UCLV
Universidad Central
"Marta Abreu" de Las Villas

FIE
Facultad de
Ingeniería Eléctrica

Departamento de Ingeniería Automática

TRABAJO DE DIPLOMA

Título: Sistema de supervisión y configuración local basado en IoT para casas de cultivo

Autor: Javier Lázaro Castellón Dorta

Tutores: Ms. C. Richar Sosa López

Dr. C. Iván Santana Ching

Ing. Diony Roely Castillo Rodríguez

Santa Clara, junio 2018

PENSAMIENTO

"Si tú no trabajas por tus sueños, alguien te contratará para que trabajes por los suyos."

Steve Jobs

DEDICATORIA

A mi hermanito Alejandro, al flaco, a mi abuelo Mongo y mi abuelo Luis.

AGRADECIMIENTOS

A mis padres, que lo han dado todo siempre por mí. Y aunque quisiera que no lo hicieran, no me van a hacer caso.

A Lauri, por aguantar lo inaguantable. Por entenderme.

A mis abuelas Olga y Nilda, porque las quiero.

A mis tutores, Ching y Richar, porque no hay mejor elección posible que ellos. Por toda su ayuda.

A Ale (Pita) y a Javier (el Mono). El primero, por ser un hermano más, y el segundo, porque sin beneficio alguno, me brindó su mano.

Al Guille, por estar siempre cuando uno lo necesita.

A mi abuelo Luis, mi primo Vladimir y mi padrino Héctor, porque fueron mi ejemplo para estudiar Ingeniería.

A mi primo Daniel, por todo lo que has hecho por mí.

A mi familia, porque se lo merecen.

A mis amigos, los nuevos y los viejos, porque siempre están en mi corazón.

A Abelito, porque me mata si no lo pongo en los agradecimientos.

RESUMEN

El desarrollo actual de nuevas tecnologías permite su aplicación a nuevas esferas de la sociedad, permitiendo el propio desarrollo de estas a un nivel sin precedentes. La implementación del Internet de las Cosas, las redes de sensores inalámbricos y circuitos integrados en la agricultura de precisión, eleva el ritmo de producción de las empresas agrícolas y genera un crecimiento económico y social. En este proyecto se realiza el diseño de un sistema de supervisión y configuración local basado en IoT para casas de cultivo. Este sistema permite la supervisión de parámetros ambientales medidos por una red de sensores inalámbricos y el levantamiento fenotípico de las cosechas. La arquitectura consta de una Raspberry PI como Gateway IoT que funciona como servidor de datos y una aplicación cliente de dispositivos móviles de la plataforma Android. El servidor fue programado en lenguaje Python, usando el protocolo TCP. Fueron realizadas pruebas de funcionalidad, conectividad, portabilidad, seguridad y rendimiento del sistema propuesto.

ABSTRACT

The current development of new technologies allows their application to new fields of society, allowing their own development to an unprecedented level. The implementation of the Internet of Things, wireless sensors networks and integrated circuits in precision agriculture, increases the production rate of agricultural companies and generates economic and social growth. In this project the design of an IoT based local supervision and configuration system for greenhouses is made. This system allows the monitoring of environmental parameters measured by a wireless sensors networks and the phenotypic survey of crops. The architecture consists of a Raspberry Pi as IoT Gateway that works as a data server and a client application for Android mobile devices. The server was programmed in Python language, using the TCP protocol. Tests of functionality, connectivity, portability, security and performance were carried out for the proposed system.

TABLA DE CONTENIDOS

INTRODUCCIÓN

Actualmente, los avances científicos forman parte cada vez más de la vida cotidiana. Esferas como las redes de comunicaciones, los dispositivos inteligentes y el Internet de las Cosas (IoT), han sentado las bases para un modelo de desarrollo sin precedentes. Una década atrás, para realizar operaciones de supervisión, control, procesamiento y adquisición de información, era necesario remitirse a una computadora personal, pero con el auge de los dispositivos móviles inteligentes, todas las operaciones pueden ser realizadas mientras exista alguna conexión a la red que ofrezca los servicios (Gubbi et al., 2013).

En el campo de la Ingeniería Automática, es cada vez más común la implementación de las redes de sensores inalámbricos y circuitos integrados potentes para dar solución a proyectos en la industria, la domótica y la agricultura de precisión. Se ha impulsado el desarrollo de aplicaciones de la plataforma móvil, capaces de mostrar, controlar e imponer valores de consignas en los distintos dispositivos (sensores y actuadores) de una red. Permitiendo de esta manera la comunicación desde distancias comprendidas dentro de la misma área local, hasta conexiones desde cualquier lugar del mundo a través de la red de Internet.

En la actualidad, la automatización aplicada a las cosechas ha logrado que la agricultura de precisión sea de interés cada vez más para las naciones y empresas multinacionales. El objetivo principal es contar con casas de cultivo, que constituyan instalaciones de producción de alta tecnología. Las cuales estén equipadas con sistemas de detección, calefacción, refrigeración, iluminación y riego. Además, puedan ser controladas por una computadora para optimizar las condiciones de crecimiento de las plantas. Para lograrlo, es necesario la aplicación de diferentes

técnicas para evaluar los grados de optimización y la relación de confort del microclima (temperatura del aire, humedad relativa, déficit de presión de vapor). De esta forma se reduce el riesgo de producción del cultivo de una planta específica (McBratney et al., 2005).

Cuba, apoyada en las universidades, sigue la estrategia de implementar nuevas y menos costosas tecnologías de automatización con el objetivo de dar un paso más en el camino hacia la independencia tecnológica (Zubizarreta Luján, 2017). La Empresa de Automatización Integral CEDAI ha emprendido un proyecto para acometer la reconversión tecnológica de las casas de cultivo en el país. Hasta el momento se han implementado sistemas de automatización que logran el suministro de los diferentes fertilizantes y componentes que necesitan las plantas con las dosis necesarias. Se instalan estaciones de bombeo de agua, estaciones de ferti-riego y paneles de interfaz de usuario para interacción con los operarios y los sistemas de control.

Esta investigación responde a una necesidad de la empresa CEDAI de Villa Clara de implementar un sistema de supervisión y configuración en la UEB de Cultivos Protegidos ¨Valle del Yabú¨ perteneciente a la misma provincia, que posea mayores prestaciones que los sistemas existentes en el país. De esta manera, se probaría la viabilidad y aplicabilidad del nuevo sistema para su posterior instalación en otras casas de cultivo.

El nuevo proyecto es una colaboración entre la empresa CEDAI y el Grupo de Internet de las Cosas, Automatización e Inteligencia Artificial perteneciente a la Universidad Central “Marta Abreu” de Las Villas. El alcance esperado comprende la instalación de un sistema que permita la supervisión de variables ambientales y la configuración del sistema. Además, se espera que los operadores interactúen con el sistema a través de una interfaz de usuario para dispositivos móviles. Se mantendrá también un control estricto del acceso de usuarios del sistema.

Una aplicación standalone (nativa) es la elección para dispositivos móviles debido a su estabilidad. Solo serían transmitidos los datos pertenecientes a valores de consigna y de variables, y no de información visual de la interfaz de usuario. Lo cual

asegura un mayor rendimiento y mejor aprovechamiento del ancho de banda. Además, no existiría dependencia hacia ningún navegador web (Jobe, 2013).

Situación del problema

La UEB de Cultivos Protegidos ¨Valle del Yabú¨ constituye una unidad productiva de referencia nacional, debido a sus excelentes resultados. Por tanto, es necesario registrar las condiciones bajo las cuales se obtienen los mismos. De esta forma crear una base digital de conocimientos sobre el proceso productivo. Dicha base, puede ser utilizada como referente para otras empresas agrícolas del país. Además, puede emplearse para la construcción de modelos de producción basados en inteligencia artificial.

Actualmente, el sistema automatizado existente en la empresa, cuenta con una HMI (interfaz humano – computador) que ofrece pocos datos de configuración y lectura de valores instantáneos de los sensores conectados a un PLC (Controlador Lógico Programable). Además, no existe una forma de controlar el acceso de los operarios y los cambios realizados al sistema.

El proyecto colaborativo, incluye el montaje de una red de sensores inalámbricos y la utilización de un modelo predictivo basado en inteligencia artificial, por tanto, es necesaria la supervisión de las nuevas variables medidas por la red de sensores y el levantamiento de las características fenotípicas de las cosechas. De esta manera contribuir a la gestión asistida por parte de los operarios de la agricultura, de las condiciones ambientales adecuadas dentro de las casas de cultivo. Además, se necesita una vía por la cual los especialistas introduzcan los levantamientos fenotípicos al modelo basado en inteligencia artificial. Para esta última actividad es necesario el desplazamiento del especialista de la agricultura por el interior de las casas de cultivo, por lo que se requiere de una interfaz de usuario móvil.

Por tanto, el problema científico de esta investigación es la no existencia de un sistema de supervisión y configuración local para dispositivos móviles del proceso automatizado de las casas de cultivo, que permita el control de acceso de usuarios, la supervisión simultánea de varios usuarios y el levantamiento fenotípico de las cosechas en la UEB de Cultivos Protegidos ¨Valle del Yabú¨.

Objetivo General:

Diseñar un sistema de supervisión y configuración local basado en IoT para casas de cultivo.

Objetivos Específicos:

- Identificar las variables objeto de supervisión y configuración del proceso productivo en las casas de cultivo.
- Determinar los requerimientos y especificaciones de diseño.
- Diseñar una aplicación móvil para la supervisión y configuración local de casas de cultivo.
- Diseñar la arquitectura de hardware y software del Gateway IoT para la comunicación entre los dispositivos.
- Valorar el desempeño del sistema desarrollado en el entorno de trabajo mediante pruebas de funcionamiento.

Con este trabajo se espera diseñar una propuesta de implementación de un sistema de supervisión y configuración local basado en IoT de casas de cultivos mediante el desarrollo de una aplicación cliente para dispositivos móviles. Para ello es necesario el empleo de un circuito empotrado que funcione como Gateway IoT (Puerta de enlace), la elección de la plataforma móvil y el desarrollo de un software de servidor que administre los datos del sistema.

