconocido)
parpadeo_led(LED_ERROR, 2, 20);
}
//Llenamos array buffer con datos provenientes
en puerto serial XBee
for (i = 0; i < 10; i++)
{
buffer[i] = rx.getData(i);
}
//Determinamos ID Nodo Origen de Informacion
origen = buffer[0];
//Asignamos timeout a nodo actual
nodo[origen].timeout = NUM_NODOS;
//Asignamos valor de nodo padre a nodo actual
nodo[origen].padre = buffer[1];
//Llenamos estructura de datos con demás datos
provenientes de Router
for (i = 0; i < 4; i++)
{
datos1.ff1[i] = buffer[i + 2];
}
nodo[origen].grupo1 = datos1.ff1;
for (i = 0; i < 4; i++)
{
datos2.ff2[i] = buffer[i + 6];
}
nodo[origen].grupo2 = datos2.ff2;
for (i = 0; i < 4; i++)

```

```

{
    datos3.d3[i] = buffer[i + 10];
}
nodo[origen].grupo3 = datos3.dd3;

//Encendemos o apagamos electrovalvula
if (nodo[origen].grupo1 == 1000)
{
    digitalWrite(bomba, HIGH);
}
else
{
    digitalWrite(bomba, LOW);
}
//Si se encuentra otro tipo de mensaje
//(Control, Error, Desconocido, etc)
//El LED_ERROR parpadea para indicarlo
else if (xbee.getResponse().getApiId() ==
MODEM_STATUS_RESPONSE)
{
xbee.getResponse().getModemStatusResponse(msr);
if (msr.getStatus() == ASSOCIATED)
{
    parpadeo_led(LED_ERROR, 10, 10);
}
else if (msr.getStatus() == DISASSOCIATED)
{
    parpadeo_led(LED_ERROR, 10, 10);
}
else
{
    parpadeo_led(LED_ERROR, 5, 10);
}
}
else
{
    parpadeo_led(LED_ERROR, 1, 25);
}
}

void medir_caudal()
{
    // Cada segundo calcular e imprimir Litros/seg
    if ( millis() - tiempoAnterior > 1000)
    {
        tiempoAnterior = millis(); // Updates cloopTime
        // Pulse frequency (Hz) = 6.67 Q, Q is flow rate in
        L/min. (Results in +/- 3% range)
        // Q = frecuencia / 6.67 (L/min)
        // Q = (frecuencia * 60) / 6.67 (L/hora)
        pulsosAcumulados += pulsos;
        litrosPorMinuto = (pulsos/6.67); // (Pulse
        frequency ) / 7.5Q = flow rate in L/min
        //litrosPorHora = (litrosPorMinuto* 60);
        //pulsos = 0; // Reset Counter
        //Serial.print(" L/hour: ");
        //Serial.print(litrosPorHora, DEC); // Print
        litres/hour
        //Serial.println();
        //Serial.print(" Pulsos totales: ");
        //Serial.print(pulsosAcumulados);
        //Serial.println();
        //litros = pulsosAcumulados * 1.0 / 400; //Cada
        400 pulsos = 1 litro
        //Serial.print(" Litros: ");
        //Serial.print(litros);
        //Serial.println();
        //Serial.println(litrosPorMinuto);
    }
}
}

```

## Anexo 4: Software del Nodo Control

```
//Añadimos Libreria XBee

#include <XBee.h>
#include <SoftwareSerial.h>
#include <SD.h>
#include <Wire.h>
#include <SPI.h>
#include <RTCLib.h>
#define valvula 6
File myFile;
RTC_DS1307 RTC;
////////////////////////////////////
////////////////////////////////////
SoftwareSerial mySerial(4, 5);
SoftwareSerial mySerial1(2, 3);////2,3
////////////////////////////////////
//Definimos DH_Electrovalvula
#define DH_ELECTROVALVULA 0x0013A200

//Definimos DL_Electrovalvula
#define DL_ELECTROVALVULA 0x40CAA949

//Definiciones RED
#define NUM_NODOS 3
#define NUM_ROUTERS 2

//Declaración de constantes generales
//const unsigned int LED_SENALIZADOR = 13;
//const unsigned int LED_ERROR = 12;
const unsigned int BAUD_RATE = 57600;
const float ALIMENTACION = 5.0;
uint8_t comandoON[] = { 0x05 };
uint8_t comandoOFF[] = { 0x04 };

