

Universidad Autónoma de Bucaramanga

Prototipo de sistema de monitorización y riego automático que mide la humedad y la temperatura para evaluar la vegetación

Un trabajo de grado entregado a
la Facultad de Ingeniería

Directores
René Alejandro Lobo Quintero.
Diana Teresa Parra Sánchez.

Por
Frank Wilmar Santander González

Bucaramanga, Santander

31 de octubre de 2016.

Agradecimientos

Este proyecto de grado está dedicado a mi profesor René Lobo, por su dedicación y enseñanza durante el desarrollo de proyectos para garantizar resultados óptimos, a mi profesora Diana Parra por enseñarme a implementar una mejor metodología para desarrollar mis proyectos y obtener una mejor educación, a Jorge Andrick Parra por su optimismo y enseñanza de la ética del ingeniero de sistemas, y a José Daniel Cabrera por introducirme al maravilloso mundo de la investigación, por enseñarme a defenderme en este, y por cambiar mi forma de ver las cosas al enseñarme sobre cómo ser un mejor profesional.

Agradezco a mis compañeros porque han estado conmigo en las buenas y en las malas situaciones, aportándome sus ideas para elaborar mejores proyectos de investigación y para formarme como un buen profesional.

Agradezco a mis padres por apoyarme siempre en mi vida, en la carrera que curso, y también en este proyecto, al aportar financieramente para conseguir los materiales usados en este proyecto.

Los ideales que iluminan mi camino y una y otra vez
me han dado coraje para enfrentar la vida con alegría
han sido: la amabilidad, la belleza y la verdad.

— Albert Einstein.

Contenido

Resumen.....	x
Abstract.....	xi
Planteamiento del problema.....	xii
1. Objetivos.....	1
1.1. Objetivo general.....	1
1.2. Objetivos específicos.....	1
2. Revisión de la literatura.....	2
2.1. Estado del arte.....	2
2.2. Marco teórico y/o conceptual.....	1
2.2.1. Arduino.....	1
2.2.2. Automatización.....	1
2.2.3. Código QR (Quick Response Code).....	1
2.2.4. Comunicación de campo cercano o NFC (Near Field Communication).....	1
2.2.5. Fritzing.....	1
2.2.6. <i>Hardware</i> libre.....	2
2.2.6.1. Tipos principales de <i>hardware</i> libre.....	3
2.2.6.2. Estándares de conexión de módulos y sensores.....	4
2.2.6.2.1. I ² C.....	4
2.2.6.2.2. UART.....	4
2.2.7. Internet de las cosas.....	4
2.2.8. Lux.....	4
2.2.9. Monitorización.....	5
2.2.10. Red de sensores inalámbricos.....	5
2.2.11. <i>Software</i> de código abierto.....	5
2.2.11.1. Tipos de <i>software</i> de código abierto seleccionado para diseñar y programar <i>sketches</i> de Arduino.....	5
2.2.12. WiFi.....	6
2.3. Marco legal.....	7
2.3.1. Arduino.....	7
2.3.1.1. Uso de la marca Arduino.....	7
2.3.1.2. Uso del entorno de programación de Arduino.....	7
2.3.1.3. Uso del <i>hardware</i> y del logotipo.....	7
2.3.2. Uso de Fritzing.....	8

2.3.3.	Tipos de licencias usadas	8
2.3.3.1.	CC-BY-SA	8
2.3.3.2.	CC-BY-ND.....	8
2.3.3.3.	GPL	8
2.3.3.4.	LGPL	9
3.	Cronograma de actividades.....	10
4.	Presupuesto	11
4.1.	Versión 1 de SIMORIA (SIMORIA 1P).....	11
4.2.	Versión 2 de SIMORIA (SIMORIA 1P2).....	13
4.3.	Adaptación a SIMORIA.....	14
5.	Metodología	15
5.1.	Elaboración del estado del arte.....	15
5.2.	Diseño del sistema de monitorización y riego automático.....	15
5.3.	Implementación del diseño realizado	16
5.4.	Pruebas funcionales y análisis de datos.....	17
6.	Avances de SIMORIA	18
6.1.	Elaboración del estado del arte.....	18
6.1.1.	Búsqueda de palabras clave.....	18
6.1.2.	Revisión de la literatura.....	18
6.1.3.	Organización de los recursos encontrados	18
6.2.	Diseño del sistema de monitorización y riego automático.....	18
6.2.1.	Diseño de los planos del prototipo	18
6.2.2.	Análisis del <i>hardware</i> y la calibración de sensores	22
6.2.2.1.	Sensor de humedad relativa y temperatura DHT22	22
6.2.2.2.	Sensor de iluminancia TSL2591	22
6.2.2.3.	Sensor de humedad en el suelo capacitivo	22
6.2.3.	Ensamblaje del <i>hardware</i> y configuración de la conectividad.....	25
6.2.3.1.	Paso 1	25
6.2.3.2.	Paso 2	26
6.2.3.3.	Paso 3	26
6.2.3.4.	Paso 5	27
6.2.3.5.	Paso 6	28
6.2.3.6.	Paso 8	30
6.3.	Implementación del diseño realizado	31
6.3.1.	Selección del servidor web para almacenamiento de datos.....	31

6.3.1.1.	Uso del servidor web seleccionado	33
6.3.2.	Selección de la aplicación móvil y conexión con el servidor.....	36
6.3.2.1.	Selección de la aplicación móvil.....	36
6.3.2.2.	Conexión con el servidor web	38
6.3.2.3.	Uso de la aplicación móvil seleccionada.....	40
6.4.	Pruebas funcionales y análisis de los datos	43
6.4.1.	Prueba funcional del sistema.....	43
6.4.2.	Análisis de los datos obtenidos	44
7.	Trabajo adicional	45
7.1.	Versiones de SIMORIA	45
7.1.1.	Versión 1P (SIMORIA 1P)	45
7.1.2.	Versión 1P2 (SIMORIA 1P2)	45
7.1.2.1.	Desarrollo de SIMORIA 1P2	45
8.	Propuesta futura	53
8.1.	Versión oficial de SIMORIA	53
8.2.	Alimentación de SIMORIA por panel solar.....	54
8.3.	Análisis de datos avanzado	54
9.	Conclusiones	55
10.	Referencias.....	56

Tablas

Tabla 1. Criterios de búsqueda utilizados para encontrar fuentes oficiales de información.	3
Tabla 2. Cantidad de referencias recuperadas de fuentes oficiales.	3
Tabla 3. Primera parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas.	1
Tabla 4. Segunda parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas.	2
Tabla 5. Tercera parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas.	3
Tabla 6. Cuarta parte del listado de autores, títulos de publicaciones y fuentes donde fueron encontradas.	4
Tabla 7. Tipos principales de hardware libre para elaborar proyectos electrónicos.	3
Tabla 8. Software utilizado en el desarrollo del prototipo de sistema de monitorización y riego automático.	6
Tabla 9. Cronograma de actividades.	10
Tabla 10. Listado de materiales utilizados en el desarrollo la versión 1 de SIMORIA (prototipo en Arduino).	11
Tabla 11. Segunda parte del listado de materiales utilizados en el desarrollo de la versión 1 de SIMORIA.	12
Tabla 12. Cálculo del precio total invertido para el desarrollo de la versión 1 de SIMORIA.	12
Tabla 13. Listado de materiales utilizados en el desarrollo de la versión 2 de SIMORIA.	13
Tabla 14. Listado de materiales utilizados en el desarrollo del sistema riego automático de SIMORIA (versiones 1 y 2).	14
Tabla 15. Tabla de del mínimo y del máximo valor de humedad en el suelo obtenidos por el sensor.	23
Tabla 16. Selección del servidor web (parte 1).	31
Tabla 17. Selección del servidor web (parte 2).	32
Tabla 18. Aplicaciones móviles existentes y relevantes utilizadas en Internet de las cosas.	36

Figuras

Figura 1. Metodología presentada para la elaboración del prototipo.....	15
Figura 2. Representación de los elementos utilizados en el prototipo.....	16
Figura 3. Vista previa del plano del prototipo de sistema de monitorización y riego automático.....	19
Figura 4. Esquema del plano del prototipo de sistema de monitorización y riego automático.....	20
Figura 5. Circuito del sistema de monitorización y riego automático.....	20
Figura 6. Logotipo de la comunidad de Arduino.....	21
Figura 7. Logotipo de SIMORIA.....	21
Figura 8. Código QR contenido en el logotipo de SIMORIA.....	21
Figura 9. Gráfica que determina la función de porcentaje de humedad en el suelo a partir de las lecturas del sensor.....	25
Figura 10. Vistas delantera y posterior de la tarjeta Arduino Mega2560.....	25
Figura 11. Sensor Shield V4.....	26
Figura 12. Sensor de humedad capacitivo.....	26
Figura 13. Sensor de humedad relativa y temperatura DHT22.....	27
Figura 14. Sensor de iluminancia TSL2591.....	27
Figura 15. Módulo relé.....	28
Figura 16. Módulo WiFi ESP8266MOD.....	28
Figura 17. Programación del funcionamiento de los sensores del prototipo en el entorno de programación Arduino.....	29
Figura 18. Selección de tarjetas en el entorno Arduino.....	29
Figura 19. Carcasa diseñada para el encaje del dispositivo electrónico.....	30
Figura 20. Dispositivo electrónico puesto en marcha.....	30
Figura 21. Creación de cuenta en ThingSpeak.....	33
Figura 22. Interfaz de usuario en ThingSpeak.....	34
Figura 23. Interfaz de creación de canales en ThingSpeak (primera parte).....	34
Figura 24. Interfaz de creación de canales en ThingSpeak (segunda parte).....	35
Figura 25. Interfaz de visualización de datos en ThingSpeak.....	36
Figura 26. Cuota del mercado de los dispositivos móviles y tabletas en 2016 mostrado en NetMarketShare.....	37
Figura 27. Cuota del mercado de los dispositivos móviles y tabletas en 2016 según StatCounter.....	38
Figura 28. Opción «API Keys» en ThingSpeak.....	38
Figura 29. Clave API de escritura del canal creado en ThingSpeak.....	39
Figura 30. Visualización de datos en ThingSpeak.....	40
Figura 31. ID del canal en ThingSpeak (parte subrayada).....	41
Figura 32. Clave API de lectura del canal creado en ThingSpeak.....	41
Figura 33. Ingreso de datos del canal creado en ThingSpeak.....	41
Figura 34. Visualización de datos en ThingView.....	42
Figura 35. Visualización de temperatura en ThingView.....	42
Figura 36. Vistas delantera y trasera de la tarjeta de desarrollo NodeMCU.....	46
Figura 37. <i>Motor Shield</i> para tarjetas NodeMCU.....	46
Figura 38. Sensor de humedad en el suelo.....	47

Figura 39. Sensor de humedad relativa y temperatura DHT22.	47
Figura 40. Sensor de iluminancia BH1750.	47
Figura 41. Programación de funciones de SIMORIA 1P2 en el entorno Arduino.	48
Figura 42. SIMORIA 1P2 puesto en marcha.	49
Figura 43. Interfaz de creación de proyectos en Blynk.	50
Figura 44. Interfaz del proyecto creado en Blynk.....	50
Figura 45. Valores etiquetados o <i>Labeled Values</i> en Blynk.	51
Figura 46. <i>Widget WebHook</i> en Blynk.	51
Figura 47. Botón comenzar (marcado) en el menú del proyecto creado en Blynk.....	52
Figura 48. Recepción de Datos de Blynk en ThingSpeak.	52
Figura 49. Módulo ESP8266-12E.....	53
Figura 50. Regleta hembra con forma de «L».	53
Figura 51. Baquelita de cobre usada para el diseño de circuitos electrónicos.....	54

Resumen

El Internet de las cosas es una tendencia global que permite la recopilación de la información generada por los sensores utilizados, y la almacena a través de servidores web, por el cual se realiza el posterior análisis teórico y gráfico de los datos evaluados. Actualmente, con el clima cambiante en un tiempo aleatorio, es necesario evaluar las condiciones ambientales de la vegetación, ya que, al regar la planta se usan cantidades deficientes o excesivas de agua. Como solución a la problemática, se desarrolla un prototipo de sistema de monitorización y riego automático usando una tarjeta Arduino, cuyos sensores utilizados son: de humedad en el suelo, de humedad relativa y temperatura, y de iluminación, los cuales indican que se debe regar la planta y accionan el riego automático a través de una bomba de agua conectada cuando las condiciones sean poco favorables tales como altas temperaturas, alta iluminación y poca humedad en el suelo. El prototipo es desarrollado utilizando *hardware* de código abierto. Adicionalmente, incorpora una aplicación móvil para Android que sirve para transmitir los datos generados por el *hardware*, y un servidor web que almacena los datos generados en la aplicación.

Palabras clave: Arduino, automatización, Fritzing, *hardware* libre, Internet de las cosas, lux, monitorización, red de sensores inalámbricos, *software* de código abierto, WiFi.

Abstract

Internet of things is a global tendency that enables recompilation of generated information by used sensors and stored on web servers, which subsequent theoretical and graphical analysis of evaluated data is performed. Currently, with the changing weather in a random time, it is necessary to evaluate ambient conditions of vegetation, because when people are watering a plant, deficient or excessive quantities of water are used. As a solution to the problematics, a prototype of a monitoring and automatic watering system has been developed using an Arduino board, which used sensors are: moisture, relative humidity and temperature, and illumination, that indicate automatic watering when conditions are less favourable such as: high temperatures, high illumination and low moisture. The prototype is developed using open-source hardware. In addition, a mobile app for Android is incorporated, and it is useful for transmitting generated data by the hardware, and a web server that stores generated data by the app.

Keywords: Arduino, automation, Fritzing Internet of things, lux, monitoring, open-source hardware, open-source *software*, WiFi, wireless sensor network.

Planteamiento del problema

Actualmente, con el clima cambiante (sequía y lluvias) en un tiempo aleatorio, es necesario evaluar constantemente las condiciones ambientales de la vegetación como la cantidad de agua en el suelo, la temperatura ambiental y la iluminación producida por el sol. Con plantas presentes en las casas, es común el riego de cantidades deficientes o excesivas de agua, cuyas consecuencias son: putrefacción en las raíces (por exceso de agua), «decoloración de las hojas (por deficiencia de agua)» (Gómez Rodríguez 2015, 21), o también, por las temperaturas altas y por la iluminación del sol, cuyas consecuencias podrían ser: coloración negra en las hojas (por altas temperaturas y excesiva iluminación), putrefacción en las hojas (por bajas temperaturas y deficiente iluminación), y sequedad de la planta (por la evaporación del agua en períodos de temperaturas altas).

Como una solución posible del problema, se analizarán las condiciones ambientales de una planta para determinar el momento preciso del riego y su cantidad adecuada.

Se diseña un prototipo de un sistema de monitoreo y riego automático que evaluará las condiciones ambientales de la planta como humedad en el suelo, temperatura, humedad relativa e iluminación, y que posteriormente, riegue la planta con una cantidad adecuada y en el momento preciso. Adicionalmente, se desarrollará una aplicación móvil que permitirá visualizar los datos en tiempo real.