Se pretende realizar una valoración de la funcionalidad de la aplicación móvil que permita un acceso seguro a los datos y parámetros de interés en el proceso.

Con la implementación de este sistema de supervisión y configuración local, los operarios de las casas de cultivo, así como directivos y especialistas de la empresa agrícola, tendrán acceso a las variables ambientales medidas por la red de sensores, control de usuarios y configuración de parámetros del proceso. Además, se elevarán las prestaciones de la interacción con el sistema y se contribuirá con la informatización de la sociedad.

Organización del informe

El informe de la investigación presentará la siguiente estructura:

En el Capítulo 1 se presentan los conceptos fundamentales acerca de las tecnologías IoT y la agricultura de precisión aplicadas a las casas de cultivo. Se hace una caracterización de los distintos dispositivos existentes capaces de realizar la función de Gateway IoT, las posibles estrategias de programación de la aplicación y los elementos fundamentales relacionados con la seguridad del sistema.

En el Capítulo 2 se muestran las técnicas metodológicas usadas para llevar a cabo el proyecto. Se identifican las variables objeto de supervisión y configuración del proceso, así como los requerimientos y especificaciones del sistema. Se selecciona y desarrolla la arquitectura de hardware y software necesaria para la implementación del proyecto.

En el Capítulo 3 se muestran los resultados obtenidos con la investigación y la propuesta de implementación de la misma mediante la valoración de la funcionalidad de la aplicación en tres casas de cultivo de la empresa.

Por último, se mostrarán una serie de conclusiones y recomendaciones sobre el proyecto realizado y se presentarán otros elementos anexos de interés.

CAPÍTULO 1. SISTEMAS DE SUPERVISIÓN Y CONFIGURACIÓN BASADOS EN IOT

El rápido avance de la tecnología tanto en la electrónica como en las comunicaciones, ha permitido la amplia propagación por el mundo del Internet de las Cosas como estrategia para el desarrollo. La vinculación de estas tecnologías con la automatización y el control es el enfoque de este capítulo, en el cual se presentará una revisión bibliográfica acerca de los conceptos y referentes teóricos fundamentales, así como antecedentes del presente trabajo, importantes para el desarrollo del mismo.

1.1 Las tecnologías IoT en la automatización

Internet de las Cosas (IoT) es la red de dispositivos físicos, vehículos, electrodomésticos y otros elementos integrados con componentes electrónicos, software, sensores, actuadores, que permite que estos objetos se conecten e intercambien datos. Cada “cosa” es identificable de forma única a través de su sistema informático integrado y puede operar dentro de la infraestructura existente de Internet (Atzori et al., 2010).

Desde su comienzo como Arpanet en 1969, Internet ha sido un importante factor de impulso para la industria de alta tecnología. El Internet de las Cosas es la próxima ola en la evolución de la red de redes. Comienza una nueva era de informática ubicua y comunicación, donde los sensores y dispositivos electrónicos inteligentes se están conectando a Internet a un ritmo acelerado. El objetivo es apoyar la realización de actividades de la vida diaria de manera fácil y natural usando información e inteligencia (Yeo et al., 2014).

Internet de las Cosas se basa en un modelo de referencia compuesto por cuatro capas (Ver Figura 1.1.):

- Capa de aplicación, la cual contiene las aplicaciones de IoT.
- Capa de Soporte de Servicio y Soporte de Aplicación, la cual se encarga del almacenamiento y procesamiento de los datos.
- Capa de Red, la encargada de las Capacidades de Transporte de los Datos y la conexión de red con la red empresarial o Internet.
- Capa de Dispositivos, compuesta por dispositivos que pueden conectarse directamente con la capa superior y dispositivos que necesitan conectarse a través de un Gateway (SECTOR and ITU, 2012).

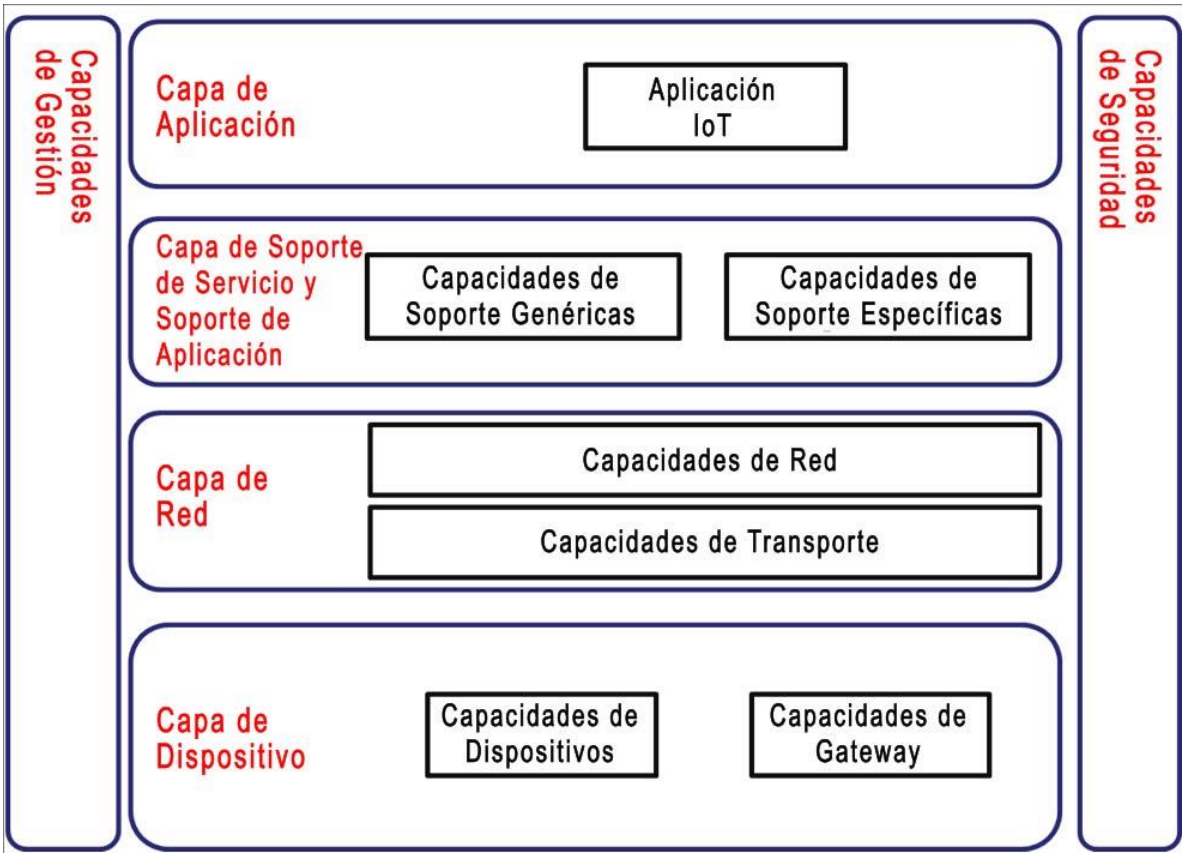


Figura 1.1. Modelo de referencia IoT (SECTOR and ITU, 2012).

Las Capacidades de Gestión de IoT cubren el manejo de fallas, administración de configuraciones, administración contable, gestión del rendimiento y administración

de seguridad. Mientras que las Capacidades de Seguridad se encargan de las Capacidades de Seguridad Genérica (atiende la seguridad en las capas de dispositivos, red y aplicación) y las Capacidades de Seguridad Específica, las cuales están estrechamente relacionadas con los requisitos específicos de la aplicación, como son el pago móvil y los requisitos de seguridad (SECTOR and ITU, 2012).

Este trabajo se basa en las cuatro capas del modelo de referencia IoT. En la Capa de Dispositivos de realizará el diseño de la arquitectura de hardware y software del Gateway IoT necesario para la implementación del sistema de supervisión y configuración. En la Capa de Red se encuentra el protocolo y el medio de comunicación entre el Gateway y los clientes de la red local. En la Capa de Soporte se hará uso del almacenamiento en base de datos. La Capa de Aplicación será el sistema final incluyendo las interfaces visuales. Por otra parte, con el diseño de una aplicación segura, se hará uso de las Capacidades de Seguridad.

1.1.1 Aplicaciones industriales

Comúnmente, las tecnologías IoT están integradas en aplicaciones de consumo, como hogares, automóviles y dispositivos portátiles inteligentes. Sin embargo, se prevé que las aplicaciones de IoT industrial (IIoT[1]), tendrán la capacidad de modificar la producción de sectores como la fabricación, el petróleo, el gas, la agricultura y la minería (Zhou et al., 2017).

Las IoT han estado ganando atracción en la industria logística, venta al por menor y productos farmacéuticos. Con los avances en comunicaciones inalámbricas, teléfonos inteligentes y tecnologías de redes de sensores, cada vez más “cosas” en la red u objetos inteligentes pasan a formar parte de las IoT. Como resultado, estas han tenido también un gran impacto en las nuevas tecnologías de información y comunicaciones (TIC), así como en las tecnologías de sistemas empresariales (Da Xu et al., 2014).

[1] Industrial IoT.

Los sensores integrados a una fábrica pueden ayudar a identificar anomalías en el desempeño del proceso de fabricación, resultando en la reducción del tiempo de fabricación y el desperdicio. En lugar de mantenimiento preventivo estándar, puede usarse mantenimiento predictivo. Debido a que permite que se realice el mantenimiento solo cuando las máquinas lo necesitan, reduciendo los costos totales y el tiempo que las máquinas permanecen inactivas (Zhou et al., 2017).

Existe un gran número de proyectos de desarrollo de aplicaciones de las IoT y la industria. En la Universidad de Jimei, China, se desarrolló un sistema IoT de supervisión del consumo de energía de edificios, donde los datos son recolectados por medidores inteligentes con puerto RS485, y luego los datos son transportados por protocolo TCP a un servicio de supervisión de energía para ser procesados y analizados. Utilizando el complemento MapXtreme[2], se incorporaron los datos de energía al mapa del edificio, para luego ser publicados en Internet a través de un sitio web (Wan et al., 2010).