//Constantes principales
int j = 0;
byte origen = 0 ;
byte padre = 0 ;
int medida_caudal = 0;
int valor_entero = 0;

int valor_dip_switch = 0;

//Estructura de payload (datos a enviar)
// Payload conformado por:
// Byte id nodo, Byte padre nodo, Bytes temperatura,
// Bytes humedad, Bytes grupo 3
// En total 14 bytes.
byte buffer[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//Trama para envio de datos para encendido de
electrovalvula
byte datos[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

//Creamos definiciones para XBee
XBee xbee = XBee(); //Creación objeto Xbee
XBeeAddress64 direccion64 = XBeeAddress64(
DH_ELECTROVALVULA,
DL_ELECTROVALVULA); // Dirección XBee 64bits
Electrovalvula
XBeeResponse response = XBeeResponse();
//Respuesta de XBee
ZBRxResponse rx = ZBRxResponse(); //Respuesta
de XBee en ZNet
ModemStatusResponse msr =
ModemStatusResponse(); //Estado de Respuesta de
Módem
ZBTxRequest comando_remoto =
ZBTxRequest(direccion64, datos, sizeof(datos));
//Requerimiento de transmisión XBee
ZBTxStatusResponse txStatus =
ZBTxStatusResponse(); //Dato de estado de solicitud
de transmisión XBee

//Creación de estructura de datos
union u_tag1
{
byte t[4];
float ftemp;
} datos1;
```

```

union u_tag2
{
  byte h[4];
  float fhum;
} datos2;

//Posible tercer dato en 4 bytes
union u_tag3
{
  byte d3[4];
  float dd3;
} datos3;

typedef struct
{
  byte padre;
  byte timeout;
  float grupo1;
  float grupo2;
  float grupo3;
} Nodo;

Nodo nodo[NUM_NODOS];

//Byte para detectar id de nodo actual
byte id = 0xFF;

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

void setup()
{
  //////////////////////////////////////
  //-----RTC SETUP -----
  Wire.begin();
  RTC.begin();
  if (! RTC.isrunning())
  {
    //Serial.println("El RTC no esta funcionando!");
    //Cogemos como fecha y hora aquella en la que el
    programa fue compilado.
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  //Cogemos la fecha y la hora del reloj.
  DateTime now = RTC.now();

  //Cogemos la hora en que compilamos el programa.
  DateTime compiled = DateTime(__DATE__,
  __TIME__);

  if (now.unixtime() < compiled.unixtime()) {
    // Serial.println("La hora del RTC es mas antigua
    que la compilacion! Actualizando...");
    RTC.adjust(DateTime(__DATE__, __TIME__));
  }
  //////////////////////////////////////
  pinMode(valvula,OUTPUT);
  Serial.begin(9600);
  if (SD.begin())
  {
    //Serial.println("SD exitosa");
  } else
  {
    //Serial.println("SD fallida");
    return;
  }

  // Create/Open file
  myFile = SD.open("Mora.txt", FILE_WRITE);

  // if the file opened okay, write to it:
  if (myFile)
  {
    //Serial.println("Archivo nuevo, Escribiendo
    encabezado(fila 1)");

    myFile.println("Id_Nodo,Dia,Mes,Year,Hora,Minut
    os,Temperatura,Humedad,Caudal");
    myFile.close(); // close the file
    //Serial.println("Done.");
  }
  // if the file didn't open, print an error:
  else
  {
    //Serial.println("error opening test.txt");
  }
  // Reading the file
  myFile = SD.open("Mora.txt");
  if (myFile)
  {

```



```

if (Serial.available() >= 21)
{
  if (Serial.read() == 0x7E) ////(0x17)
  {
    for (int j = 1; j < 24; j++)
    {
      // Serial.print(j);
      // Serial.print(": ");
      byte discardByte = Serial.read();
      // Serial.println(discardByte, HEX);
    }
  }
}

if (nodo[1].grupo2 > 20 || nodo[1].grupo1 < 5)
{
  digitalWrite(valvula,HIGH);
  encender_electrovalvula();
  //Serial.print("Encendida");
}
else
{
  digitalWrite(valvula,LOW);
  apagar_electrovalvula();
  //Serial.print("Apagada");
}
//Serial.println();
valor_dip_switch = nodo[1].grupo3;
medida_caudal = nodo[2].grupo1;
//Serial.print("Medida Caudal: ");
//Serial.print(medida_caudal);
//Serial.println();

actualizar_nodos();

myFile = SD.open("Mora.txt",
FILE_WRITE);//abrimos el archivo
if (myFile)
{
  DateTime now = RTC.now();

  myFile.print(valor_dip_switch);
  myFile.print(",");
  //Serial.println(nodo[1].grupo3);
  //Serial.print('/');