1. Objetivos

1.1. Objetivo general

Implementar un dispositivo en Arduino que evalúe las condiciones ambientales de la vegetación a estudiar para determinar las condiciones óptimas de la misma.

1.2. Objetivos específicos

Los objetivos específicos mostrados en este proyecto son:

1. Elaborar un estado del arte sobre Internet de las cosas y sobre la implementación de sensores para Arduino en la agricultura.
2. Diseñar el sistema de monitorización y riego automático, evaluando las necesidades y características del *hardware* y *software* necesarios.
3. Implementar el sistema usando el diseño realizado.
4. Realizar pruebas funcionales y análisis de los datos obtenidos.

2. Revisión de la literatura

2.1. Estado del arte

Según Dasios *et al.* (2015, 1), una «red de monitorización es basada en el registro de parámetros ambientales como temperatura, humedad e iluminancia, y es realizada por medio de interfaces web».

Barath *et al.* (2014, 1) propusieron un «sistema de monitorización de humedad del suelo que mide en tiempo real y que muestra la información al usuario final».

Jindarat y Wuttidittachotti (2015, 284) idearon un «sistema embebido que monitoriza humedad en el aire y temperatura, y que también permite el control del flujo de aire en un gallinero por medio de un relé».

Calderón (2015, 3) propuso un «prototipo de dispositivo basado en Arduino que registre los datos de temperatura y humedad en tiempo real, usando el concepto de redes de sensores inalámbricos».

Paphitou, Constantinou y Kapitsaki (2015, 2) desarrollaron un «sistema que permite controlar bombillos y aire acondicionado mediante un relé».

A continuación, se presentan las tablas: los criterios utilizados para encontrar las fuentes consultadas en la tabla 1, el número de recursos encontrados en la tabla 2, y el listado de referencias en las tablas 3, 4, 5 y 6).

Tabla 1. Criterios de búsqueda utilizados para encontrar fuentes oficiales de información.

Criterios usados para referenciar artículos	
Palabras clave	Arduino, automatización, <i>hardware</i> libre, Internet de las cosas, lux, monitorización, redes de sensores inalámbricos, <i>software</i> de código abierto, WiFi.
Fechas de búsqueda	Enero 28 y 29 de 2016. Febrero 8 y 16 de 2016. Marzo 2, 17, 27, 29 y 30 de 2016. Abril 1, 7, 10 y 12 de 2016. Agosto 25 y 29 de 2016. Octubre 4, 5, 14 y 31 de 2016.
Criterios para incluir referencias	Rango de fecha de búsqueda: 2011 – 2016. Frases clave: «Arduino monitoring system», «Arduino monitoring».

Tabla 2. Cantidad de referencias recuperadas de fuentes oficiales.

Referencias	
Fuente	Cantidad
ACM	3
Arduino LLC	6
Arduino SRL	3
Creative Commons	2
Denso ADC	1
Friends of Fritzing	4
GitHub	3
GNU	2
IDIOSI Publication	1
IEEE	2
ITU	1
Journal of Communications	1
Net Applications	1
NXP	2
Omnigatherum	2
Oxford Dictionaries	1
Pearson Education	1
RAE	2
Routledge	1
SparkFun	1
StatCounter	1
UNAB	1
Universidad Autónoma Agraria Antonio Narro	1
Total	43

Tabla 3. Primera parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas.

Autor	Título	Fuente	Entidad o país	Año	Número de citas
Arduino LLC	<i>Función analogRead()</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino LLC	<i>Arduino Community Logo</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino LLC	<i>Frequently Asked Questions</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino LLC	<i>Trademark</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino LLC	<i>Two Arduinos Become One</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino LLC	<i>What is Arduino?</i>	Arduino LLC	Estados Unidos/Italia	2016	Desconocido.
Arduino SRL	<i>Copyright Notice</i>	Arduino SRL	Italia	2016	Desconocido.
Arduino SRL	<i>Two Arduinos Become One</i>	Arduino SRL	Italia	2016	Desconocido.
Arduino SRL	<i>What is Arduino</i>	Arduino SRL	Italia	2016	Desconocido.
Bakker, Kees <i>et al.</i>	<i>Sodaq DS3231</i>	GitHub	Estados Unidos	2016	Desconocido
Barath, Pratap <i>et al.</i>	<i>A Novel Adaptive Framework for Efficient and Effective Management of Water Supply System Using Arduino</i>	ACM	Estados Unidos	2014	0
Bruneau, Richard	<i>Fritzing Parts – First Set</i>	Omnigatherum	Canadá	2014	Desconocido.
Bruneau, Richard	<i>Fritzing Parts – Second Set</i>	Omnigatherum	Canadá	2015	Desconocido.
Calderón Calderón, Cristian René	<i>Prototipo de sistema informático para la captura ambiental en espacios cerrados y monitoreo en tiempo real. Caso de estudio en la Facultad de Ingeniería de Sistemas (UNAB)</i>	UNAB	Colombia	2015	Desconocido.

Tabla 4. Segunda parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas.

Autor	Título	Fuente	Entidad o país	Año	Número de citaciones
Chun-zi	<i>A Universal Asynchronous Receiver Transmitter Design</i>	IEEE	Estados Unidos	2011	28
Cortez, Pamela	<i>Make Your Own Fritzing Parts</i>	SparkFun	Estados Unidos	2014	Desconocido.
Creative Commons	<i>Attribution-ShareAlike 4.0 International</i>	Creative Commons	Estados Unidos	2013	Desconocido.
Creative Commons	<i>Attribution-NoDerivatives 4.0 International</i>	Creative Commons	Estados Unidos	2013	Desconocido
Dasios, Athanasios <i>et al.</i>	<i>Wireless Sensor Network Deployment for Remote Elderly Care Monitoring</i>	ACM	Estados Unidos	2015	1
De Cristofaro, John	<i>2.1 mm DC Barrel Jack</i>	GitHub	Estados Unidos	2013	Desconocido
Denso ADC	<i>Qr Code Essentials</i>	Denso ADC	Estados Unidos	2011	Desconocido
DiCola, Tony <i>et al.</i>	<i>Adafruit TSL2591 Library</i>	GitHub	Estados Unidos	2016	Desconocido.
DiCola, Tony <i>et al.</i>	<i>DHT Sensor Library</i>	GitHub	Estados Unidos	2016	Desconocido.

Tabla 5. Tercera parte del listado de autores, títulos de publicaciones y fuentes donde fueron publicadas

Autor	Título	Fuente	Entidad o país	Año	Número de citaciones
GNU	<i>GNU General Public License</i>	GNU	Estados Unidos	2016	Desconocido
GNU	<i>GNU Lesser General Public License</i>	GNU	Estados Unidos	2016	Desconocido
Herstatt, Cornelius y Ehls, Daniel.	<i>Open Source Innovation – The Phenomenon, Participant’s Behavior, Business Implications</i>	Routledge	Estados Unidos	2015	Desconocido
Howell, Brendan; van Osterbosch, Dirk; Michel Lionel	<i>DC Motor</i>	Fritzing	Estados Unidos	2015	Desconocido
ITU	<i>Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next Generation Networks</i>	International Telecommunication Union	Suiza	2012	61
Jinda, S. y Wutty, P.	<i>Smart Farm Monitoring Using Raspberry Pi and Arduino</i>	IEEE	Estados Unidos	2015	Desconocido
Net Applications	<i>Mobile/Tablet Operating System Market Share</i>	NetMarketShare	Estados Unidos	2016	Desconocido
Paphitou, Athina C; Constantinou, Stella; Kapitsaki, Georgia M.	<i>SensoMan: Remote Management of Context Sensors</i>	ACM	Estados Unidos	2015	1
NXP Semiconductors	<i>NFC Knowledge Base</i>	NXP	Países Bajos	2016	Desconocido

Tabla 6. Cuarta parte del listado de autores, títulos de publicaciones y fuentes donde fueron encontradas

Autor	Título	Fuente	Entidad o país	Año	Número de citas
NXP Semiconductors	<i>I2C-Bus Components Selection Guide</i>	NXP	Países Bajos	2013	Desconocido
Oxford University Press	<i>Automation</i>	Oxford Dictionaries	Reino Unido	2016	Desconocido
RAE	<i>Lux</i>	Diccionario de la Lengua Española	España	2016	Desconocido
RAE	<i>Monitorizar</i>	Diccionario de la Lengua Española	España	2016	Desconocido
Rajeshwari, P; Shanthini, B; Prince, Mini	<i>Hierarchical Energy Efficient Clustering Algorithm for WSN</i>	IDIOSI Publication	Emiratos Árabes Unidos	2015	Desconocido
Shao, Xing; Wang, Cui-Wang; Rao, Yuan	<i>Network Coding Aware QoS Routing for Wireless Sensor Network</i>	Journal of Communications	Estados Unidos	2015	2
StatCounter	<i>Top 8 Mobile Operating Systems from Jan to May 2016</i>	StatCounter	Irlanda	2016	Desconocido

2.2. Marco teórico y/o conceptual

2.2.1. Arduino

Es definido por Arduino como una «plataforma de diseño de prototipos de código abierto basado en *software* y *hardware* fácil de usar» (Arduino LLC 2016e; Arduino SRL 2016c).

2.2.2. Automatización

«Uso o introducción de equipo automático en una fabricación u otros procesos o instalaciones» (Oxford Unity Press 2016a).

2.2.3. Código QR (Quick Response Code)

«Código de matriz bidimensional que mantiene una gran cantidad de datos en un espacio pequeño y que se puede leer a una alta velocidad por computadoras» (Denso ADC 2016, 1-2).

2.2.4. Comunicación de campo cercano o NFC (Near Field Communication)

«Subconjunto de la identificación por radiofrecuencia que opera a 13,56 MHz y que funciona cuando dos dispositivos están cerca, y por ende, dificulta su espionaje» (NXP Semiconductors 2016).

2.2.5. Fritzing

Es descrito como:

Iniciativa de *hardware* que hace a la electrónica accesible como un material creativo para cualquiera. Ofrece una aplicación, un sitio web de la comunidad y servicios en el espíritu de procesamiento y Arduino, adoptando un ecosistema creativo que permite a los usuarios documentar sobre sus prototipos, compartirlos

con otros y manufacturar circuitos impresos profesionales. (Friends of Fritzing 2016b)

2.2.6. *Hardware libre*

«*hardware* cuyos archivos de origen están disponibles públicamente para que cualquiera lo use, vuelva a manufacturar, rediseñe y revenda» (Gibb et al. 2015, xiii).

2.2.6.1. Tipos principales de *hardware* libre

A continuación, es presentada la tabla 7 una comparación de los tipos principales de dispositivos usados para crear dispositivos más complejos.

Tabla 7. Tipos principales de hardware libre para elaborar proyectos electrónicos.

Tarjeta	Arduino	Raspberry Pi	BeagleBone	UDOO	Intel
Modelo	Arduino Mega2560	Raspberry Pi 3	BeagleBone Black	UDOO Neo Full	Intel Galileo Gen 2
Microprocesador o controlador	ATMega2560 de 8 bits y de 16 MHz	Broadcomm BCM2837 de 64 bits y de 1,2 GHz	AM3358 (ARM Cortex-A8) de 32 bits y de 1 GHz	Freescale i.MX 6SoloX (ARM-Cortex-A9) de	Intel Quark SoC X1000 de 32 bits y de 400 MHz
Almacenamiento	256 KB	Variable (depende de la memoria microSD)	4 GB	Variable (depende de la memoria microSD)	8 MB (Expandible con memoria microSD)
Memoria RAM	8 KB SRAM	1 GB LPDDR2	512 GB DDR3	1 GB	512 KB SRAM
Cantidad y tipo de puertos	1 puerto USB. 1 puerto de alimentación.	1 puerto Ethernet. 1 puerto microSD. 1 puerto HDMI. 4 puertos USB. 1 puerto microUSB para alimentación.	1 puerto Ethernet. 1 puerto microSD. 1 puerto microHDMI. 1 puerto USB de salida.	1 puerto Ethernet. 1 puerto microSD. 1 puerto USB. 1 puerto microHDMI. 1 puerto de alimentación.	1 puerto Ethernet. 1 puerto microSD. 1 puerto USB de entrada. 1 puerto USB de salida. 1 puerto RS-232. 1 puerto de alimentación.

2.2.6.2. Estándares de conexión de módulos y sensores

2.2.6.2.1. I²C. Significa *Inter Integrated Circuit* o ‘circuito inter-integrado’ en español. Es definido como «tipo de bus serial que está conformado por dos pines: SDA (*Serial Data line* o línea de datos) y SDL (*Serial Clock Line* o línea de reloj en serie) el que el módulo tiene los pines SCA y SCL. A cada dispositivo conectado a estos pines le es asignada una única dirección» (NXP Semiconductors 2013, 1).

2.2.6.2.2. UART. Significa *Universal Asynchronous Receiver-Transmitter* o ‘transmisor-receptor asíncrono universal’ en español. Es un «Tipo de dispositivo de comunicación en serie conformado por un transmisor y un receptor que se comunican entre sí por relojes separados» (Chun-zhi, Yin-shui, y Lun-yao 2011, 691).

2.2.7. Internet de las cosas

De acuerdo con la Unión Internacional de Telecomunicaciones, es definido como una «infraestructura global para la sociedad de la información que habilita servicio avanzados por interconectar cosas (físicas y virtuales) basadas en tecnologías de información y comunicación (TIC) interoperables existentes y evolutivas» (ITU 2012, 2).

2.2.8. Lux

«Unidad lumínica del sistema internacional, que equivale a la iluminancia de una superficie que recibe un flujo luminoso de 1 lumen por metro cuadrado» (Real Academia Española 2016a).

2.2.9. Monitorización

«Observación mediante aparatos especiales del curso de parámetros fisiológicos o de otra naturaleza para la detección de posibles anomalías» (Real Academia Española 2016b).

2.2.10. Red de sensores inalámbricos

«Red inalámbrica compuesta de nodos de sensores que sirve para un gran rango de aplicaciones tales como automatización de hogares, monitorización del ambiente y otras cosas» (Shao, Wang, y Rao 2015, 24).

La red de sensores inalámbricos consiste en «dispositivos autónomos distribuidos espacialmente que usan sensores para monitorizar situaciones ambientales y físicas tales como sonido, vibración, movimiento e intensidad de la luz en varios lugares» (Rajeshwari, Shanthini, y Prince 2015, 108).

2.2.11. *Software* de código abierto

Describe un «producto que es distribuido a través de una licencia abierta, y que permite el acceso a su diseño, con el fin de inspeccionarlo, usarlo, modificarlo y redistribuirlo» (Herstatt y Ehls 2015, 8).

2.2.11.1. Tipos de *software* de código abierto seleccionado para diseñar y programar *sketches* de Arduino

A continuación, se muestra una tabla comparativa del *software* utilizado para el desarrollo del prototipo de sistema de monitorización y riego automático (ver Tabla 8).