Mientras que en la Universidad de Jiaotong, China, se propuso una aplicación IoT para su uso en el campo de la industria de la construcción. El estado de funcionamiento de cada grúa se detecta mediante un conjunto de sensores personalizados. Los datos de sensores son recopilados y procesados en la sala de operaciones del conductor. La comunicación inalámbrica entre las grúas y el sitio de trabajo en tierra se realiza a través de una red inalámbrica Zigbee. La supervisión remota se puede realizar utilizando una computadora personal o un teléfono inteligente (Zhong et al., 2014).

Otro proyecto, en la Universidad de Jilin, China, presentó un diseño para construir la plataforma de supervisión de calidad de todo el proceso de producción de alimentos con el uso de la tecnología IoT, que incluye la arquitectura y la estructura de funciones de la plataforma diseñada (Jia and Yang, 2011).

En el Instituto Indio de Ciencia y Tecnología Ingenieril (IIEST), Shibpur, India, se desarrolló una unidad de supervisión y control remoto para plantas solares

[2] Software de localización inteligente. Sitio oficial del fabricante: www.pitneybowes.com.

fotovoltaicas inteligentes basada en el modelo de referencia IoT de cuatro capas (Adhya et al., 2016).

Por otra parte, para la agricultura al aire libre, un ejemplo podría ser detectar la humedad del suelo y tener en cuenta el clima, de modo que los sistemas de riego inteligentes solo rieguen los cultivos cuando sea necesario, reduciendo la cantidad de agua utilizada. En la Universidad de Pekín se desarrolló una aplicación IoT basada en WebGIS[3] para un sistema de gestión de la agricultura de precisión para plataforma móvil (Ye et al., 2013).

Para la agricultura protegida, IoT permite el monitoreo y la gestión de las condiciones micro climáticas (humedad, temperatura, luz, etc.) para maximizar la producción (Zhao et al., 2010). Ejemplo de ello fue el método presentado por la Hebei University of Technology, China, para realizar la transmisión entre una red de sensores inalámbricos e Internet. El Gateway IoT se usa como parte del sistema de supervisión de una casa de cultivo, implementando acceso múltiple compatible con interfaces Ethernet, Wifi, GPRS, 3G y los datos pueden almacenarse localmente. Se utiliza como Gateway IoT el microcontrolador STM32[4]. La aplicación demuestra que el Gateway es confiable, compatible y extensible. Mediante el uso del Gateway, el sistema de supervisión logró la medición de los parámetros de interés y el control en tiempo real de la casa de cultivo (Li et al., 2014).

En la Universidad Agrícola de China, se diseñó un sistema de gestión de casas de cultivo que incluye tres partes: una red de sensores inalámbricos (WSN), una plataforma de servicios que opera en Internet y un sistema de apoyo para la toma de decisiones. El sistema permite una gestión de optimización de alta resolución en las casas de cultivo. Se mejora la eficiencia del proceso agrícola, desde los índices ambientales hasta la sostenibilidad económica (Li et al., 2012).

[3] Extensión a la web de las aplicaciones nacidas y desarrolladas para gestionar la cartografía numérica.
[4] Sitio oficial del fabricante: www.st.com.

1.2 Ámbito de aplicaciones de supervisión y configuración de sistemas de automatización

Acerca de este tema, se han realizado disímiles estudios sobre aplicaciones prácticas, arribando a conclusiones útiles para aquellos que pretenden profundizar en él. Ejemplo de ello es el sistema basado en aplicación para la regulación del nivel del agua en las plantas (Ver Figura 1.2.), desarrollado en el Shaheed Rajguru College, utilizando Raspberry Pi como Gateway IoT y sensores capacitivos de bajo costo para proporcionar un bajo consumo de energía y una solución rentable para el riego automático de plantas. El sistema está programado usando Python y una aplicación Android basada en Java donde la comunicación entre ellos se lleva a cabo mediante peticiones HTTP. Supervisando continuamente el estado del suelo, se pudo controlar el flujo de agua y así reducir el desperdicio. El diseño es de baja potencia, bajo costo, tamaño pequeño y robusto, evita el riego excesivo, bajo riego y reduce el desperdicio de agua (Chawla et al., 2016).

Figura 1.2. Esquema de arquitectura de hardware (Chawla et al., 2016).

En el Sri Jayachamarajendra College of Engineering, se diseñó un sistema de control de flujo de agua automatizado, donde la cantidad de agua consumida se calcula con la ayuda de un sensor de flujo y es enviada a la base de datos

centralizada. De esta forma se almacenan las estadísticas de uso de agua para cada casa y se pueden utilizar para un posterior análisis de datos, los cuales pueden ser mostrados en una aplicación para dispositivos móviles. El sistema utiliza como nodo sensor y actuador una placa Arduino Uno y utiliza una Raspberry Pi para enviar y recibir los datos hacia y desde la nube (Hegde et al., 2016).

Un proyecto desarrollado en la Anna University, implementó un control para robot utilizando una aplicación para la plataforma Android, programada utilizando el IDE Android Studio. Una placa Raspberry Pi es la encargada de actuar sobre los diferentes motores del robot mediante un driver LM293d[5]. Además, realiza el procesamiento de las imágenes tomadas por la cámara montada en ella mediante la biblioteca OpenCV[6]. Un teléfono inteligente con la aplicación de Android instalada puede conectarse vía WiFi con la Raspberry. Al presionar un botón en la aplicación de Android, se ejecuta un archivo **.php* seguido de una dirección IP para realizar una operación en la Raspberry. La actividad se ejecuta en segundo plano utilizando un objeto que hereda de la clase *AsyncTask* en la aplicación de Android, para evitar salir de la interfaz de usuario (Dheivamanogari and Uma, 2016).

Otro ejemplo es el diseño e implementación de un sistema de automatización del hogar basado en WiFi utilizando una estructura de red de sensores donde el nodo central es una PC de escritorio con software programado en Microsoft Visual Studio 2010, y los nodos sensores y actuadores son placas Arduino. El servidor es accesible desde una red local o desde Internet utilizando cualquier dispositivo, ya sea laptop o teléfono inteligente. (ElShafee and Hamed, 2012).

En la Institución Educacional Tecnológica de Atenas se desarrolló un sistema de automatización del hogar basado en aplicación Android y Web, utilizando la computadora empotrada Raspberry Pi como server, utilizando la librería de Java *WebSockets* standard que permite la comunicación de datos en tiempo real y simultánea con las aplicaciones Web y Android (Kehagias and Nini, 2015).

[5] Sitio oficial del fabricante: www.st.com.
[6] Biblioteca libre de visión artificial. Sitio oficial: opencv.org.

En la Koneru Lakshmiah University se implementó un sistema de gestión agrícola a través de sensores inalámbricos e IoT (Ver Figura 1.3.). La arquitectura de hardware está basada en Raspberry Pi conectada a una placa de sensores GrovePi+[7]. Para interactuar con el sistema se utiliza un dispositivo de usuario y se accede a través de Internet. El sistema cuenta con:

- Una unidad de recopilación de datos que comprende un conjunto de sensores inalámbricos para detectar actividades agrícolas y recopilar datos relacionados con parámetros agrícolas.
- Una unidad de estación base que comprende un registrador de datos, un servidor y una aplicación de software para procesar, recopilar y enviar los datos al dispositivo del usuario.
- El dispositivo de usuario (móvil o tableta), puede conectarse a una red de Internet, por lo que una aplicación móvil instalada en él, facilita la visualización de una lista de datos recopilados de los sensores (Navulur et al., 2017).

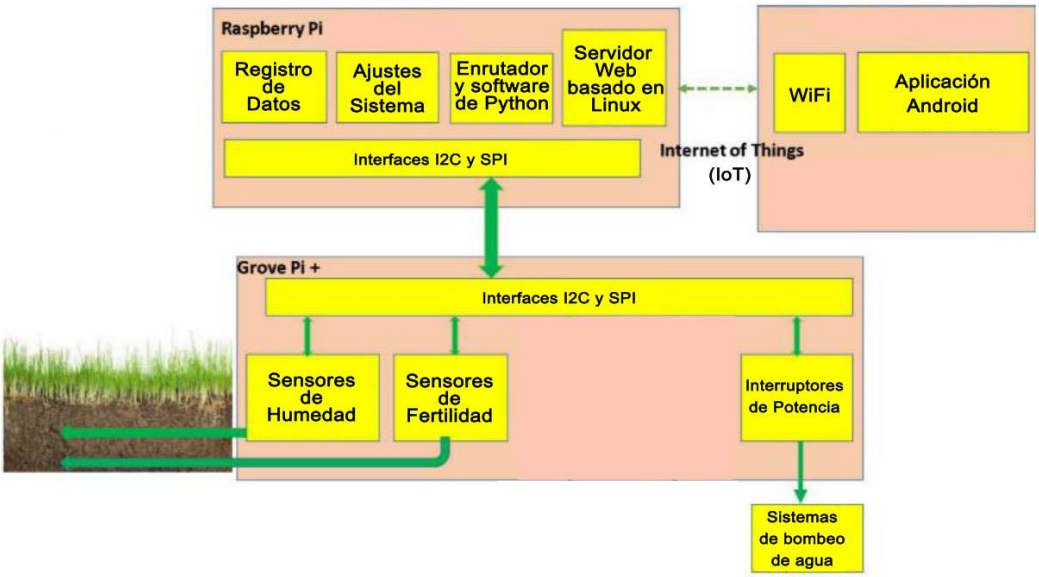


Figura 1.3. Esquema de arquitectura de hardware (Navulur et al., 2017).

[7] Placa adicional con 15 interfaces de 4 pines Grove, que lleva los sensores Grove a la Raspberry Pi. Sitio oficial: www.seeedstudio.com.

Un grupo de investigadores de la Universidad del Magdalena desarrolló una aplicación para teléfonos inteligentes para la supervisión y control de un proceso industrial en tiempo real a través una Raspberry Pi. Esta fue conectada a Internet utilizando el protocolo TCP. La aplicación fue probada en una planta piloto para verificar su correcto funcionamiento y rendimiento, demostrando su aplicabilidad a gran escala (Castro et al., 2016).