  //int dia = now.day();
  myFile.print(now.day());
  //Serial.print(dia);
  myFile.print(",");
  //Serial.print(",");

  //Serial.print(now.month(), DEC);
  //Serial.print('/');
  //int mes = now.month();
  myFile.print(now.month());
  //Serial.print(mes);
  myFile.print(",");
  //Serial.print(",");

  //Serial.print(now.year(), DEC);
  // Serial.print('/');
  // int ano = now.year();
  myFile.print(now.year());
  //Serial.print(ano);
  myFile.print(",");
  //Serial.print(",");

  //Serial.print(now.hour(), DEC);
  //Serial.print(':');
  //int hora = now.hour();
  myFile.print(now.hour());
  //Serial.print(hora);
  myFile.print(",");
  //Serial.print(",");

  //Serial.print(now.minute(), DEC);
  //Serial.print(':');
  //int minutos = now.minute();
  myFile.print(now.minute());
  //Serial.print(minutos);
  myFile.print(",");
  //Serial.print(",");

  myFile.print(nodo[1].grupo1);
  //Serial.print(nodo[1].grupo1);
  myFile.print(",");
  //Serial.print(",");

  myFile.print(nodo[1].grupo2);
  //Serial.print(nodo[1].grupo2);
}

```



```

myFile.print(",");
//Serial.print(",");

myFile.println(medida_caudal);
//Serial.print(medida_caudal);
//myFile.print(",");
//Serial.print(",");

myFile.close();

} else
{
//Serial.println("Error al abrir el archivo");
}

////////////////////////////////////
Serial.print(" ");
Serial.print(valor_dip_switch);
Serial.print(" ");
Serial.print(nodo[1].grupo1);
Serial.print(" ");
Serial.print(nodo[1].grupo2);
Serial.print(" ");
Serial.println(medida_caudal);

//delay(2000);

////////////////////////////////////
/
}

void actualizar_nodos()
{
for (j = 0; j < NUM_NODOS; j++)
{
if (nodo[j].timeout > 0)
{
nodo[j].timeout -= 1;
}
}
}

void envio_comando_remoto(int valor_orden)
{
if (valor_orden == 1)
{
valor_entero = 1000;
}
else
{
valor_entero = 0;
}
datos1.ftemp = valor_entero;
datos2.fhum = valor_entero;
datos[0] = id;

datos[1] = 0;

for (j = 0; j < 4; j++)
{
datos[j + 2] = datos1.t[j];
}

for (j = 0; j < 4; j++)
{
datos[j + 6] = datos2.h[j];
}

for (j = 0; j < 4; j++)
{
datos[j + 10] = datos3.d3[j];
}

xbee.send(comando_remoto);
if (xbee.readPacket(500))
{
// Se obtuvo una respuesta!
//Serial.println("Enviado");
// Debe haber un valor de znet tx status
if (xbee.getResponse().getApiId() ==
ZB_TX_STATUS_RESPONSE)
{
xbee.getResponse().getZBTxStatusResponse(txStatus);
if (txStatus.getDeliveryStatus() == SUCCESS)
{
} else
{
}
}
}
}
}

```

```

}
else
{
}
}

void encender_electrovalvula()
{
    envio_comando_remoto(1);
}

void apagar_electrovalvula()
{
    envio_comando_remoto(2);
}

/////////////////////////////////////////FUNCIONES
void obtener_ID()
{
    byte comando_NI[] = {'N', 'T'};
    AtCommandRequest    atRequest    =
    AtCommandRequest(comando_NI);
    AtCommandResponse    atResponse    =
    AtCommandResponse();

    xbee.send(atRequest);

    if (xbee.readPacket(5000))
    {
        if (xbee.getResponse().getApiId() ==
        AT_COMMAND_RESPONSE)
        {
            xbee.getResponse().getAtCommandResponse(atResponse);
            if (atResponse.isOk())
            {
                if (atResponse.getValueLength() > 0)
                {
                    id = atResponse.getValue()[0] - '0';
                }
            }
        }
    }
}

```