Tabla 8. Software utilizado en el desarrollo del prototipo de sistema de monitorización y riego automático.

Aplicación	Arduino	Fritzing
Versión	1.6.12	0.9.2 beta
Sistemas operativos soportados	Windows, distribuciones de Linux, OS X.	
Formatos de archivos ejecutables	«.exe» (archivo de instalación o archivo ejecutable comprimido en un archivo «.zip»)	«.exe» (comprimido en un archivo «.tar.gz»)
	«.desktop» (comprimido en un archivo «.tar.gz»)	
	«.app»	
Formato de archivos utilizados	«.ino» (código)	
	«.h» y «.cpp» (librerías)	«.fzz» (proyectos)
		«.fzpz» (partes)
		«.svg» (diseños gráficos vectoriales utilizados en las partes)
Licencia	GPL (entorno de programación)	GPL (código fuente de la aplicación)
	LGPL (librerías)	CC BY-SA (diseños gráficos)

2.2.12. WiFi

«Medio que permite a las computadoras, teléfonos inteligentes u otros dispositivos conectarse a Internet o con otros inalámbricamente en un área particular» (Oxford Unity Press 2016b).

2.3. Marco legal

2.3.1. Arduino

2.3.1.1. Uso de la marca Arduino

«El nombre, el logotipo y los diseños gráficos de las tarjetas son una marca registrada de Arduino LLC en Estados Unidos y Arduino SRL en el resto del mundo» (Arduino LLC 2016d; Arduino SRL 2016b).

Para evitar usar el logotipo oficial, Arduino LLC publica el logotipo «para la comunidad para que la gente identifique sus proyectos como provenientes de Arduino y de la comunidad» (Arduino LLC 2016b).

A partir de 2016, fue anunciada la unión de Arduino LLC y Arduino SRL como «Arduino Holdings» (Arduino LLC 2016f; Arduino SRL 2016d).

2.3.1.2. Uso del entorno de programación de Arduino

«El *software* Arduino es de código abierto. El código abierto para el ambiente Java es lanzado bajo GPL y las librerías de microcontroladores (en C y C++) están bajo LGPL» (Arduino LLC 2016c).

2.3.1.3. Uso del *hardware* y del logotipo

«Los diseños del hardware están protegidos bajo la licencia Creative Commons BY SA, el cual permite a los aficionados y vendedores diseñar sus propias tarjetas si son publicadas bajo la misma licencia» (Arduino LLC 2016c).

2.3.2. Uso de Fritzing

«La documentación de Fritzing está protegida bajo la licencia CC-BY-SA, en la cual se comparten los diseños gráficos de componentes electrónicos dando crédito a sus diseñadores, y asignando estas partes bajo la licencia Creative Commons» (Friends of Fritzing 2016a).

2.3.3. Tipos de licencias usadas

2.3.3.1. CC-BY-SA

Significa Creative Commons Attribution-ShareAlike. «Esta licencia permite a otros re-mezclar, modificar y crear un material, siempre y cuando se acredite al autor y se distribuyan las contribuciones bajo la misma licencia que la original» (Creative Commons 2013b).

2.3.3.2. CC-BY-ND

Significa Creative Commons Attribution-NoDerivatives. «Esta licencia permite a otros distribuir las creaciones, siempre y cuando se acredite al autor y no se remezcle, transforme, ni tampoco se construya sobre el material» (Creative Commons 2013a).

2.3.3.3. GPL

«Licencia que permite distribuir copias de *software* (gratuito o con cobro), y limitar la finalidad del mismo, siempre y cuando se permita la edición, uso de las piezas en otros programas y el acceso al código fuente, y también» (GNU 2016b).

2.3.3.4. LGPL

«licencia que funciona de la misma de manera que GPL, pero que permite a los autores limitar la edición del *software*» (GNU 2016a).

4. Presupuesto

4.1. Versión 1 de SIMORIA (SIMORIA 1P)

A continuación, es presentada la lista de precios de los materiales utilizados en el diseño del prototipo en las tablas 10, 11 y 12.

Tabla 10. Listado de materiales utilizados en el desarrollo la versión 1 de SIMORIA (prototipo en Arduino).

Proveedor	Cantidad	Componente electrónico	Valor (COP)
Dynamo Electronics	1	Arduino Mega2560 original	160.439,20
	1	Módulo relé	14.500,00
		Domicilio por Movilenvíos	5.020,00
Electrónica I+D	1	Módulo WiFi ESP8266MOD	22.088,00
	1	Sensor de humedad capacitivo	28.312,80
		Envío por Deprisa	9.036,00
Vistrónica S.A.S.	1	Shield para conectar módulos y sensores (Sensor Shield V4)	33.332,80
	1	Sensor de humedad en el suelo (sonda FC-08 y módulo LM393)	7.530,00
	1	Módulo RTC DS3231/AR24C32	10.844,20
	1	Paquete de 40 cables <i>jumper</i> hembra-hembra	8.032,00
	1	Paquete de 40 cables <i>jumper</i> macho-hembra	8.032,00
		Envío por Deprisa	8.534,00
Subtotal 1			315.701,00

Tabla 11. Segunda parte del listado de materiales utilizados en el desarrollo de la versión 1 de SIMORIA.

Proveedor	Cantidad	Componente electrónico	Valor (COP)
Electronilab	1	Sensor de iluminancia TSL2591	19.578,00
	1	Sensor de temperatura y humedad relativa DHT22	23.393,20
		Envío por Interrapidísimo	8.534,00
Metalcril	1	Carcasa para el prototipo	60.000,00
	1	Carcasa para sensor	23.000,00
Subtotal 2			134.505,20

Tabla 12. Cálculo del precio total invertido para el desarrollo de la versión 1 de SIMORIA.

Subtotal 1	315.701,00
Subtotal 2	134.505,20
Total en la version 1	450.206,20

4.2. Versión 2 de SIMORIA (SIMORIA 1P2)

Modelo portátil SIMO que consume de 5 a 10 V.

Tabla 13. Listado de materiales utilizados en el desarrollo de la versión 2 de SIMORIA.

Proveedor	Cantidad	Componente electrónico	Valor (COP)
Electrónica I+D	1	NODEMCU V3 con <i>shield</i> para motores	48.192,00
		Envío por Deprisa	9.036,00
Vistronica S.A.S.	1	Sensor de humedad en el suelo (sonda FC-08 y módulo LM393)	7.530,00
	1	Sensor de temperatura y humedad relativa DHT11	8.734,80
	1	Sensor de iluminancia BH1750FVI/GY30	14.325,07
	1	Módulo relé	6.405,52
		Envío por Deprisa	8.534,00
Total			102.757,39

4.3. Adaptación a SIMORIA

Tabla 14. Listado de materiales utilizados en el desarrollo del sistema riego automático de SIMORIA (versiones 1 y 2).

Proveedor	Cantidad	Componente electrónico	Valor (USD)	Equivalencia de 1 USD (COP)	Valor (COP)
Electronilab	1	Bomba peristáltica de 4-5 vatios y de 500 litros por hora			34.638,00
Ferrelectric la 33	2,5	Metro de manguera de 1 cm			2.500,00
Electrónica General	2	Regleta macho			1.800,00
	2	Adaptador de poder hembra			3.400,00
	2	Adaptador de poder macho			3.400,00
	1	Metro de cable termo-incogible			300,00
	1	Metro de pares de cables rojo y negro de 22 AWG			600,00
Vistrónica S.A.S.	1	Adaptador de corriente de 12 voltios y 4 amperios			25.703,40
	1	Paquete de 40 cables <i>jumper</i> hembra-hembra			8.032,00
	1	Paquete de 40 cables <i>jumper</i> macho-hembra			8.032,00
Linio	1	Paquete de 6 Etiquetas adhesivas NTag216			21.900,00
Google Play	1	Aplicación móvil (NFCTools Pro)	2,82	3069,17	8.655,06
Total					123.360,46

5. Metodología

A continuación, son presentadas las fases llevadas a cabo para realizar el prototipo en la figura 1.

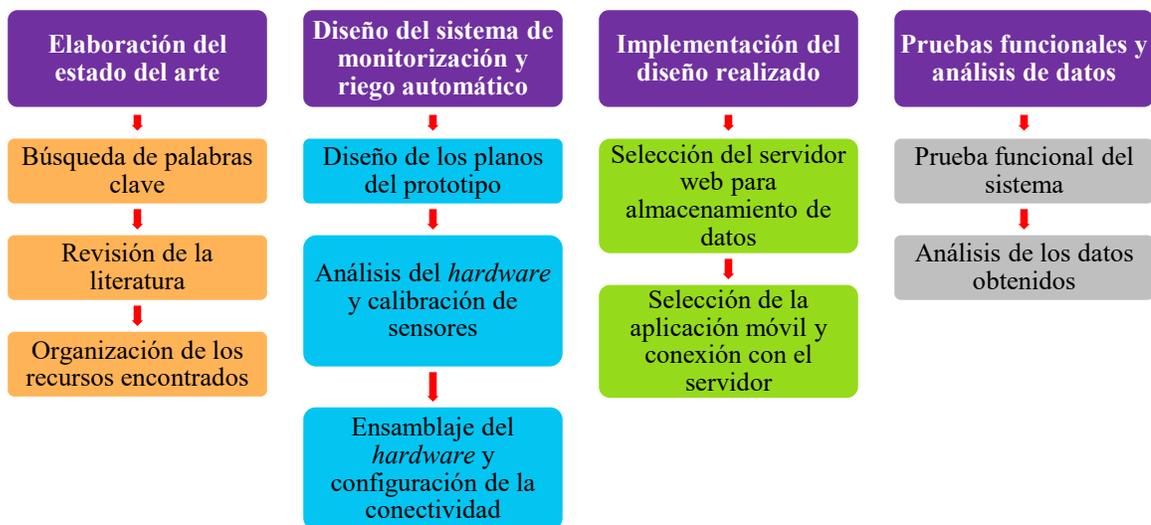


Figura 1. Metodología presentada para la elaboración del prototipo.

5.1. Elaboración del estado del arte

Los artículos encontrados son encontrados y organizados de acuerdo con los criterios de búsqueda utilizados (ver tabla 1).

5.2. Diseño del sistema de monitorización y riego automático

En la figura 2, es presentado un diagrama que representa el campo de acción y los elementos utilizados en el prototipo.



Figura 2. Representación de los elementos utilizados en el prototipo.

5.3. Implementación del diseño realizado

Fue encontrado y seleccionado un servidor web adecuado de acuerdo con su funcionalidad y capacidad para el registro de datos obtenidos por los sensores conectados a Arduino. Posteriormente, fue seleccionada una aplicación móvil que permita a los usuarios ver los datos almacenados en el servidor seleccionado.

5.4. Pruebas funcionales y análisis de datos

Fueron realizadas pruebas de estabilidad del servidor web y la aplicación móvil seleccionada. También fue analizado el límite de visualización y análisis de datos de estas herramientas.

6. Avances de SIMORIA

6.1. Elaboración del estado del arte

6.1.1. Búsqueda de palabras clave

Fueron determinadas las palabras clave que muestren la esencia del proyecto, y posteriormente, se utilizaron como criterio de búsqueda de referencias (ver tabla 1).

6.1.2. Revisión de la literatura

Los artículos fueron seleccionados usando las palabras clave y la fecha de publicación (de 2011 en adelante).

6.1.3. Organización de los recursos encontrados

Fueron seleccionados artículos relevantes que resalten la esencia del proyecto, o una parte de este. Se observó que fueron encontradas fuentes de páginas web, libros, artículos científicos, tesinas y diseños incluidos en la aplicación Fritzing.

6.2. Diseño del sistema de monitorización y riego automático

6.2.1. Diseño de los planos del prototipo

Para el diseño de los planos, se usó una aplicación de código abierto llamada Fritzing. Antes de diseñar los planos, es necesario realizar el diseño gráfico de los componentes electrónicos no existentes en Fritzing. Pueden usarse «editores de gráficos vectoriales como Adobe Illustrator, CorelDraw o también Inkscape» (Cortez 2014). En este caso, se usaron CorelDraw (versión de prueba) e Inkscape (software de código abierto). Para que los componentes electrónicos funcionen en Fritzing, fueron elaborados tres diseños gráficos para cada componente: tarjeta, esquema electrónico y circuito

impreso. Posteriormente, fueron añadidos los diseños elaborados en Fritzing y fueron asignadas las conexiones entre los componentes (en la vista de tarjetas). Las conexiones en la vista de esquema y de circuito electrónico son asignadas automáticamente después de editar en la vista de tarjetas.

Finalmente, es mostrado el diseño completo en las vistas: de tarjeta (figura 3), de esquema (figura 4) y de circuito electrónico (ver figura 5).

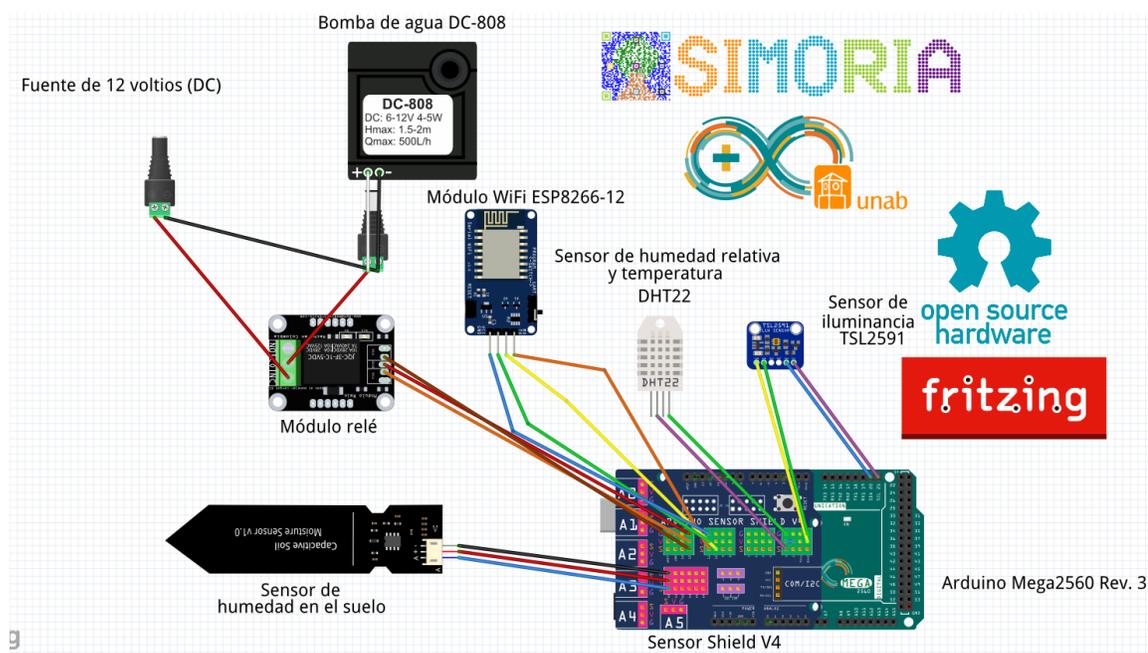


Figura 3. Vista previa del plano del prototipo de sistema de monitorización y riego automático.