La Universidad Técnica de Ambato presidió el diseño de un sistema de monitoreo y control de cultivos utilizando Raspberry y Arduino como micro controladores (Rojas Pérez, 2015).

En el Doaba Institute of Engineering and Technology se elaboró el diseño de un sistema de monitoreo y control a distancia de variables ambientales mediante una red de sensores inalámbricos, utilizando Raspberry Pi como Gateway IoT y comunicación mediante protocolo Zigbee. La información de los sensores puede ser observada a través de una aplicación para dispositivos móviles (Singh and Singh, 2015).

Otro antecedente es el artículo que presenta un sistema de código abierto de bajo costo para la agricultura de precisión y controlada a distancia mediante una aplicación para dispositivos móviles. Se utilizan placas Arduino y Raspberry como unidades de procesamiento y comunicación mediante protocolo HTTP (Cimino et al., 2016).

En el Priyadarshini College of Engineering se automatizó un sistema de riego utilizando Arduino como nodo sensor y supervisión a distancia a través de una aplicación para la plataforma Android (Bawane et al., 2017).

También en el College of Technology, Pantnagar, India, se desarrolló un sistema de supervisión remoto en tiempo real de celdas solares utilizando tecnología IoT y

Raspberry Pi 3 como Gateway IoT. Los datos recopilados son enviados a través de Internet y visualizados mediante el software LabView[8] (Gupta et al., 2017).

Otro proyecto, esta vez en la Universidad SRM, India, implementó un sistema inteligente de irrigación automatizada basado en IoT, en el cual los datos de humedad de suelo y temperatura de los sensores, son usados por un algoritmo de aprendizaje para realizar predicciones en el riego. El sistema emplea Raspberry Pi y Arduino. Todos los datos son supervisados a través de Internet empleando tecnología IoT (Shekhar et al., 2017).

En Dallas, Texas, Estados Unidos, se diseñó e implementó un sistema de supervisión y control remoto para casas de cultivo utilizando un teléfono inteligente y una computadora con Internet. Este sistema guarda automáticamente los datos recopilados y está compuesto por una unidad de sensor, una unidad terminal remota(RTU), un servidor local, un panel de control y cámaras. El servidor se comunica con la unidad de sensor y la RTU mediante protocolo Zigbee. Los datos recopilados pueden observarse en una página web desde una PC o teléfono inteligente. Diariamente se guardan los datos en una base de datos. Ante condiciones atípicas, el servidor envía un SMS o un correo electrónico a los dispositivos de usuarios (Kim et al., 2012).

Otra investigación relacionada propone el sistema de supervisión y adquisición de datos en tiempo real de parámetros agrícolas y medioambientales usando redes de sensores inalámbricos y aplicaciones Web para la plataforma Android. El microcontrolador utilizado en las unidades de medición es el ATmega2560[9]. Los datos recopilados de temperatura, pH, humedad relativa y del suelo, son almacenados en un servidor HTTP. Los datos se pueden observar desde cualquier aplicación Web de Android desde un teléfono inteligente (Lukman, 2017).

[8] Plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico.
[9] Sitio oficial del fabricante: www.atmel.com.

1.3 Plataforma móvil

La utilidad de un teléfono inteligente o *smartphone* está dada por su portabilidad, la integración que presenta con diversos sensores y las capacidades tan amplias de conectividad. La selección de la plataforma móvil a utilizar en el proyecto, basada en el sistema operativo que presentan, es de vital importancia. Existen varios sistemas operativos para dispositivos móviles inteligentes, aunque los más usados son Android e iOS (“Gartner,” 2017).

Android es un sistema operativo móvil desarrollado por Google, basado en el kernel de Linux, diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes y tabletas. Además, Google ha desarrollado Android TV para televisores, Android Auto para automóviles, Wear OS para relojes de pulsera y Android Things para dispositivos IoT. Las variantes de Android también se usan en consolas de videojuegos, cámaras digitales, PC y otros dispositivos electrónicos. Posee la ventaja sobre otras plataformas de que las aplicaciones pueden ser desarrolladas sobre cualquiera de los grandes sistemas operativos como Windows, Mac OS X o Linux, ya que los entornos de desarrollo tienen versiones para cada uno de ellos (Goadrich and Rogers, 2011).

Un teléfono inteligente utilizando el sistema operativo Android es ideal para aplicaciones basadas en el modelo de referencia del Internet de las Cosas. Puede tener conectividad mediante Bluetooth, WiFi, NFC y datos móviles de alta velocidad. Las aplicaciones desarrolladas pueden utilizar SQLite para el almacenamiento de los datos. Tiene soporte web y multimedia, y posee un conjunto de sensores y hardware potentes que lo convierten en un dispositivo inteligente y lo hacen capaz de situarse tanto en la capa de dispositivo como en la capa de red y de aplicación del modelo de referencia IoT (Gronli et al., 2014).

Por su parte, iOS, es un sistema operativo móvil creado y desarrollado exclusivamente para el hardware de Apple Inc. Está presente en muchos de los dispositivos fabricados por la compañía, como son el iPhone y el iPad. Es el segundo sistema operativo móvil más popular después de Android (“Gartner,” 2017) (Ver Figura 1.4.). Está basado en el kernel Darwin. Es un sistema muy robusto,

aunque presenta una dificultad en cuanto al desarrollo de aplicaciones para esta plataforma. Solo los usuarios de Mac pueden descargar y usar el Kit de Desarrollo de Software (SDK) para desarrollar aplicaciones. Esta es una gran desventaja respecto a Android, el cual posee una gran comunidad de desarrollo. En iOS, la multitarea no está completamente disponible para el usuario. Además, este sistema operativo no es de código abierto (“Apple Developer,” 2017).

Operating System	2017 Units	2017 Market Share (%)	2016 Units	2016 Market Share (%)
Android	1,320,118.1	85.9	1,268,562.7	84.8
iOS	214,924.4	14.0	216,064.0	14.4
Other OS	1,493.0	0.1	11,332.2	0.8
Total	**1,536,535.5**	**100.0**	**1,495,959.0**	**100.0**

Figura 1.4. Ventas mundiales de teléfonos inteligentes a usuarios finales por sistema operativo en 2017 (miles de unidades) (“Gartner,” 2017).

1.4 Interconexión entre dispositivos

En el ámbito de las IoT, la conexión de los diferentes dispositivos inteligentes a la red empresarial o a Internet, representa parte fundamental de su funcionamiento. La interconexión de máquina a máquina (M2M), y de interfaces de red de alta velocidad con los dispositivos de campo (sensores, actuadores, microcomputadoras y otros dispositivos inteligentes), constituyen la capa más baja sobre la cual se sostiene el desarrollo de estas tecnologías. En una segunda capa, se encuentran los protocolos de comunicación que permiten la interconexión segura de estos dispositivos.

Es entonces necesaria, la adecuada selección de la arquitectura de hardware para Gateway IoT y la arquitectura de software necesaria para asegurar los requerimientos y especificaciones del sistema a desarrollar.

1.4.1 Sistemas empotrados como Gateway

El Internet de las Cosas es un concepto de rápido crecimiento que constituye la revolución tecnológica, empleando conexiones M2M y sistemas empotrados funcionando como Gateway IoT. La popularidad de las plataformas educativas basadas en código abierto está en aumento y representa componentes comerciales listos para ser adquiridos fácilmente que pueden ser utilizados en el diseño de prototipos para crear dispositivos IoT (Chen et al., 2011).

Han aparecido en el mercado dispositivos empotrados capaces de preparar la base para la comunicación entre la interfaz de entrada y salida (I/O) y la interfaz de cara a Internet, capaces de realizar la función de Gateway y que a la vez resultan una solución económica para tareas de automatización utilizando las tecnologías IoT. Ejemplo de ellos son los circuitos empotrados Raspberry Pi, Arduino y BeagleBoard, en todas sus versiones y teniendo en cuenta además la gran cantidad de módulos y accesorios fabricados para ellos (Hodges et al., 2013).

Arduino es una compañía de hardware y software de código abierto y una comunidad de usuarios que diseña y fabrica microcontroladores de placa única y kits de microcontroladores para construir dispositivos digitales y objetos interactivos (Ver Figura 1.5.). Logrando que estos puedan detectar y controlar objetos en el mundo físico y digital. Los productos del proyecto se distribuyen como hardware y software de código abierto, licenciados bajo la Licencia Pública General Reducida de GNU (LGPL) o la Licencia Pública General de GNU (GPL), que permiten la fabricación de placas Arduino y distribución de software por parte de cualquier persona. Las placas Arduino están disponibles comercialmente en forma pre ensamblada, o como kits DIY[10] (“hazlo tú mismo”) (“Arduino,” 2017).

[10] DIY: Del inglés “*do-it-yourself*”. Práctica de la fabricación o reparación de cosas por uno mismo.

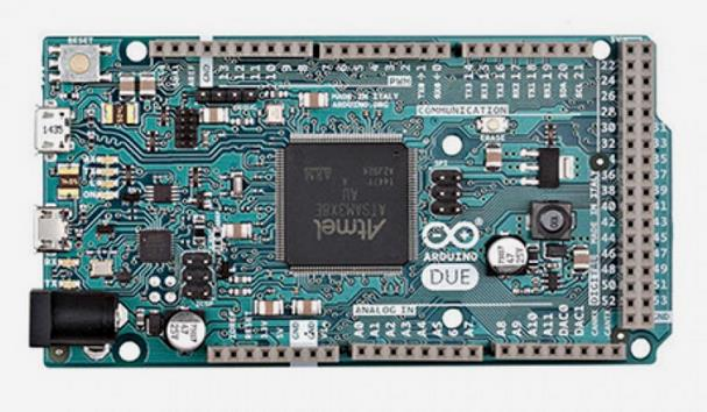


Figura 1.5. Arduino Due (“Arduino,” 2017).