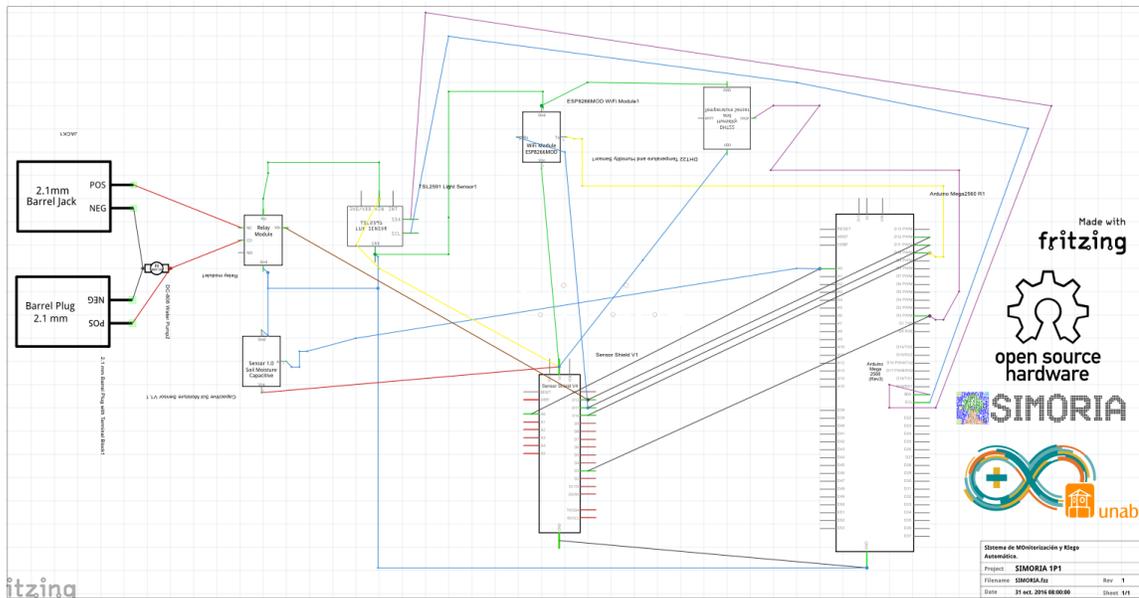


Figura 4. Esquema del plano del prototipo de sistema de monitorización y riego automático.

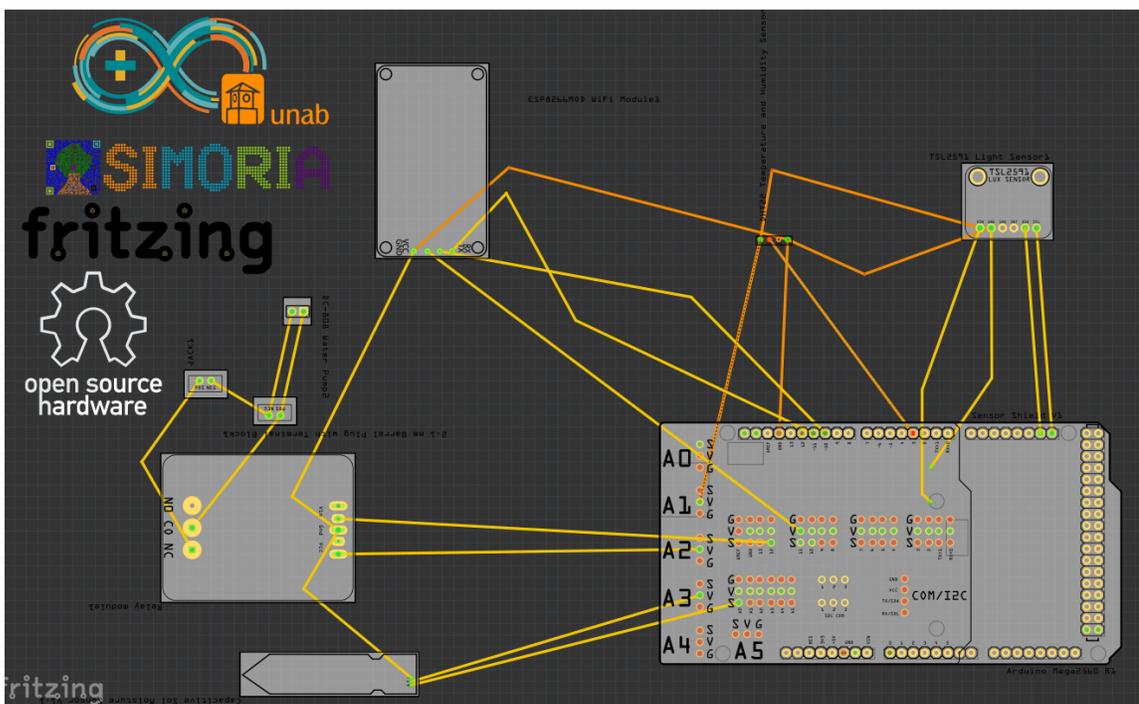


Figura 5. Circuito del sistema de monitorización y riego automático.

En los planos del prototipo son incluidos el logotipo de la comunidad de Arduino (ver figura 6) y el logo de SIMORIA (Sistema de MONitorización y RIego Automático) (ver figura 7). El logotipo de SIMORIA contiene un código QR que puede ser leído por aplicaciones lectoras de QR para Android y iOS (ver figura 8).



Figura 6. Logotipo de la comunidad de Arduino.

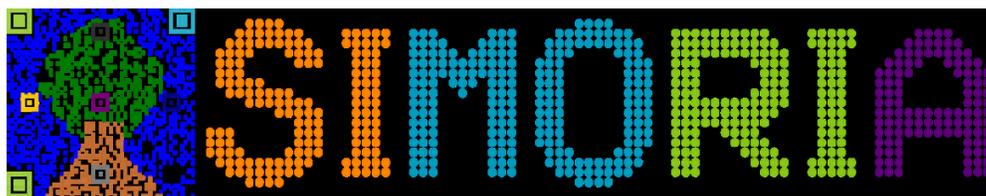


Figura 7. Logotipo de SIMORIA.



Figura 8. Código QR contenido en el logotipo de SIMORIA.

6.2.2. Análisis del *hardware* y la calibración de sensores

6.2.2.1. Sensor de humedad relativa y temperatura DHT22

Según Aosong Electronics (2016), el sensor de humedad relativa y temperatura DHT22 tiene un margen de error de « $\pm 2\%$ de humedad relativa, y $\pm 0,2^{\circ}\text{C}$ de temperatura» (2), y un alcance «de 0-100% de humedad relativa y desde -40°C hasta aproximadamente 125°C » (1).

6.2.2.2. Sensor de iluminancia TSL2591

Según Adafruit (2016), El sensor de iluminancia TSL2591 tiene alcance desde « $188\mu\text{lx}$ hasta $88,000\text{ lx}$ » (3), y tiene un error aproximado a 0 lx, porque solo registra valores enteros de acuerdo con las pruebas experimentales (ver **¡Error! No se encuentra el origen de la referencia.**). En estas pruebas es mostrado un alcance de 0 hasta $88,000\text{ lx}$.

6.2.2.3. Sensor de humedad en el suelo capacitivo

El alcance del sensor de humedad fabricado por DFROBOT es de 268 a 652, porque es alcance hallado en las pruebas realizadas (ver **¡Error! No se encuentra el origen de la referencia.**), y tiene un error aproximado a 0, porque el sensor solo registra números enteros.

La calibración del sensor de humedad en el suelo fue necesaria debido a la falta de unidades en las medidas observadas.

Para resolver este problema, los porcentajes fueron añadidos a estas medidas. Primero, fue organizada una tabla con los valores medidos, y fue asumido que un

ambiente seco la humedad en el suelo es 0%, y que en un ambiente totalmente húmedo (vaso con agua) la humedad en el suelo es de 100% (ver tabla 15).

Tabla 15. Tabla de del mínimo y del máximo valor de humedad en el suelo obtenidos por el sensor.

Medidas del sensor de humedad h (sin unidades)	Porcentaje de humedad en el suelo p (%)	Ambiente en el que fue probado el sensor
268	100	Totalmente mojado (sumergido en un vaso con agua)
652	0	Totalmente Seco (expuesto al aire)

Después del registro de datos iniciales en la tabla, fue empleada la ecuación de la pendiente, asumiendo que es una función matemática lineal, debido a que los conjuntos de valores iniciales son dos:

$$(268,100) \text{ y } (652,0)$$

A continuación, es mostrada la ecuación de la pendiente de funciones lineales.

$$p = mh + c$$

Para convertir los datos de humedad en el suelo « h » a porcentaje « p », es necesario calcular la pendiente « m » primero:

$$m = \frac{0 - 100}{652 - 268} = -\frac{100}{384} = -\frac{25}{96}$$

Segundo, es reemplazado el valor de la pendiente en la ecuación:

$$p = -\frac{25}{96}h + c$$

Tercero, son reemplazados los valores de humedad y porcentaje (ver tabla 15). En este caso, fueron usados los valores 268 para humedad, y 100 para porcentaje

$$100 = -\frac{25}{96}(268) + c$$

Posteriormente, es despejada la constante que complementa la función lineal «c» de la ecuación:

$$100 = -\frac{6700}{96} + c = -\frac{1675}{24} + c$$

Despejando «c»,

$$c = 100 - \left(-\frac{1675}{24}\right) = 100 + \frac{1675}{24}$$

$$c = \frac{4075}{24}$$

Finalmente, es armada la ecuación de porcentaje:

$$p = -\frac{25}{96}h + \frac{4075}{24}$$

Después de hallar la ecuación, fue graficada la función que permite hallar el porcentaje mostrado por el sensor de humedad en el suelo (ver figura 9).

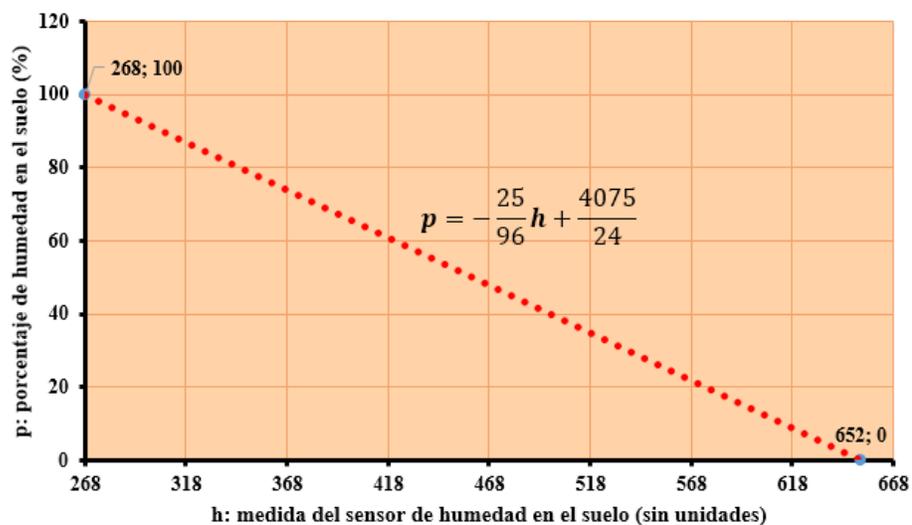


Figura 9. Gráfica que determina la función de porcentaje de humedad en el suelo a partir de las lecturas del sensor.

6.2.3. Ensamblaje del *hardware* y configuración de la conectividad

A continuación, son mostrados los pasos de ensamblaje y configuración de SIMORIA.

6.2.3.1. Paso 1

Para el diseño de SIMORIA, fue usada una tarjeta Arduino Mega2560 V3 como un dispositivo de procesamiento de datos obtenidos por los sensores (ver figura 10).

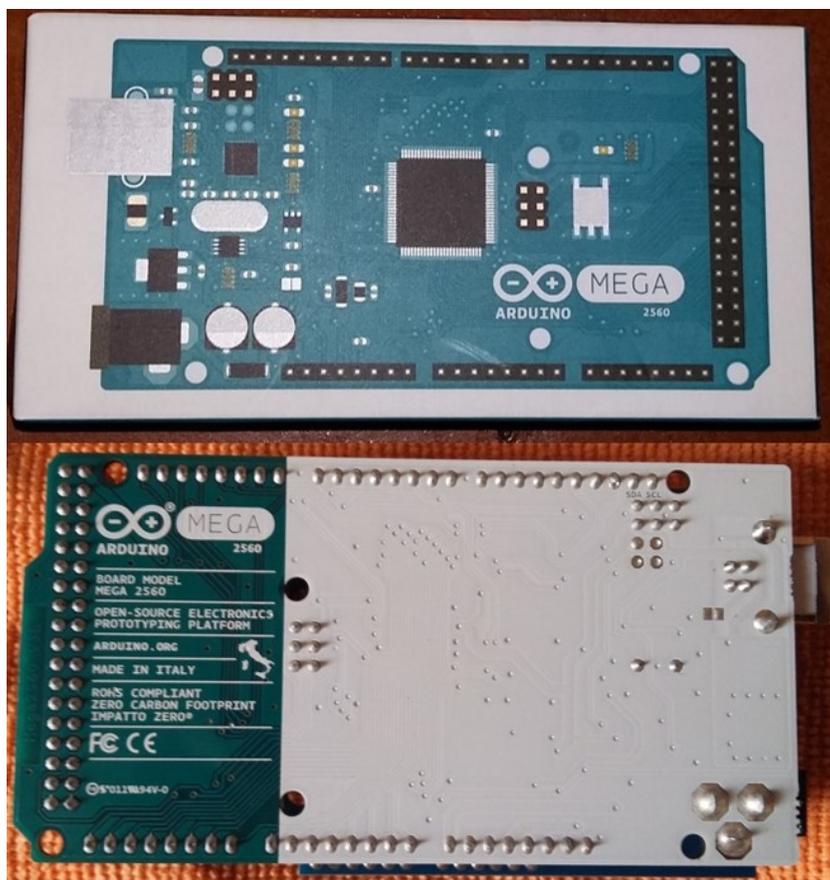


Figura 10. Vistas delantera y posterior de la tarjeta Arduino Mega2560.

6.2.3.2. Paso 2

Posteriormente, fue ensamblado el *Sensor Shield V4* (ver figura 11), lo cual permitió conectar los sensores y módulos a Arduino Mega2560.