Los diseños de placas Arduino usan una variedad de microprocesadores y controladores. Las tarjetas están equipadas con conjuntos de pines de I/O digitales y analógicas que pueden interconectarse con varias tarjetas de expansión o placas (*shields*) y otros circuitos. Las tarjetas cuentan con interfaces de comunicaciones seriales, que incluyen USB (Universal Serial Bus) en algunos modelos, que también se utilizan para cargar programas desde computadoras personales. Los microcontroladores son típicamente programados usando un dialecto de características de los lenguajes de programación C y C ++. Además de utilizar las cadenas de herramientas de compilación tradicionales, el proyecto Arduino proporciona un entorno de desarrollo integrado (IDE) basado en el proyecto de lenguaje de procesamiento (“Arduino,” 2017).

La Raspberry Pi es una serie de pequeñas computadoras de placa única desarrolladas en el Reino Unido por la Fundación Raspberry Pi para promover la enseñanza de la informática básica en las escuelas y en los países en desarrollo (Ver Figura 1.6.). Estas computadoras tienen propiedad registrada, pero de uso completamente libre. Se han lanzado varias generaciones de Raspberry Pi, todos los modelos cuentan con un sistema Broadcom en un chip (SoC) con una unidad de procesamiento central (CPU) integrada compatible con ARM y una unidad de procesamiento de gráficos en el chip (GPU) (“Raspberry Pi - Teach, Learn, and Make with Raspberry Pi,” 2018).

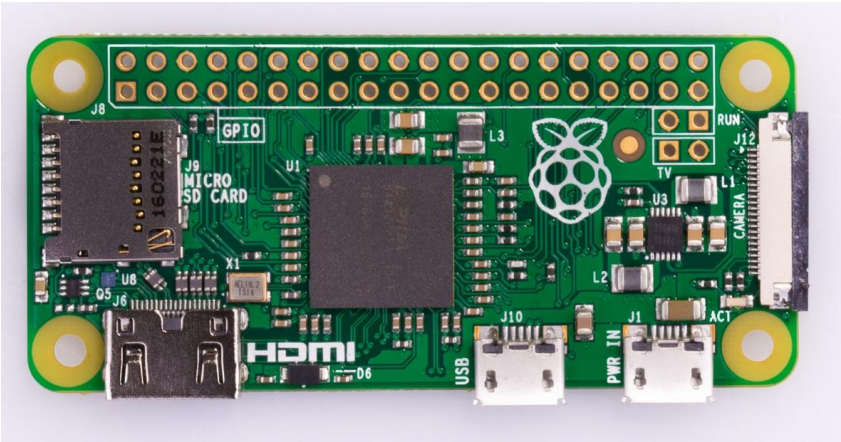


Figura 1.6. Raspberry Pi Zero (“Raspberry Pi - Teach, Learn, and Make with Raspberry Pi,” 2018).

Los modelos B y B+ están compuestos por CPU/GPU, RAM, pines I/O (entradas y salidas), puertos USB y Ethernet. Los modelos A, A+ y Pi Zero son similares, pero carecen de puertos USB y Ethernet. El adaptador Ethernet está conectado internamente a un puerto USB adicional. En el Modelo A, A + y Pi Zero, el puerto USB se conecta directamente al SoC. En los modelos Pi 1 Modelo B + y posteriores, el chip USB/Ethernet contiene un concentrador USB de cinco puntos, de los cuales cuatro puertos están disponibles, mientras que la Pi 1 Modelo B solo proporciona dos. En el Pi Zero, el puerto USB también está conectado directamente al SoC, pero usa un puerto micro USB (OTG) (“Raspberry Pi - Teach, Learn, and Make with Raspberry Pi,” 2018).

BeagleBoard es una placa reducida de código abierto de bajo producida por Texas Instruments en asociación con Digi-Key y Newark element14 (Ver Figura 1.7.). El BeagleBoard también se diseñó pensando en el desarrollo de software de código abierto y como una forma de demostrar el sistema OMAP3530 de Texas Instruments en un chip. La placa fue desarrollada por un pequeño equipo de ingenieros como una computadora empotrada educativa que podría ser utilizada en universidades de todo el mundo para enseñar capacidades de hardware y software de código abierto. También se vende al público bajo la licencia de Creative Commons. La placa se

diseñó utilizando Cadence OrCAD[11] para los esquemas y Cadence Allegro[12] para la fabricación de PCBs[13]. No se utilizó ningún software de simulación (He et al., 2016).

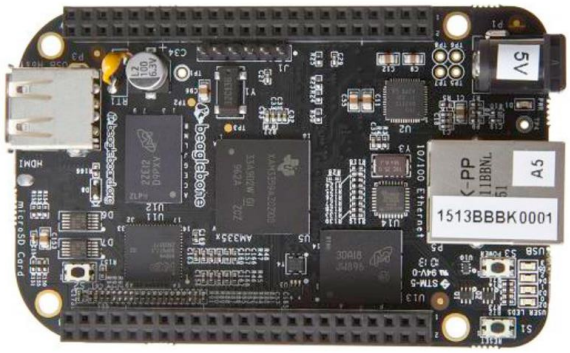


Figura 1.7. BeagleBone Black (“BeagleBoard.org,” 2017).

Un hardware Gateway IoT (Ver Figura 1.8.) se compone de un procesador o microcontrolador, sensores, circuitos de protección, interfaces de I/O hacia los sensores, como son Zigbee, 6LoWPAN, EnOcean, BLE, Modbus, PROFIBUS, I2C, Serie (Al-Sarawi et al., 2017). Además, posee interfaces hacia la red empresarial, como son, Bluetooth, WiFi, Ethernet, etc. El tipo de hardware, la velocidad de procesamiento y el espacio de memoria se deciden según el sistema operativo del dispositivo Gateway (Murikipudi et al., 2015). La aplicación de usuario también tiene una gran influencia en el diseño del hardware IoT. Si la aplicación es de escala simple a media, se utiliza un sistema operativo en tiempo real, y si el Gateway tiene que realizar operaciones considerablemente complejas, entonces se utiliza un sistema operativo visual. Una aplicación de pequeña a mediana escala puede ejecutarse en un microcontrolador; sin embargo, si se espera que el Gateway realice operaciones complejas, se necesita un microprocesador. Esto genera un gran

[11] Software propietario utilizado para automatización de diseño electrónico.
[12] Herramientas para codiseño de circuitos integrados, paquetes y PCBs.
[13] PCB: Del ingés *Printed Circuit Board o* Placa de Circuito Impreso.

impacto sobre el costo del dispositivo que funciona como Gateway IoT (Lee et al., 2017).

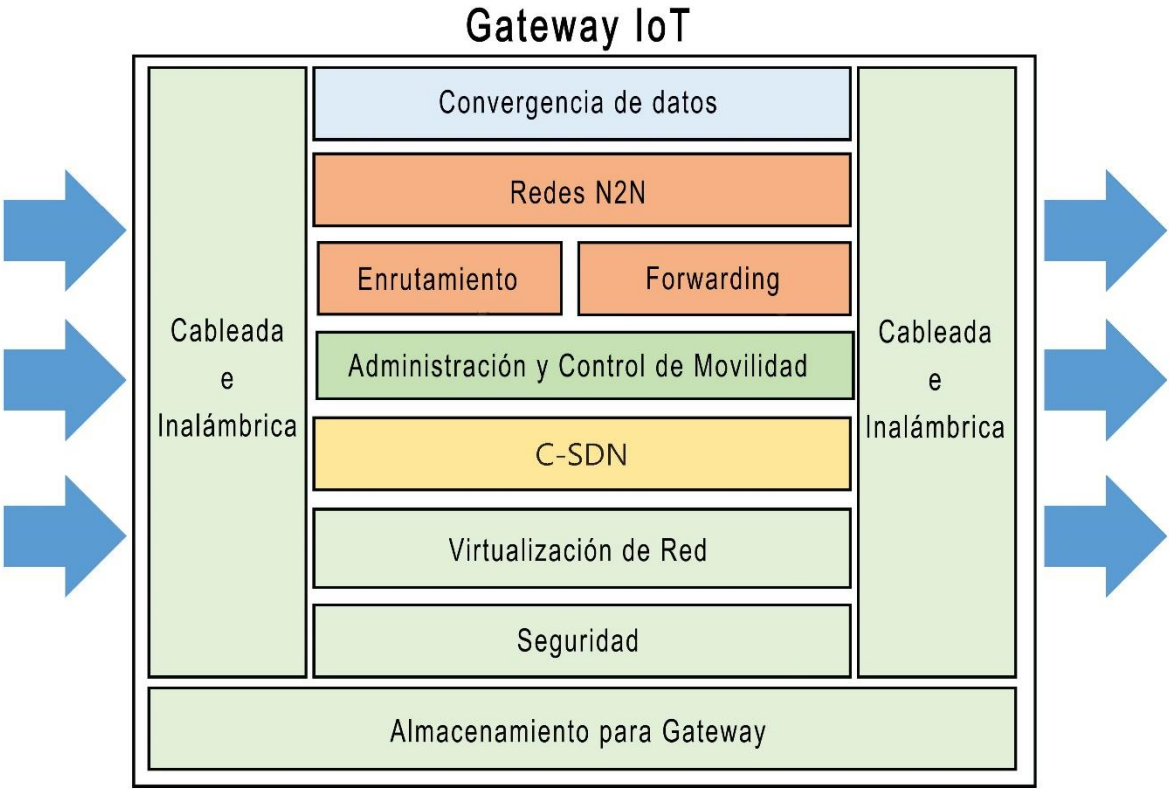


Figura 1.8. Componentes de un Gateway IoT (Lee et al., 2017).

En este contexto los sensores recopilan datos y estos datos deben almacenarse en una memoria temporal en la microcomputadora (Gateway) antes de transmitirse a la nube o Internet. Luego puede haber instrucciones desde la microcomputadora o desde la nube (la aplicación) al actuador o sensor para realizar una acción. El sensor sigue transmitiendo datos a la aplicación, lo que a su vez puede generar inteligencia con esos datos y, por lo tanto, desencadenar algunas acciones basadas en la intuición (Atzori et al., 2010).