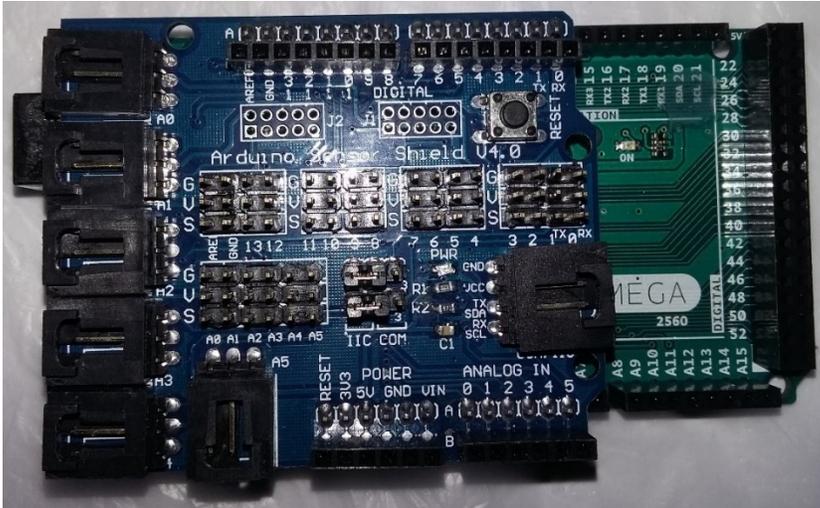


Figura 11. Sensor Shield V4.

6.2.3.3. Paso 3

Luego, fueron conectados los sensores de humedad en el suelo (ver figura 12), humedad relativa y temperatura (ver figura 13), y también de iluminancia (ver figura 14).



Figura 12. Sensor de humedad capacitivo.



Figura 13. Sensor de humedad relativa y temperatura DHT22.



Figura 14. Sensor de iluminancia TSL2591.

6.2.3.4. Paso 5

Fueron conectados módulos como el relé (ver figura 15) y el módulo WiFi ESP8266MOD (ver figura 16).



Figura 15. Módulo relé.



Figura 16. Módulo WiFi ESP8266MOD.

6.2.3.5. Paso 6

El funcionamiento de los sensores y módulos conectados fue programado usando la aplicación Arduino (ver figura 17).



```

1 | /*-----SIMORIA - Versión 1P-----*/
2 | //-----Bibliotecas incluidas-----//
3 | #include<SoftwareSerial.h>//Librería de módulos ESP8266.
4 | #include "DHT.h"//Librería de sensores DHT.
5 | #include <Wire.h>//Librería de pines SDA y SCL.
6 | #include <Adafruit_Sensor.h>
7 | //Librería de sensores Adafruit.
8 | #include <Adafruit_TSL2591.h>
9 | //Librería del sensor de//iluminancia TSL2591.
10 |
11 | //-----Código inicial de DHT22-----//
12 | #define DHTPIN 3//Pin D3.
13 | #define DHTTYPE DHT22//DHT 22 o AM2302.
14 | DHT dht(DHTPIN, DHTTYPE);//Inicio del sensor DHT22.
15 |
16 | //-----Código inicial para TSL2591-----//
17 | Adafruit_TSL2591 tsl = Adafruit_TSL2591(2591);
18 | //Conexión de SDA al pin D20 y SCL a D21
19 | //(Arduino Mega2560).
20 |
21 | //----Inicio de variable existente (bomba de agua)----//
22 | int bomba_agua=12;//Pin D12.
23 |
24 | //-----Código inicial para el módulo ESP8266-----//
25 | // conexión TX a D10 y de RX a D11.
26 | SoftwareSerial serie(10, 11);
27 | //Asignación de D10 como RX y de D11 como TX.

```

Figura 17. Programación del funcionamiento de los sensores del prototipo en el entorno de programación Arduino.

Después de la escritura del código para la programación de funciones de sensores, fue seleccionada la tarjeta Arduino Mega2560 en la aplicación (ver figura 18).

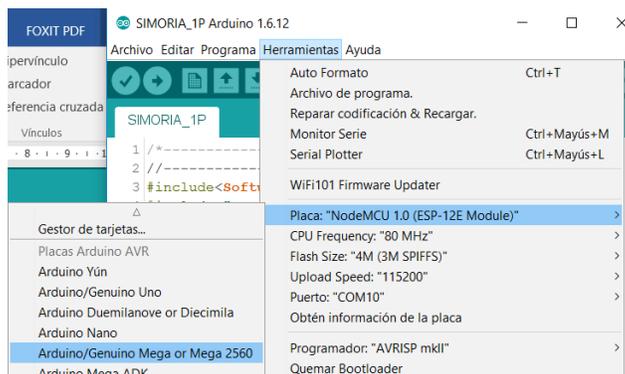


Figura 18. Selección de tarjetas en el entorno Arduino.

6.2.3.6. Paso 8

Fue utilizada una carcasa (ver figura 19) para la protección del dispositivo y garantizar un aspecto estético adecuado (ver figura 20). Finalmente, el dispositivo fue puesto en marcha.



Figura 19. Carcasa diseñada para el encaje del dispositivo electrónico.

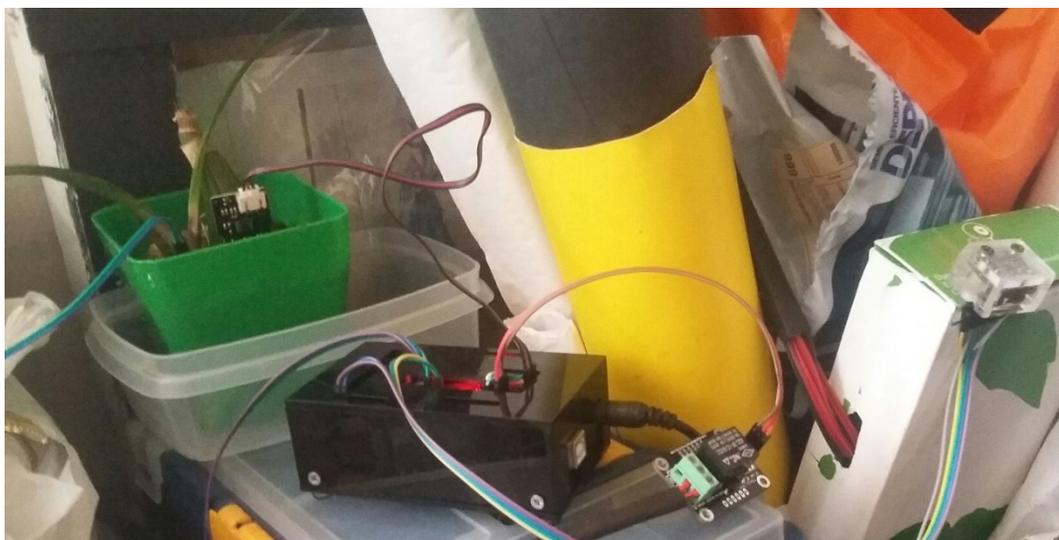


Figura 20. Dispositivo electrónico puesto en marcha.

6.3. Implementación del diseño realizado

6.3.1. Selección del servidor web para almacenamiento de datos

Actualmente, fueron examinados los servidores web de acuerdo con los criterios empleados (ver tablas 16 y 17).

Tabla 16. Selección del servidor web (parte 1).

Servidor web	Adafruit	Blynk	Freeboard	PubNub
Nombre de interfaz	<i>Dashboard</i>	Proyecto	Dashboard	Aplicación
Forma de visualización de datos	Bloque	Plugin	<ul style="list-style-type: none"> • Pane (conjunto de <i>widgets</i>). • <i>Widget</i> (lectura de sensores). 	Nodo
	Privado	Privado o privado	Público o privado	Privado
Muestra de datos en tiempo real	Sí (cada 2 segundos)	Sí (cada 3 segundos)	Sí (cada 2 segundos)	Sí (cada Segundo, 5-10 dólares por gráfico)
Muestra fecha de registro de datos	Sí	Sí	No	No
Almacena datos	No	Sí	No	No
Permite extraer datos obtenidos	No	Sí	No	No
Analiza datos	No	No	No	No
Genera reportes de datos	Sí (por correo electrónico)	Sí	Sí (por correo a través del código de Dweet.io)	Sí (por Google Cloud Messaging)
Adición de complementos (Google Maps, imágenes, videos y/u otros)	No	No	Sí	No
Integración con otras apps	No	Sí (con aplicaciones web)	No	No
Total	3	6	3	2

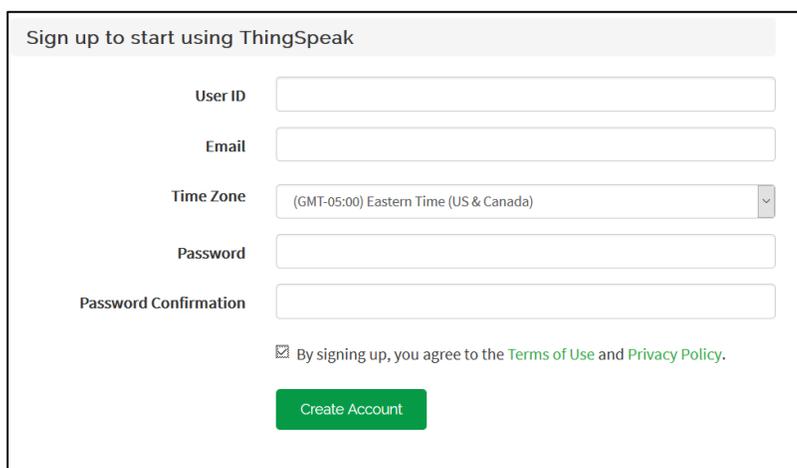
Tabla 17. Selección del servidor web (parte 2).

Servidor web	Thingier.io	ThingSpeak	Ubidots	Xively
Nombre de interfaz	<i>Dashboard</i>	Channel	<i>Dashboard</i>	<i>Dispositivo</i>
Forma de visualización de datos	<i>Widget</i>	Campo	<i>Widget</i> (conjunto de variables)	Canal
	Privado	Público o privado	Privado	Privado
Muestra de datos en tiempo real	Sí (cada segundo)	Sí (cada 15 segundos)	Sí (cada 3 segundos)	Sí (cada segundo)
Muestra fecha de registro de datos	Sí	Sí	Sí	No
Almacena datos	Sí (hasta 100 MB mensuales)	Sí	Sí (3 meses de datos generados)	No
Permite extraer datos obtenidos	Sí	Sí	Sí	No
Analiza datos	No	Sí (una cosa a la vez)	No	No
Genera reportes de datos	Sí (por correo electrónico)	Sí (por Twitter)	Sí (por correo electrónico o por SMS)	Sí (por SMS)
Adición de complementos (Google Maps, imágenes, videos y/u otros)	Sí	Sí	Sí	No
Integración con otras apps	No	Sí (con MATLAB)	No	Sí (con Xively Client Feed)
Total	6	8	6	3

Después de la revisión, fue escogido ThingSpeak como servidor web, porque los datos son mostrados en gráficas y en forma detallada, posee una base de datos con gran almacenamiento, y posee una interfaz personalizable con complementos web como Google Maps, videos de YouTube, y presentación de imágenes como logotipos e imágenes personalizadas.

6.3.1.1. Uso del servidor web seleccionado

Para el uso del servidor web seleccionado, fue necesario crear una cuenta en ThingSpeak (ver figura 21).



The image shows a sign-up form for ThingSpeak. The form is titled "Sign up to start using ThingSpeak" and contains the following fields and elements:

- User ID:** A text input field.
- Email:** A text input field.
- Time Zone:** A dropdown menu with the selected option "(GMT-05:00) Eastern Time (US & Canada)".
- Password:** A text input field.
- Password Confirmation:** A text input field.
- By signing up, you agree to the [Terms of Use](#) and [Privacy Policy](#).
- Create Account:** A green button.

Figura 21. Creación de cuenta en ThingSpeak.

Posteriormente, el usuario creado es accedido, y el nuevo canal es creado oprimiendo el botón «New Channel» (ver figura 22).

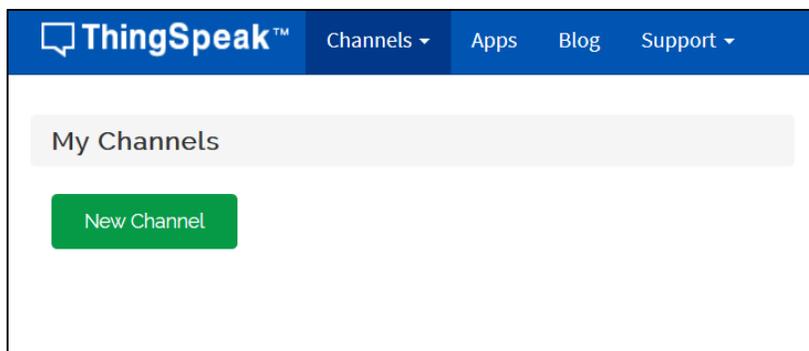


Figura 22. Interfaz de usuario en ThingSpeak.

En la página web de creación de canales es necesario escribir el nombre del canal y el de los campos a utilizar (ver figura 23). En este caso, fueron escogidos cuatro campos: humedad en el suelo (HS), humedad relativa (HR), temperatura (T) e iluminancia (I).

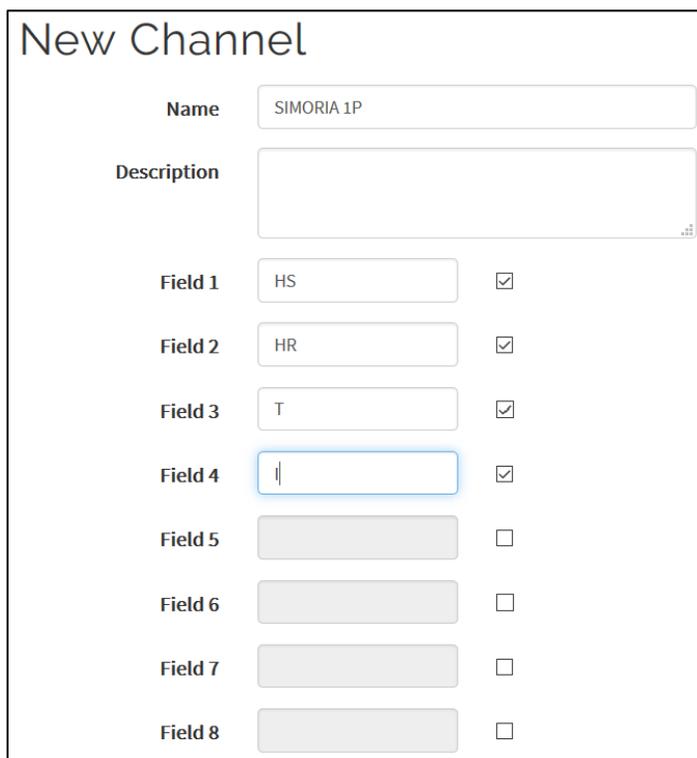
The image displays the 'New Channel' form. At the top, the title 'New Channel' is visible. Below it, there are several input fields: 'Name' (containing 'SIMORIA 1P'), 'Description' (an empty text area), and a list of eight 'Field' entries. Each field entry consists of a text input box and a checkbox. Fields 1 through 4 are active, with checkboxes checked and values 'HS', 'HR', 'T', and 'I' respectively. Fields 5 through 8 are disabled, with greyed-out input boxes and unchecked checkboxes.