1.4.2 Interfaces hacia Internet y protocolos de comunicación

El entorno IoT consta de una gran cantidad de dispositivos inteligentes, pero con muchas limitaciones. El volumen de almacenamiento, la vida útil y la distancia de conectividad están entre estas restricciones. Por lo tanto, la implementación de IoT

requiere un protocolo de comunicación que pueda manejar eficientemente estas condiciones (Al-Sarawi et al., 2017).

El medio físico de comunicación por excelencia para los elementos de la capa de dispositivos del modelo de referencia IoT, es el inalámbrico, utilizando la conexión WiFi.

La tecnología WiFi es la marca que representa al protocolo IEEE 802.11, el cual constituye un conjunto de especificaciones de control de acceso a medios (MAC) y capa física para implementar comunicación de red inalámbrica de área local (WLAN) en las bandas de frecuencia de 900 MHz, 2.4 GHz, 3.6 GHz, 5 GHz y 60 GHz. Estos son los estándares de red de computadoras inalámbricas más utilizados en el mundo, siendo usados en la mayoría de las redes domésticas y de oficina. Permitiendo que las computadoras portátiles, impresoras y teléfonos inteligentes se comuniquen entre sí y accedan a Internet sin necesidad de conectar cables. Está disponible en la placa Arduino mediante módulos accesorios y en la Raspberry viene incorporada en la placa en algunos modelos (Crow et al., 1997).

En la interfaz de red WiFi de cara a Internet, una vez establecida una conexión física, es necesario emplear protocolos de comunicación de la capa de transporte del modelo OSI, como los protocolos TCP y UDP. Ya en la capa de aplicación pueden aplicarse protocolos como MQTT, HTTP, CoAP, que aseguran la presentación y manejo adecuado de la información a través de la red.

El protocolo TCP (Protocolo de Control de Transmisión) es el encargado de tomar la información que se desea transmitir por la red y dividirla en paquetes o segmentos. Enumera los paquetes para la posterior verificación de la información. El destinatario extrae la información de dichos paquetes y la ordena adecuadamente. Si algún paquete se pierde durante la transmisión, el receptor solicita su retransmisión al emisor. Una vez la información llega al destinatario en el orden adecuado, es transportada a la aplicación que esté utilizando los servicios de red (Fernández Barcell, 2014).

El Protocolo de transferencia de hipertexto (HTTP) es un protocolo de nivel de aplicación para sistemas de información. HTTP permite un conjunto abierto de

métodos y encabezados que indican el propósito de una solicitud. Se basa en la disciplina de referencia proporcionada por el Identificador Uniforme de Recursos (URI), como una ubicación (URL) o nombre (URN), para indicar el recurso al que se aplicará un método (Fielding et al., 1999).

El protocolo MQTT consiste en mensajes de publicación y suscripción construido sobre el Protocolo de control de transmisión (TCP) y diseñado para ser liviano. Además, MQTT admite mensajería sin conexión para manejar clientes desconectados. MQTT se encuentra estandarizado dentro del consorcio Advancing Open Standard for the Society (OASIS) (Collina et al., 2014).

CoAP es un protocolo de petición y respuesta que sigue de manera flexible el protocolo de transferencia de hipertexto (HTTP), pero a través del protocolo de datagramas de usuario (UDP) en lugar de TCP. Al igual que HTTP, CoAP se está estandarizando dentro del Grupo de trabajo de ingeniería de Internet (IETF). Para garantizar la entrega del mensaje, CoAP presenta un mecanismo de retransmisión basado en retroceso exponencial y un máximo de cuatro intentos de retransmisión (Al-Fuqaha et al., 2015).

1.4.3 Seguridad

La principal diferencia entre las redes cableadas e inalámbricas es la forma en que transmiten los datos. En cuanto a los riesgos de seguridad, la principal diferencia entre las redes cableadas e inalámbricas es cómo acceder a los datos transmitidos. En las redes cableadas esto solo es posible tocando el medio que se usa para la red de comunicación. En redes inalámbricas, el medio utilizado para la comunicación es el aire. Los datos transmitidos a través de radio frecuencias pueden ser recibidos por equipos con la capacidad de interpretarlas, como computadoras personales y teléfonos inteligentes. Por tanto, es de vital importancia la protección de estos datos ante ataques y robo de información (Bulbul et al., 2008).

La primera solución a la seguridad inalámbrica, es la protección de la conexión a la red, mediante el estándar Wi-Fi Protected Access II (WPA2), protocolo de seguridad y programa de certificación de seguridad desarrollado por WiFi Alliance para asegurar redes de computadoras inalámbricas en respuesta a las serias

deficiencias encontradas en el sistema anterior, Wired Equivalent Privacy (WEP). La tecnología WPA incluye principalmente tres mejoras sobre WEP: cifrado de datos mejorado, autentificación de usuario e integridad (Bulbul et al., 2008).

La segunda solución sería la configuración de un filtraje de la dirección física (MAC) de los dispositivos que pueden conectarse a la red. En tercer lugar la implementación del protocolo de encriptación TLS/SSL en la capa de transporte del modelo OSI (Rescorla, 2001), asegurando la conexión de sockets TCP. O en la capa de aplicación utilizando el mismo sistema de encriptación sobre el protocolo HTTP para así utilizar HTTPS o HTTP seguro. En la misma capa del modelo OSI, TLS/SSL se utiliza para encriptar la conexión mediante el protocolo MQTT para lograr el llamado Secure-MQTT (MQTT seguro) (Singh et al., 2015).

Transport Layer Security (TLS) y su predecesor, Secure Sockets Layer (SSL), son protocolos criptográficos que proporcionan seguridad de comunicaciones a través de una red informática. Varias versiones de los protocolos encuentran uso generalizado en aplicaciones tales como navegación web y servicios de red. Los sitios web y aplicaciones en una red pueden usar TLS para asegurar todas las comunicaciones entre sus servidores y navegadores web (Dierks, 2008).

Normalmente, el servidor debe elegir una versión de TLS/SSL particular, y el cliente debe adaptarse a la elección del servidor. La mayoría de las versiones no son interoperables con las otras versiones (“3.6.5 Documentation,” 2017).

Según (Dierks, 2008),el protocolo de registro TLS se utiliza para la encapsulación de varios protocolos de alto nivel. Uno de estos protocolos encapsulados, el TLS Handshake Protocol, permite que el servidor y el cliente se autentiquen entre sí y negocien un algoritmo de cifrado y claves criptográficas antes de que el protocolo de la aplicación transmita o reciba su primer byte de datos. El protocolo TLS Handshake proporciona seguridad de conexión que tiene tres propiedades básicas:

- La identidad del par se puede autenticar usando criptografía asimétrica o de llave pública (RSA). Esta autenticación puede ser opcional, pero generalmente es necesaria para al menos uno de los pares.

- La negociación de un secreto compartido es segura: el secreto negociado es mucho menos vulnerable a ataques del tipo “man in the middle” (hombre en el medio).
- La negociación es confiable: ningún atacante puede modificar la comunicación de negociación sin ser detectado por las partes de la comunicación.

Por otra parte, el método de encriptación RSA consiste en la supuesta dificultad de factorizar enteros grandes (factorización de enteros). Se considera que el descifrado completo de un texto cifrado RSA no es viable suponiendo que no exista un algoritmo eficiente para la factorización de enteros. Un usuario de RSA crea y luego publica el producto de dos números primos grandes, junto con un valor auxiliar, como su clave pública. Los factores primos deben mantenerse en secreto. Cualquiera puede usar la clave pública para encriptar un mensaje, pero solo alguien con conocimiento de los factores primos puede decodificar el mensaje de manera factible (Singh, 2013).

1.5 Conclusiones del capítulo

Las aplicaciones de supervisión y configuración de sistemas automatizados aplicadas a la agricultura de precisión constituyen un campo de investigación de gran importancia para el desarrollo tecnológico y económico de la sociedad. La interconexión de estas aplicaciones con los dispositivos móviles de la vida cotidiana del hombre, basada en el modelo de referencia IoT, permite un mayor aprovechamiento de las ventajas que representa este modelo en cuanto a eficiencia, toma de decisiones, versatilidad y escalabilidad. En el próximo capítulo se realizará la selección de los protocolos de comunicación de cara a la red empresarial, el hardware que servirá como Gateway IoT y la gestión de la seguridad. De esta forma lograr el diseño del sistema final de supervisión y configuración en concordancia con los objetivos de esta investigación.

CAPÍTULO 2. DISEÑO DEL HARDWARE Y EL SOFTWARE DEL SISTEMA

En el ámbito de este trabajo, la selección de la estructura física y lógica del sistema automatizado en relación estricta con los requisitos establecidos por el cliente que contrata servicios, constituye parte crucial del proceso de desarrollo del proyecto. En este capítulo se aborda lo relacionado a la propuesta de arquitectura de hardware y software necesaria para el diseño de la aplicación de supervisión y configuración para la plataforma Android. Se establece la relación entre los elementos técnicos y de software necesarios para el correcto funcionamiento de la misma. Entre estos se encuentran el Gateway IoT, el servidor de datos y sus configuraciones correspondientes.

2.1 Requerimientos y especificaciones del proceso

Las primeras dos etapas de un proyecto de automatización son las encargadas de la definición de los requerimientos y las especificaciones de hardware y la configuración del sistema (Izaguirre, 2012). Para efectuar esta tarea fue necesario en primer lugar, la realización del levantamiento instrumental del área de la empresa agrícola en la cual se pretende instalar el nuevo sistema de automatización. Actualmente en el área de casas de cultivo, está instalado un PLC que controla las operaciones de ferti-riego, el cual se conecta a una HMI mediante puerto Serie. A través de la interfaz, los operarios pueden solamente configurar parámetros referentes al PLC y visualizar algunos parámetros ambientales en su medición instantánea.

Debido a la contribución de este trabajo y su posible integración con otros proyectos llevados a cabo por CEDAI en la UEB de Cultivos Protegidos “Valle del Yabú”, relacionados con las redes de sensores inalámbricos y la inteligencia artificial, fueron necesarias las siguientes especificaciones del sistema:

- La aplicación debe requerir el uso de credenciales de usuario y contraseña de cada operador cada vez que se inicie una nueva sesión de operación sobre las casas de cultivo, y permitir la opción de cerrar la sesión anteriormente abierta. Se debe guardar un registro constante de las sesiones de trabajo de cada operador. De esta forma y por cuestiones de seguridad, la dirección de la empresa agroindustrial puede llevar a cabo el control de acceso de los operarios y las acciones que estos realizan sobre el sistema.
- La aplicación de supervisión debe poder mostrar una lista de nodos producto al descubrimiento de la red de sensores inalámbricos basados en Zolertia.
- Se debe poder acceder a la medición instantánea de las variables temperatura, humedad del suelo, humedad relativa e intensidad luminosa, medidas por los nodos de la red de sensores.
- La aplicación de configuración debe permitir la realización del levantamiento fenotípico de las plantas en cualquier momento de la cosecha, el cual se deberá guardar. Según las especificaciones del cliente, este levantamiento puede ser clasificado en: de crecimiento o fisiopatológico. De esta manera, al obtener las entradas (medición de los parámetros ambientales) y salidas (levantamiento fenotípico) del sistema, se puede crear un banco de datos de vital importancia para el desarrollo de mapas mentales y modelos de cosecha basados en inteligencia artificial.

Variables de supervisión y configuración

Teniendo en cuenta el criterio de los especialistas de la empresa agroindustrial, se determinaron las siguientes las variables involucradas en el sistema a desarrollar (Ver Tabla 2.1.).

Tabla 2.1. Variables del sistema.

Variables de supervisión (medidas por la red de sensores)	**Variables de configuración (producto del levantamiento fenotípico de las plantas)**
Humedad relativa	Tamaño de la planta
Humedad del suelo	Diámetro de la base del tallo
Temperatura	Número de flores
Intensidad luminosa	Número de frutos
	Coloración de los frutos

*En ambos casos se deberá registrar la hora y fecha de la medición.

2.2 Diseño de la arquitectura de hardware

En este proyecto, se propuso una arquitectura de hardware compuesta por una Raspberry Pi 3 Modelo B como Gateway IoT, que permita el enrutamiento de los datos provenientes de los dispositivos de campo hacia la red local. Esta computadora de placa reducida fue la selección para esta propuesta de automatización debido a su bajo costo en el mercado. Además, sus características de hardware la convierten en un Gateway multicapas, capaz de interconectar dispositivos de campo y automática en general con la red empresarial. Esto es posible a través de interfaces que posee, como son para el caso de este proyecto, el puerto Ethernet para Red de Área Local (LAN) y el puerto USB (Universal Serial Bus).

La Raspberry se conectará mediante el puerto Ethernet a un Enrutador WiFi. La Raspberry Pi y el Enrutador WiFi serán situados en un panel localizado en una de las instalaciones cercanas a las casas de cultivo. A pesar de que la Raspberry tiene

la posibilidad de conectarse a través de WiFi, se propuso la conexión mediante el puerto Ethernet para conectar la Raspberry al Enrutador. Esto se debe a que presenta velocidades más rápidas, menor latencia y conexiones más confiables con respecto a la conexión WiFi. Si se utiliza para la conexión un cable de Ethernet Categoría 6 o 5e, pudieran alcanzarse velocidades en la comunicación de 10Gb/s o 1Gb/s respectivamente (Seifert, 1998). Al Enrutador WiFi podrán conectarse los dispositivos móviles y tabletas inteligentes mediante conexión WiFi que, aunque puede permitir velocidades de conexión que sobrepasan la barrera del Gb/s con el estándar 802.11ac, sigue siendo menos estable con respecto al Ethernet. La conexión WiFi permite el uso de dispositivos inteligentes de la plataforma Android para la supervisión y configuración de los parámetros ambientales dentro de las casas de cultivo, debido a que estos no poseen interfaz de red cableada Ethernet y el Enrutador permite muchas más conexiones simultáneas sin mayores costos que las que ofrece mediante el puerto Ethernet (RIMT-MAEC, 2010). Además, mediante una conexión inalámbrica con el servidor de datos, los operarios de las casas de cultivo podrán tener mayor movilidad con la interfaz de usuario del cliente (Ver Figura 2.1.).

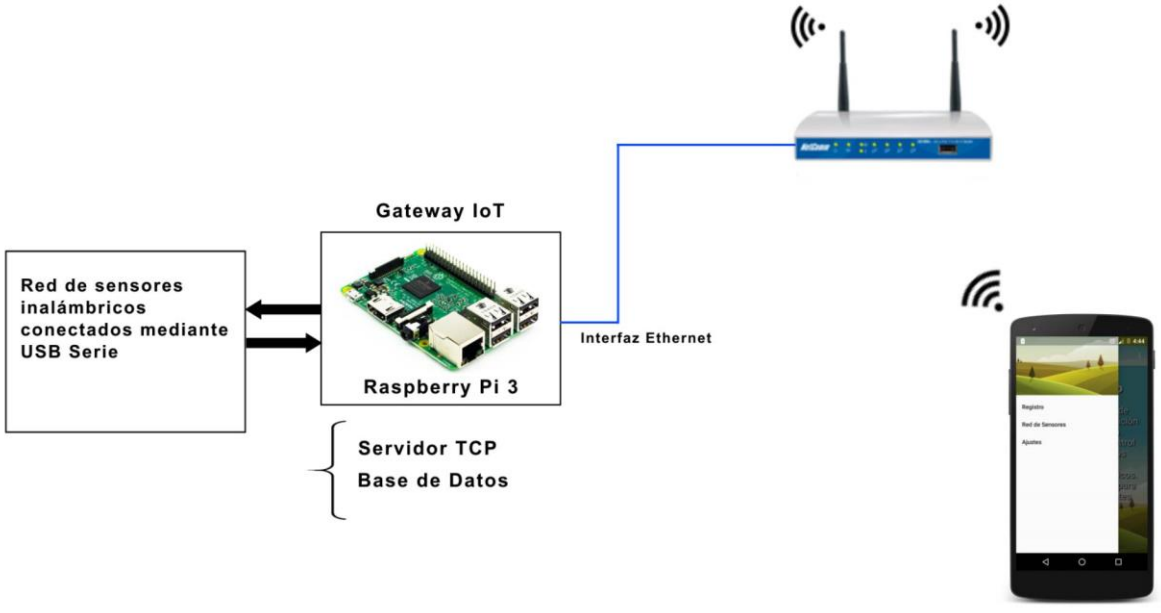


Figura 2.1. Propuesta de arquitectura de hardware.

En el caso del proyecto propuesto por la empresa CEDAI, se cuenta con una red de sensores inalámbricos basada en la plataforma hardware para IoT Zolertia Z1, que utiliza la Raspberry Pi 3 como Gateway IoT, la cual se conecta a la red mediante protocolo Serie a través del puerto USB de la Raspberry y el nodo coordinador de la red de sensores. Los nodos de la red miden parámetros como la temperatura, la humedad relativa, la humedad del suelo y la intensidad luminosa dentro de las casas de cultivos mediante sensores conectados a cada uno de los nodos Zolertia. El flujo de datos de la medición viaja hasta el nodo coordinador de la red y es almacenado en la base de datos del servidor MySQL instalado en la Raspberry. Una vez guardados, los datos de los sensores estarán disponibles para su visualización a través de dispositivos de la plataforma Android.

Al término de esta investigación, no se contaba con un módulo Ethernet para la interconexión del PLC Automation Direct desplegado en la aplicación. Esto condiciona que el desarrollo de la propuesta abordada, no permita la inclusión de la supervisión y configuración de la red de dispositivos industriales conectados a este PLC.

No obstante, las condiciones de escalabilidad de la aplicación a estos escenarios, una vez se cuente con un módulo ethernet industrial para esta familia de PLCs, están garantizadas, a partir de la inclusión de un driver Modbus TCP.

2.2.1 Raspberry Pi

El criterio de selección empleado al elegir la Raspberry como Gateway IoT, está dado por las características de esta placa reducida, las cuales la hacen idónea para este proyecto. La velocidad del procesador varía de 700 MHz a 1.4 GHz, esta última presente en la Pi 3 Modelo B +; la memoria integrada varía de 256 MB a 1 GB de RAM. Las tarjetas (SD) se utilizan para almacenar el sistema operativo y la memoria del programa en tamaños SDHC o MicroSDHC. Las placas tienen de uno a cuatro puertos USB. Para la salida de video, se admiten HDMI y video compuesto, con una toma estándar de 3,5 mm para salida de audio. La interfaz con la capa de dispositivos del modelo de referencia IoT es proporcionada por una serie de pines GPIO que admiten E/S digitales y protocolos comunes como I^2C, SPI y UART. Los

modelos B tienen un puerto Ethernet 8P8C y los Pi 3 y Pi Zero W tienen WiFi a 2,4 GHz y 5 GHz IEEE 802.11.b/g/n/ac y Bluetooth 4.2 (Maksimović et al., 2014). En la Figura 2.2. se muestra la Raspberry Pi 3 Modelo B, el modelo utilizado en la arquitectura de hardware.

La Raspberry Pi puede operarse con cualquier teclado y mouse de computadora USB genérico. También se puede usar con almacenamiento USB, conversores USB a MIDI y prácticamente cualquier otro dispositivo o componente con capacidades de conexión USB. También se pueden conectar otros periféricos a través de los pines y conectores en su superficie.

En el Anexo I, se muestran las características principales de la Raspberry Pi 3 Modelo B.