Figura 23. Interfaz de creación de canales en ThingSpeak (primera parte).

Posteriormente, es posible marcar la opción de hacer el canal público (si se desea), y es necesario oprimir en el botón «Save Channel» (ver figura 24).

The image shows a web form for creating a channel in ThingSpeak. The form is enclosed in a black border and contains the following elements:

- Make Public**: A checkbox that is checked.
- URL**: A text input field.
- Elevation**: A text input field.
- Show Location**: A checkbox that is unchecked.
- Latitude**: A text input field containing the value "0.0".
- Longitude**: A text input field containing the value "0.0".
- Show Video**: A checkbox that is unchecked, with two radio button options: "YouTube" (which is selected) and "Vimeo".
- Video ID**: A text input field.
- Show Status**: A checkbox that is unchecked.
- Save Channel**: A green button at the bottom of the form.

Figura 24. Interfaz de creación de canales en ThingSpeak (segunda parte).

Finalmente, es creada la interfaz en ThingSpeak con los datos a obtener desde la tarjeta Arduino (ver figura 25).

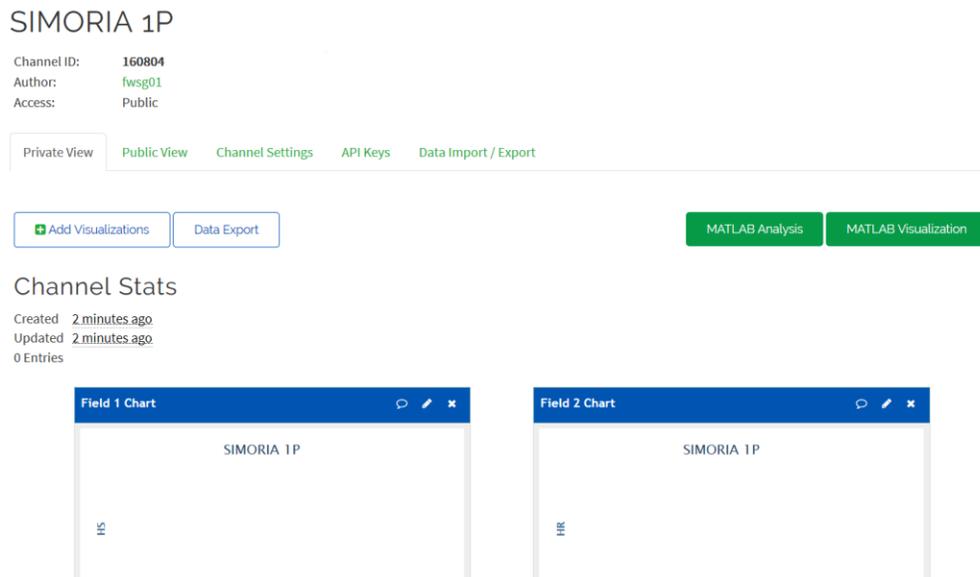


Figura 25. Interfaz de visualización de datos en ThingSpeak.

6.3.2. Selección de la aplicación móvil y conexión con el servidor

6.3.2.1. Selección de la aplicación móvil

El diseño de la aplicación móvil no fue necesario, ya que en Android fueron encontradas aplicaciones móviles que permiten la visualización de datos en ThingSpeak de forma rápida e intuitiva (ver tabla 18).

Tabla 18. Aplicaciones móviles existentes y relevantes utilizadas en Internet de las cosas.

Aplicación móvil	Blynk	ThingView
Sistema operativo	iOS, Android	Android
Visualización individual de cada dato obtenido.	Sí (valores numéricos o alfanuméricos)	Sí (valores numéricos en gráficas de datos)
Visualización general de datos obtenidos	Parcial (algunos datos son visibles).	Sí
Sincronización automática de datos	Sí	Parcial
Análisis de datos	No.	Parcial (Cálculo de datos máximos y mínimos)
Precio	0 USD Widgets: desde 0,99 USD.	0 USD Versión sin publicidad: 5000 COP
Tarjetas compatibles	Arduino WiFi, Raspberry Pi, NodeMCU.	Arduino WiFi, Raspberry Pi, módulos ESP8266.

De acuerdo con los criterios empleados para la selección de una aplicación, fue escogida ThingView, por tener compatibilidad con el módulo ESP8266 empleado en SIMORIA, y también por la visualización de datos en gráficos grandes y detallados que pueden ser observados por los usuarios.

En Google Play, fue encontrada una aplicación llamada ThingView, que permite ver datos publicados en ThingSpeak (Cinetica Tech 2016).

Las pruebas de aplicaciones fueron enfocadas en el sistema operativo Android, porque es «el más usado por los usuarios que porten teléfonos inteligentes y tabletas» (Net Applications 2016). Esto es comprobable por medio de los gráficos de la cuota de mercado de los teléfonos móviles y tabletas encontrados en NetMarketShare (ver Figura 26) y StatCounter (ver Figura 27). En la última fuente también se comprueba que Android es el «sistema operativo más usado por los usuarios» (StatCounter 2016).

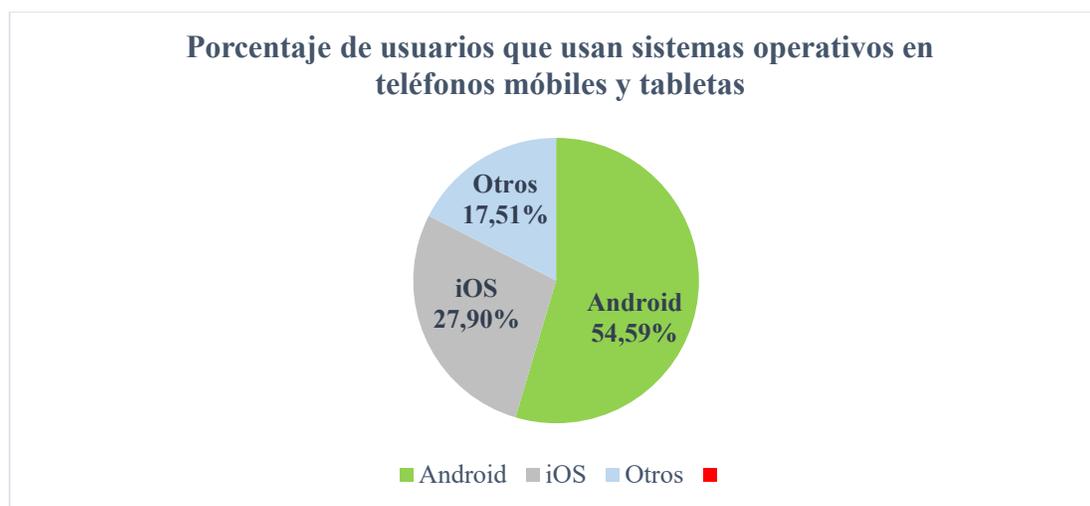


Figura 26. Cuota del mercado de los dispositivos móviles y tabletas en 2016 mostrado en NetMarketShare.

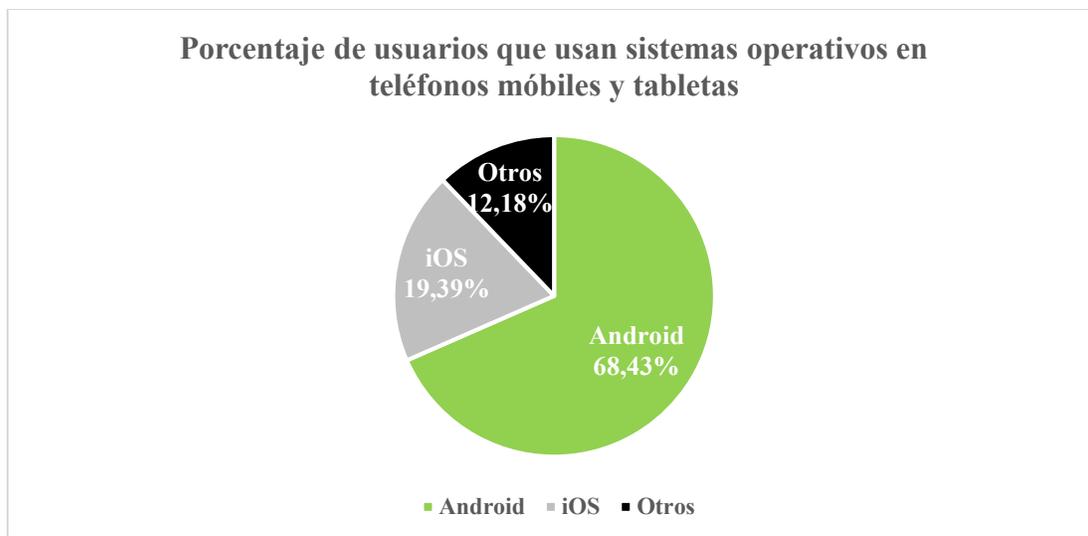


Figura 27. Cuota del mercado de los dispositivos móviles y tabletas en 2016 según StatCounter.

6.3.2.2. Conexión con el servidor web

Para lograr una conexión óptima a ThingSpeak, fue necesaria la creación de un canal (ver 6.3.1.1). Posteriormente, fue necesario oprimir en la opción «API Keys» (ver figura 28).

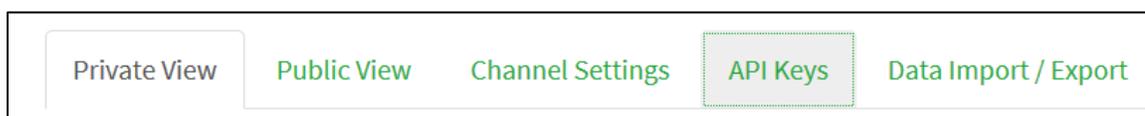


Figura 28. Opción «API Keys» en ThingSpeak.

Luego, fue copiado la clave de escritura o «Write API Key» (ver figura 29) en el código de Arduino (ver **¡Error! No se encuentra el origen de la referencia.**) para permitir el envío de datos obtenidos de los sensores a ThingSpeak.

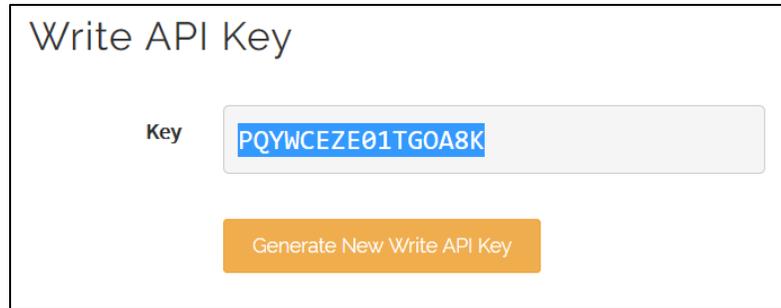


Figura 29. Clave API de escritura del canal creado en ThingSpeak.

Para el envío de datos de sensores, fue necesario escribir el enlace de acceso en el código de Arduino (ver **¡Error! No se encuentra el origen de la referencia.**):

«https://api.thingspeak.com/update?api_key=clave_api_escritura&field1=sensor1&field2=sensor2&field3=sensor3&field4=sensor4» (Blynk 2016; The Matworks Inc. 2016).

Finalmente, es posible la visualización de datos obtenidos en ThingSpeak (ver Figura 30).



Figura 30. Visualización de datos en ThingSpeak.

6.3.2.3. Uso de la aplicación móvil seleccionada

Para usar ThingView, fue necesario el ingreso del ID del canal (ver figura 31) y de la clave API de lectura o «Read API KEY» para permitir la visualización de datos (ver figura 32). Posteriormente, es pulsado el botón «Search», y luego «Done» (ver figura 33).



Figura 31. ID del canal en ThingSpeak (parte subrayada).



Figura 32. Clave API de lectura del canal creado en ThingSpeak.

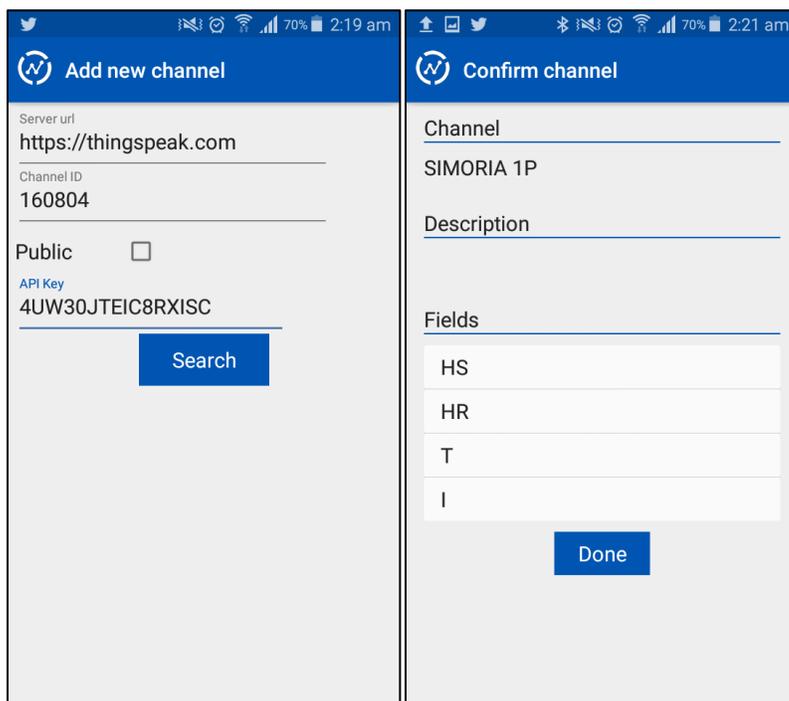


Figura 33. Ingreso de datos del canal creado en ThingSpeak.

Finalmente, son visualizados los datos obtenidos de los sensores conectados al dispositivo SIMORIA (ver figura 34). En la visualización individual de datos, es posible observar el máximo, el mínimo y el último dato obtenido (ver figura 35).

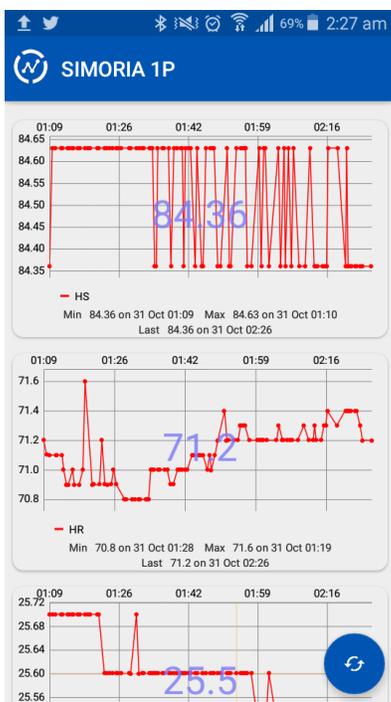


Figura 34. Visualización de datos en ThingView.

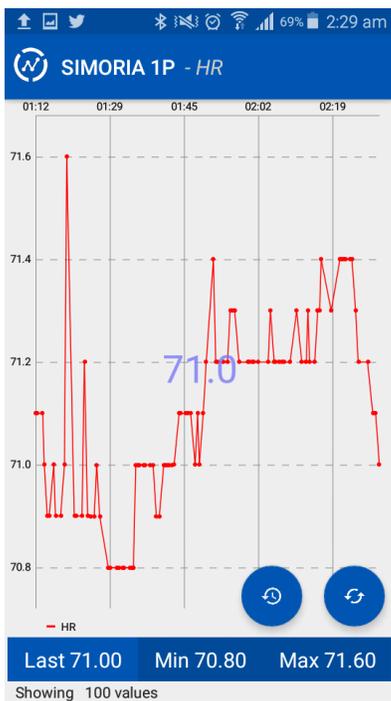


Figura 35. Visualización de temperatura en ThingView.

6.4. Pruebas funcionales y análisis de los datos

6.4.1. Prueba funcional del sistema

A continuación, se presenta un listado de pruebas realizadas en el prototipo del sistema de monitorización y riego automático:

1. El sensor de humedad en el suelo fue probado, y se observó que muestra medidas sin unidades.
2. El sensor de temperatura y humedad relativa DHT22 fue probado, y se observó que funciona a la perfección.
3. El sensor de iluminancia TSL2591 fue probado, y funciona a la perfección.
4. El módulo WiFi ESP8266 fue probado, y se observó que funciona a la perfección. Además, envía datos en un intervalo de 15 a 16 segundos.
5. Se encaja el prototipo en una carcasa diseñada por Metalcril, y se observó que uno de los huecos es demasiado pequeño, y, por lo tanto, no se podía encajar ningún módulo, ni tampoco sensores.
6. Metalcril agrandó el hueco pequeño, se encaja el prototipo de nuevo, y encajó perfectamente.
7. Fue hecha una prueba de medición hipotética de corriente, y se notó que SIMORIA puede consumir hasta 12 voltios y 691.5 amperios aproximadamente, el sensor de humedad en el suelo capacitivo consume 1 miliamperio, el de humedad relativa y temperatura DHT22 1,5 miliamperios, el de iluminación TSL2591 20 miliamperios y la bomba de agua 417 miliamperios.

6.4.2. Análisis de los datos obtenidos

De acuerdo con los datos registrados en ThingSpeak, es posible deducir que el aumento en la temperatura ambiental y la iluminación del sol disminuye la humedad en el suelo y la humedad relativa. En cambio, cuando disminuyen la temperatura y la iluminación, la humedad en el suelo disminuye, y la relativa tardan en aumentar.

En ThingView, es posible la visualización de un análisis básico como valores mínimos y máximos en cada gráfica de datos.

7. Trabajo adicional

7.1. Versiones de SIMORIA

7.1.1. Versión 1P (SIMORIA 1P)

Versión desarrollada con la tarjeta Arduino Mega2560. Fue la versión enfocada en este documento (ver 6.2, 6.3 y 6.4). La versión fue nombrada después del desarrollo de la posterior versión (ver 7.1.2). Envía datos a ThingSpeak en un intervalo de tiempo de 15 a 16 segundos aproximadamente.

7.1.2. Versión 1P2 (SIMORIA 1P2)

Versión adicional, pequeña, económica y más rápida de SIMORIA. Esta fue desarrollada con una tarjeta de desarrollo ESP8266, también conocida como NodeMCU. Envía datos de 1 a 3 segundos a Blynk, y posteriormente, de Blynk a ThingSpeak en un intervalo de 15 segundos.

predeterminado para los sensores análogos conectados a Arduino» (Arduino LLC 2016a; Arduino SRL 2016a).

7.1.2.1. Desarrollo de SIMORIA 1P2

Para el desarrollo de esta versión, fue empleada una tarjeta de desarrollo NodeMCU (ver figura 36).



Figura 36. Vistas delantera y trasera de la tarjeta de desarrollo NodeMCU.

Posteriormente, fue ensamblada la tarjeta NodeMCU en un *shield* para motores (ver figura 37).



Figura 37. *Motor Shield* para tarjetas NodeMCU.

Luego, fueron ensamblados los sensores de humedad en el suelo (ver figura 38), humedad relativa y temperatura (ver figura 39), y también, de iluminancia (ver figura 40).

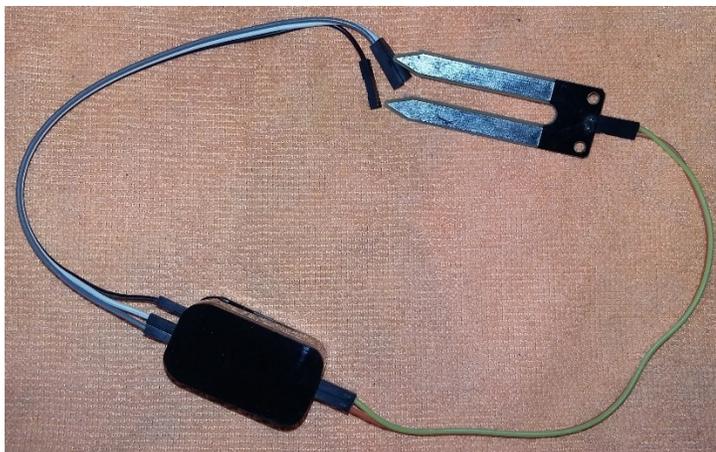


Figura 38. Sensor de humedad en el suelo.



Figura 39. Sensor de humedad relativa y temperatura DHT22.

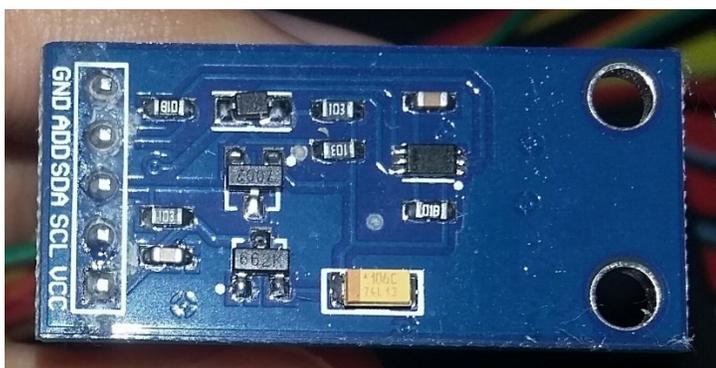
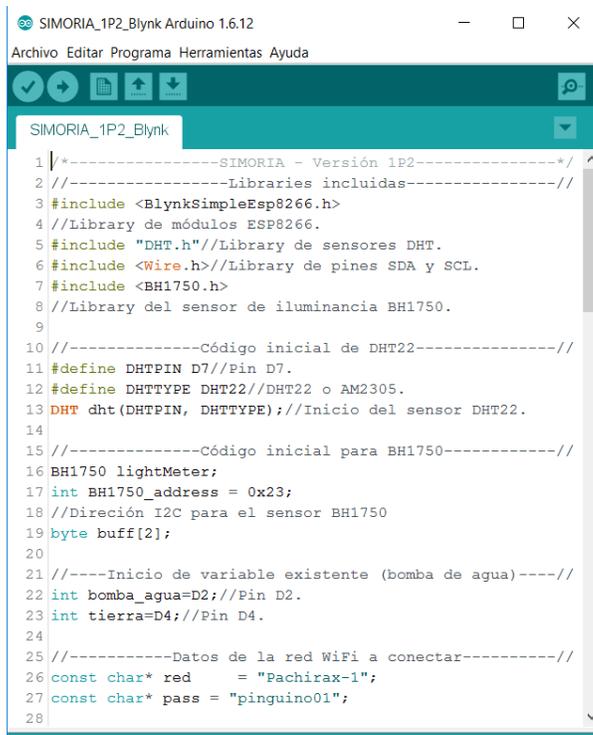


Figura 40. Sensor de iluminancia BH1750.

Finalmente, las funciones de los sensores y el envío de datos a Blynk fueron programados (ver figura 41).



```
1 |/*-----SIMORIA - Versión 1P2-----*/
2 |//-----Libraries incluidas-----//
3 |#include <BlynkSimpleEsp8266.h>
4 |//Library de módulos ESP8266.
5 |#include "DHT.h"//Library de sensores DHT.
6 |#include <Wire.h>//Library de pines SDA y SCL.
7 |#include <BH1750.h>
8 |//Library del sensor de iluminancia BH1750.
9 |
10 |//-----Código inicial de DHT22-----//
11 |#define DHTPIN D7//Pin D7.
12 |#define DHTTYPE DHT22//DHT22 o AM2305.
13 |DHT dht(DHTPIN, DHTTYPE);//Inicio del sensor DHT22.
14 |
15 |//-----Código inicial para BH1750-----//
16 |BH1750 lightMeter;
17 |int BH1750_address = 0x23;
18 |//Dirección I2C para el sensor BH1750
19 |byte buff[2];
20 |
21 |//----Inicio de variable existente (bomba de agua)----//
22 |int bomba_agua=D2;//Pin D2.
23 |int tierra=D4;//Pin D4.
24 |
25 |//-----Datos de la red WiFi a conectar-----//
26 |const char* red = "Pachirax-1";
27 |const char* pass = "pinguino01";
28 |
```

Figura 41. Programación de funciones de SIMORIA 1P2 en el entorno Arduino.

Finalmente, el dispositivo SIMORIA 1P2 es puesto en marcha (ver figura 42).



Figura 42. SIMORIA 1P2 puesto en marcha.

El servidor web seleccionado de acuerdo con los criterios empleados (ver tablas 16 y 17) fue Blynk como servidor primario, porque permite la visualización de datos en tiempo real (desde 1 hasta 3 segundos), y porque la aplicación homónima es compatible con iOS y Android. También fue escogido ThingSpeak como secundario, porque tiene una interfaz personalizable y las gráficas de valores son visibles y detalladas para los usuarios.

Para lograr una conexión óptima a Blynk, fue necesario crear una cuenta. Posteriormente, es nombrado el proyecto y seleccionada la tarjeta NodeMCU (ver figura 43).

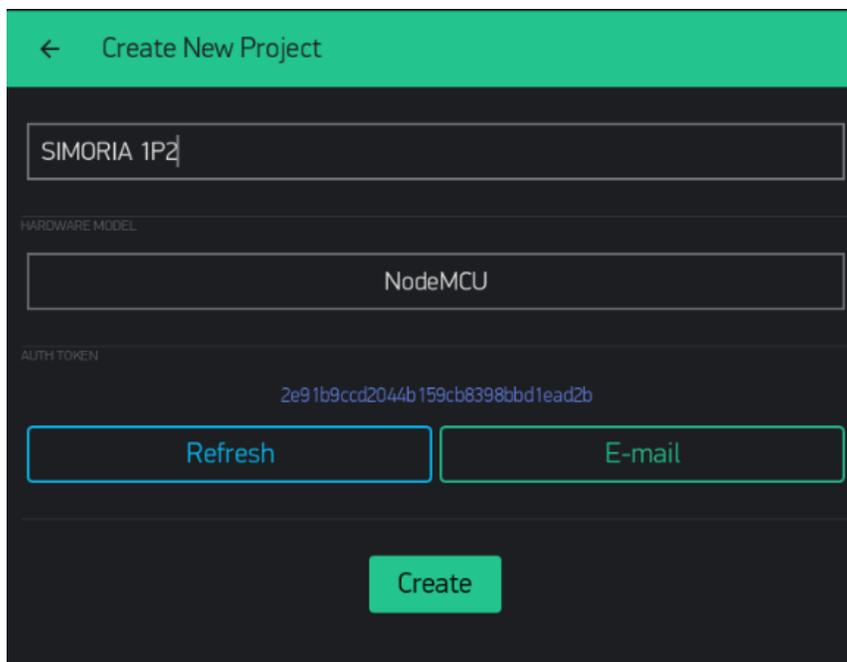


Figura 43. Interfaz de creación de proyectos en Blynk.

Después, es mostrada la interfaz inicial del proyecto creado (ver figura 44).

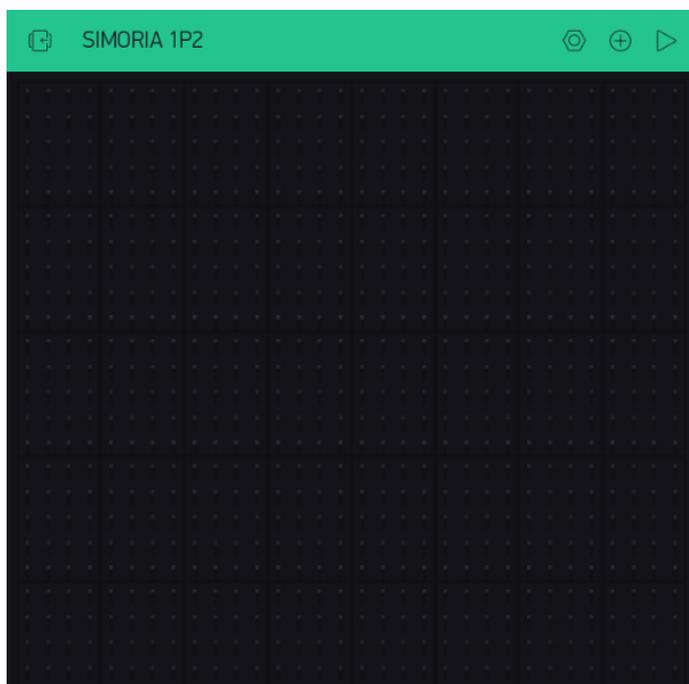


Figura 44. Interfaz del proyecto creado en Blynk.

Luego, fueron colocadas etiquetas de valores para permitir la visualización de datos obtenidos por los sensores conectados (ver figura 45).

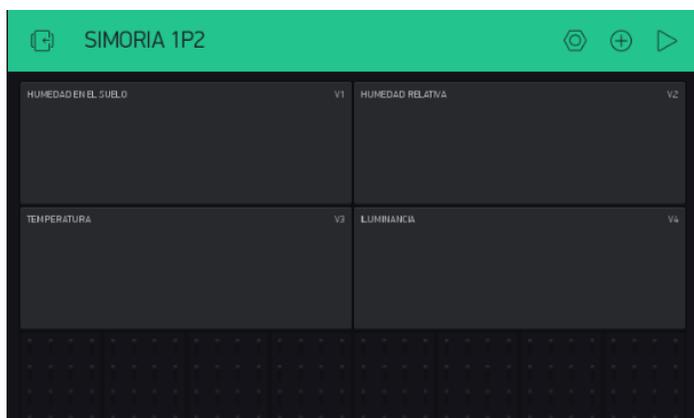


Figura 45. Valores etiquetados o *Labeled Values* en Blynk.

Finalmente, es agregado un *widget* llamado *WebHook*, que permite el envío de datos de Blynk a ThingSpeak (ver figura 46).

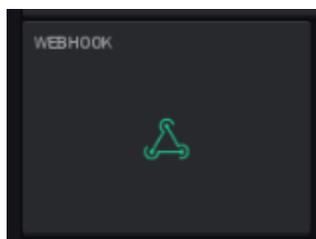


Figura 46. *Widget WebHook* en Blynk.

Para el envío de datos a ThingSpeak, fue necesario escribir el siguiente enlace:

«[https://api.thingspeak.com/update?api_key=clave_api_escritura&field1=/pin\[0\]/&field2=/pin\[1\]/&field3=/pin\[2\]/&field4=/pin\[3\]/](https://api.thingspeak.com/update?api_key=clave_api_escritura&field1=/pin[0]/&field2=/pin[1]/&field3=/pin[2]/&field4=/pin[3]/)» (Blynk 2016), y haber seguido los pasos de la sección 6.3.2.2. «/pin[0]/», «/pin[1]/», «/pin[2]/» y «/pin[3]/» son los valores asignados de cada sensor.

En el entorno de programación Arduino, los pines son representados como «Blynk.virtualWrite(V0, humedad_suelo, humedad_relativa, temperatura, iluminancia)»
 Los valores de humedad en el suelo son representados como «/pin[0]/», los de humedad relativa como «/pin[1]/», los de temperatura como «/pin[2]/» y los de iluminancia como «/pin[3]/».

Finalmente, es oprimido el botón triangular en Blynk (ver figura 47), y los datos son enviados automáticamente a ThingSpeak (ver figura 48), mientras esté conectado el dispositivo (ver figura 42).

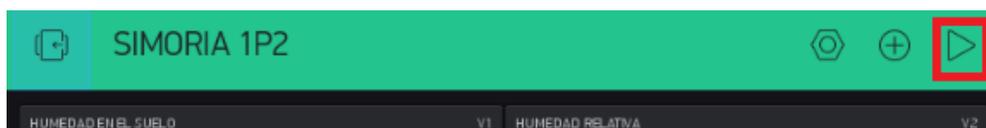


Figura 47. Botón comenzar (marcado) en el menú del proyecto creado en Blynk.



Figura 48. Recepción de Datos de Blynk en ThingSpeak.

8. Propuesta futura

8.1. Versión oficial de SIMORIA

Combina la rapidez de envío de dato a servidores web de la versión 1P2 con la exactitud y resistencia de sensores de la versión 1P.

Esta versión estaría diseñada con un módulo ESP8266-12E (módulo usado en tarjetas NodeMCU) (ver figura 49), un módulo relé (ver figura 15) y una regleta hembra con forma de «L» (ver figura 50) para permitir la conexión de sensores de humedad en el suelo capacitivo (ver figura 12), humedad relativa y temperatura DHT22 (ver Figura 13) y el de iluminación TSL2591 (ver figura 14), por su alto rango de obtención de datos (0-88,000 lx). Todos los componentes mencionados serían ensamblados en una baquelita de cobre (ver figura 51) para la formación del circuito electrónico.

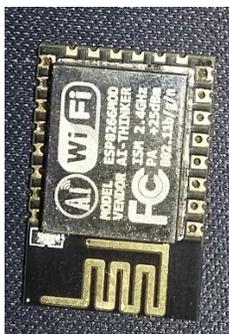


Figura 49. Módulo ESP8266-12E.



Figura 50. Regleta hembra con forma de «L».



Figura 51. Baquelita de cobre usada para el diseño de circuitos electrónicos.

8.2. Alimentación de SIMORIA por panel solar

La versión oficial es alimentada por un panel solar de 6V y 500 miliamperios conectado a una batería del mismo voltaje. Así, el dispositivo usa la energía de la batería en la noche, y es cargada de día.

8.3. Análisis de datos avanzado

En ThingSpeak, los datos podrán ser analizados de una mejor manera, como el cálculo de pronósticos para la predicción de condiciones ambientales en un futuro cercano o lejano.

9. Conclusiones

De acuerdo con el procedimiento llevado a cabo con ambos dispositivos, es posible observar que los sensores de humedad de bajo costo (ver figura 38) tienden a oxidarse rápido (en términos de semanas). Afortunadamente, este caso (para SIMORIA 1P2) fue una excepción que ha durado aproximadamente un año (antes del desarrollo de este dispositivo).

10. Referencias

- Adafruit. 2016. «Adafruit TSL2591 High Dynamic Range Digital Light Sensor». <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-tsl2591.pdf>.
- Aosong Electronics Co. 2016. «Digital-Output Relative Humidity & Temperature Sensor/Module AM2303». <https://cdn-shop.adafruit.com/datasheets/DHT22.pdf>.
- Arduino LLC. 2016a. «AnalogRead». *Arduino*. <https://www.arduino.cc/en/Reference/AnalogRead>.
- . 2016b. «Arduino Community Logo». *Arduino*. <https://www.arduino.cc/en/Trademark/CommunityLogo>.
- . 2016c. «Frequently Asked Questions». *Arduino*. <http://www.arduino.cc/en/Main/FAQ>.
- . 2016d. «Trademark». *Arduino*. <https://www.arduino.cc/en/Trademark/HomePage>.
- . 2016e. «What is Arduino?». *Arduino*. <https://www.arduino.cc/en/Guide/Introduction>.
- . 2016f. «Two Arduinos Become One». octubre 1. <https://blog.arduino.cc/2016/10/01/two-arduinos-become-one-2/>.
- Arduino SRL. 2016a. «Arduino Programming Language: AnalogRead». *Arduino*. <http://www.arduino.org/learning/reference/analogread>.
- . 2016b. «Copyright Notice». *Arduino*. <http://www.arduino.org/copyright-notice>.
- . 2016c. «What is Arduino». *Arduino*. <http://www.arduino.org/learning/getting-started/what-is-arduino>.
- . 2016d. «Two Arduinos Become One». septiembre 31. <http://www.arduino.org/blog/two-arduinos-become-one>.
- Bakker, Kees, Roman Messer, y Mike McPherson. 2016. «Sodaq DS3231». *GitHub*. abril 24. https://github.com/SodaqMoja/Sodaq_DS3231.
- Barath, Pratap Singh, Maitreyee Dutta, Ajay Chaudhary, y Madhu Sudan Jangid. 2014. «A Novel Adaptive Framework for Efficient and Effective Management of Water Supply System Using Arduino». En *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, 13:1–13:4. ICTCS '14. New York, NY, USA: ACM. doi:10.1145/2677855.2677868.
- Blynk. 2016. «WebHook». *Blynk*. <http://docs.blynk.cc/#widgets-other-webhook>.
- Brendan, Howell, Dirk van Osterbosch, y Lionel Michel. 2015. *DC Motor*. Multiplataforma. Fritzing. Friends of Fritzing.
- Bruneau, Richard. 2014. «Fritzing Parts - First Set». *Omnigatherum*. febrero 12. <http://omnigatherum.ca/wp/?p=6>.
- . 2015. «Fritzing Parts - Second Set». *Omnigatherum*. abril 16. <http://omnigatherum.ca/wp/?p=87>.
- Chun-zhi, H., X. Yin-shui, y W. Lun-yao. 2011. «A Universal Asynchronous Receiver Transmitter Design». En *Electronics, Communications and Control (ICECC), 2011 International Conference on*, 691-94. IEEE. doi:10.1109/ICECC.2011.6066542.
- Cinetica Tech. 2016. «ThingView Full». *Google Play*. septiembre 4. https://play.google.com/store/apps/details?id=com.cinetica_tech.thingview.full.

- Cortez, Pamela. 2014. «Make Your Own Fritzing Parts». *SparkFun*. marzo 18. <https://learn.sparkfun.com/tutorials/make-your-own-fritzing-parts#download-and-install>.
- Creative Commons. 2013a. «Attribution-NoDerivatives 4.0 International». <https://creativecommons.org/licenses/by-nd/4.0/>.
- . 2013b. «Attribution-ShareAlike 4.0 International». <https://creativecommons.org/licenses/by-sa/4.0/>.
- Cristian René Calderón Calderón. 2015. «Prototipo de sistema informático para la captura ambiental en espacios cerrados y monitoreo en tiempo real. Caso de estudio en la Facultad de Ingeniería de Sistemas (UNAB)». Universidad Autónoma de Bucaramanga.
- Dasios, Athanasios, Damianos Gavalas, Grammati Pantziou, y Charalampos Konstantopoulos. 2015. «Wireless Sensor Network Deployment for Remote Elderly Care Monitoring». En *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*, 61:1–61:4. PETRA '15. New York, NY, USA: ACM. doi:10.1145/2769493.2769539.
- Denso ADC. 2016. «QR Code Essentials». <http://denso-adc.com/download/whitepaper/1>.
- DiCola, Tony, Matthijs Kooijman, Limor Fried, Jose M. Dana, Phillip Burgess, Tim Forbes, Lars Sørup, Colin Miller, y Igor Scheller. 2016. «DHT Sensor Library». *GitHub*. mayo 27. <https://github.com/adafruit/DHT-sensor-library>.
- DiCola, Tony, Kevin Townsend, Limor Fried, Rick Lesniak, y 'Bbx10node'. 2016. «Adafruit TSL2591 Library». *GitHub*. mayo 27. https://github.com/adafruit/Adafruit_TSL2591_Library.
- Friends of Fritzing. 2016a. «Frequently Asked Questions». *Fritzing*. <http://fritzing.org/faq/>.
- . 2016b. «Fritzing». *Fritzing*. <http://fritzing.org/home/>.
- . 2016c. «People». *Fritzing*. <http://fritzing.org/about/people>.
- Gibb, Alicia, Steven Abadie, Ed Baafi, Matt Bolton, Kipp Bradford, Gabriella Levine, David A. Mellis, et al. 2015. *Building Open Source Hardware - DIY Manufacturing for Hackers and Makers*. Crawfordsville, Estados Unidos: Addison-Wesley. <https://books.google.com.co/books?id=wg27BQAAQBAJ&pg=PA132&lpg=PA132&dq=DIY+Manufacturing+for+Hackers+and+Makers+pdf&source=bl&ots=2Rg bx-h0im&sig=x-K9d0bfBO7cVjd4bZjXim3K9Yo&hl=en&sa=X&ved=0ahUKEwiFjc-ThOnKAhWJ6x4KHcHCBnkQ6AEILjAE#v=onepage&q=DIY%20Manufa>.
- GNU. 2016a. «GNU Lesser General Public License». *GNU Operating System*. agosto 6. <http://www.gnu.org/licenses/lgpl-3.0.html>.
- . 2016b. «GNU General Public License». *GNU Operating System*. agosto 7. <http://www.gnu.org/licenses/gpl-3.0.html>.
- Gómez Rodríguez, Atain. 2015. «Aspectos de organización, producción y comercialización del melón en la comarca lagunera». Saltillo, México: Universidad Autónoma Agraria Antonio Narro. <http://repositorio.uaaan.mx:8080/xmlui/bitstream/handle/123456789/4871/T13569%20G%C3%93MEZ%20RODR%C3%8DGUEZ%20ATAIN.%20%20TESIS.pdf?sequence=1>.

- Herstatt, Cornelius, y Daniel Ehls. 2015. *Open Source Innovation - The Phenomenon, Participant's Behavior, Business Implications*. Ciudad de Nueva York, Estados Unidos: Routledge.
http://samples.sainsburysebooks.co.uk/9781317624257_sample_951827.pdf.
- ITU. 2012. *Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks*. Génova, Suiza.
https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2060-201206-I!!PDF-E&type=items.
- Jindarat, S., y P. Wuttidittachotti. 2015. «Smart Farm Monitoring Using Raspberry Pi and Arduino». En *Computer, Communications, and Control Technology (I4CT), 2015 International Conference on*, 284-88. IEEE. doi:10.1109/I4CT.2015.7219582.
- Net Applications. 2016. «Mobile/Tablet Operating System Market Share». *NetMarketShare*. <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=1&qpcustomb=&qpsp=2016&qnp=1&qptimeframe=Y>.
- NXP Semiconductors. 2013. «I2C-Bus Components Selection Guide». http://www.nxp.com/documents/selection_guide/75017443.pdf.
- . 2016. «NFC Knowledge Base». <http://www.nxp.com/products/identification-and-security/nfc-and-reader-ics/nfc-technology-hub/nfc-knowledge-base:NFC-KNOWLEDGE-BASE>.
- Oxford Unity Press. 2016a. «Automation». *Oxford Dictionaries*. <http://www.oxforddictionaries.com/es/definicion/ingles/automation>.
- . 2016b. «Wi-Fi». *Oxford Dictionaries*. <https://en.oxforddictionaries.com/definition/wi-fi>.
- Paphitou, Athina C., Stella Constantinou, y Georgia M. Kapitsaki. 2015. «SensoMan: Remote Management of Context Sensors». En *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, 19:1–19:6. WIMS '15. New York, NY, USA: ACM. doi:10.1145/2797115.2797121.
- Rajeshwari, P., B. Shanthini, y Mini Prince. 2015. «Hierarchical Energy Efficient Clustering Algorithm for WSN». *IDIOSI Publication*, 108.
- Real Academia Española. 2016a. «Lux». *Diccionario de la Lengua española*. <http://dle.rae.es/?w=lux>.
- . 2016b. «Monitorizar». *Diccionario de la Lengua española*. <http://dle.rae.es/?w=monitorizar>.
- Shao, Xing, Cui-Xiang Wang, y Yuang Rao. 2015. «Network Coding Aware QoS Routing for Wireless Sensor Network». *Journal of Communications*, enero, 24.
- StatCounter. 2016. «Top 8 Mobile Operating Systems from Jan to Mar 2016». *StatCounter Global Stats*. http://gs.statcounter.com/#mobile_os-ww-yearly-2016-2016-bar.
- The Matworks Inc. 2016. «Update a Channel Feed». *MathWorks*. https://www.mathworks.com/help/thingspeak/update-channel-feed.html?requestedDomain=au.mathworks.com&requestedDomain=www.mathworks.com#update_feed_text.
- UNAB, UFPS, UCC, UNISANGIL, y UPB. 2016. «CIINATIC». *Arduino*. <http://congressociinatic2016.wixsite.com/ciinatic>.

Bucaramanga 10 de Mayo 2016

Referencia: autorización presentación trabajo de grado

Cordial saludo

Por medio de la presente autorizo la presentación del trabajo de grado I titulado: **Prototipo de sistema de monitorización y riego automático que mida la humedad y la temperatura para evaluar la vegetación.** Desarrollado por el estudiante **Frank Wilmar Santander González** perteneciente al programa de Ingeniería de Sistemas de la Universidad Autónoma de Bucaramanga.

Atentamente.



René Alejandro Lobo Quintero
Docente
Facultad de Ingeniería de Sistemas
Universidad Autónoma de Bucaramanga