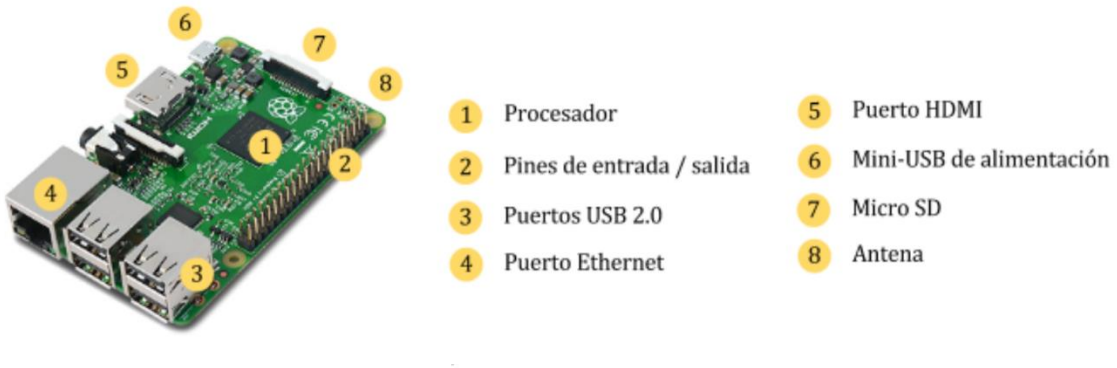


Figura 2.2. Raspberry Pi 3 Modelo B (Santamaría, 2016).

En el caso de este proyecto se utilizó para la Raspberry Pi 3, Raspbian, como sistema operativo. Aunque existen otras opciones, como Windows IoT, Pidora (derivado de Fedora) y Arch Linux ARM (derivado de Arch Linux). Raspbian en su última versión Stretch, lanzado en agosto de 2017, es un sistema operativo basado en Debian con varias versiones anteriores: Wheezy y Jessie. Desde 2015, la Fundación Raspberry Pi lo ha proporcionado oficialmente como el sistema operativo principal para la familia de computadoras Raspberry Pi de placa reducida. En este ya viene incorporado un intérprete de lenguaje Python, lenguaje principal en este sistema operativo (Zhao et al., 2015).

La arquitectura de hardware propuesta cuenta con una Raspberry Pi 3 Modelo B con Raspbian Stretch, un servidor local MySQL y un intérprete de lenguaje Python instalados.

2.2.2 Dispositivos Android

Esta interfaz de usuario se pretende diseñar en forma de aplicación para la plataforma de Android, debido a la alta difusión de dispositivos móviles de esta plataforma alrededor del mundo. Además, al utilizar Android como sistema operativo, se hace más fácil el desarrollo debido a la gran comunidad de programadores de este sistema y a la gratuidad de las herramientas para desarrollar aplicaciones(Ortega et al., 2015).

Los teléfonos inteligentes y tabletas son en su mayoría compatibles con WiFi y, por ende, capaces de manejar todo tipo de tareas de red. Debido a la naturaleza de esta propuesta de hardware para supervisión y configuración, es necesario que el dispositivo inteligente utilizado como cliente utilice el sistema operativo Android, específicamente desde la versión 4.4 (KitKat) en adelante. Debido a que, los dispositivos que tienen instaladas versiones del sistema anteriores a la versión 4.4, no poseen las mismas capacidades de red, visualización y procesamiento que los dispositivos más modernos y de versiones superiores. Al construir la aplicación cliente para la versión KitKat, se asegura que esta pueda ejecutarse en aproximadamente el 80% de los dispositivos (“Android Developers,” 2018). En la Figura 2.3. se muestran las distintas versiones del sistema operativo Android y su uso.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.7%
4.1.x	Jelly Bean	16	2.8%
4.2.x		17	4.1%
4.3		18	1.2%
4.4	KitKat	19	17.1%
5.0	Lollipop	21	7.8%
5.1		22	22.3%
6.0	Marshmallow	23	31.8%
7.0	Nougat	24	10.6%
7.1		25	0.9%

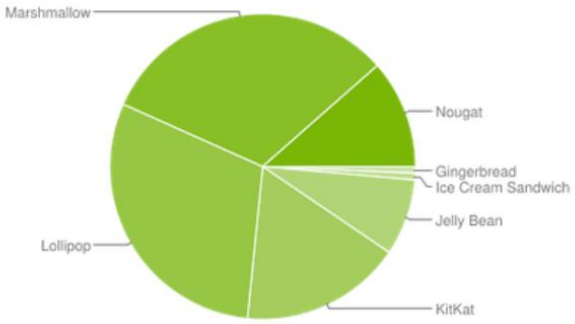


Figura 2.3. Versiones del sistema operativo Android y su distribución. Datos tomados en julio de 2017 (“Android Threat Profile,” 2017).

2.3 Diseño de la arquitectura de software

La arquitectura de software consiste en una aplicación para la plataforma Android instalada en el dispositivo móvil inteligente. Dicha aplicación hará uso de las capacidades de conexión vía WiFi del dispositivo para conectarse al Punto de Acceso (AP) configurado en el Enrutador WiFi. A través de la conexión de red establecida, la aplicación se conectará al servidor de datos ejecutándose sobre la Raspberry. Este procesará las peticiones realizadas por la aplicación Android y hará uso de un servidor de base de datos MySQL local. La base de datos almacena los datos recibidos desde los dispositivos de la red de sensores, los levantamientos fenotípicos de las cosechas y los datos de registro de usuario.

2.3.1 Funcionamiento de la aplicación

La aplicación desarrollada permite la supervisión de los parámetros ambientales (temperatura, humedad relativa, humedad del suelo e intensidad luminosa),

medidos por los nodos de la red de sensores inalámbricos y almacenados en la base de datos de la Raspberry. Además, posibilita el levantamiento fenotípico de las plantas en cualquier etapa de su crecimiento. Esta arquitectura de software soporta la conexión simultánea de varios usuarios, aunque no es imprescindible. Esto se debe a que el proyecto de CEDAI solo prevé la compra de una tableta con sistema operativo Android para el uso de los operarios en la UEB de Cultivos Protegidos “Valle del Yabú”.

Para el desarrollo del software fue necesario la determinación del número de actores de la aplicación:

- El primero es el usuario administrador, identificado por sus credenciales con permisos administrativos.
- El segundo actor es el usuario u operario estándar de las casas de cultivo, identificado por sus credenciales de usuario con permiso estándar.

El usuario estándar, podrá configurar la conexión, ingresar su usuario y contraseña personales. Podrá descubrir los nodos de la red de sensores inalámbricos, acceder a las mediciones instantáneas de cada nodo y realizar el levantamiento fenotípico de las plantas. Mientras que, el usuario administrador podrá realizar todas las operaciones de un usuario estándar, además de la posibilidad de registrar nuevos usuarios estándares para operar el sistema.

Diagrama UML de casos de uso

El lenguaje de modelado unificado (UML) es un lenguaje de modelado de desarrollo, de propósito general en el campo de la ingeniería de software, que pretende proporcionar una forma estándar de visualizar el diseño de un sistema y comprender la forma en que deberá comportarse. Ayuda a obtener los requerimientos desde el punto de vista del usuario (Schmuller, 2000). En la Figura 2.4. se muestra el diagrama UML de casos de uso de la aplicación Android.

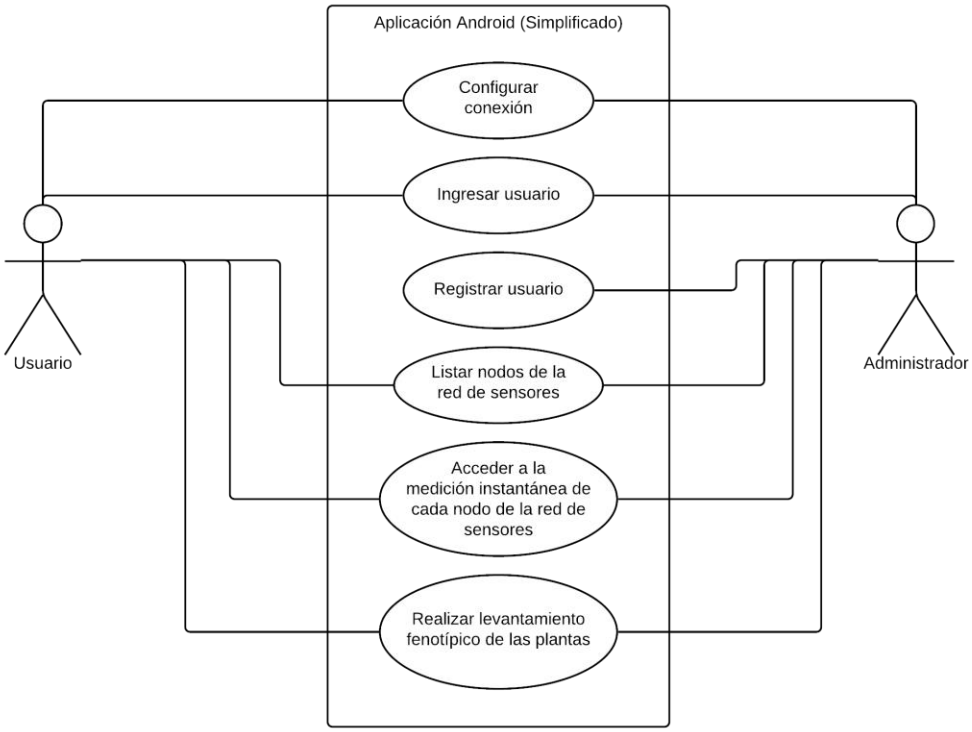


Figura 2.4. Diagrama de casos de uso de la aplicación Android.

El sistema comienza su funcionamiento una vez que se inicia la aplicación del servidor en la Raspberry. Cuando un usuario inicia la aplicación Android, si la configuración del servidor (dirección IP y puerto) es válida, la aplicación cliente se conecta automáticamente al servidor, si no, el usuario deberá introducir una configuración válida. Dicha configuración se guardará de forma tal que, al reiniciar la aplicación, no será necesario reconfigurar y en ese caso la conexión se realizaría automáticamente. Una vez establecida la conexión, será necesario ingresar el nombre de usuario y contraseña del operador que inicia sesión. Una vez que se inicia una sesión para dicho usuario, se desbloquean las utilidades correspondientes a su nivel de usuario. El usuario podrá desplazarse hacia el interior de las casas de cultivos a realizar el levantamiento fenotípico de las plantas, lo cual puede provocar una desconexión con el servidor de datos. Una vez realizado el levantamiento de los parámetros requeridos por la aplicación, el operario deberá acercarse a una distancia menor que el radio de alcance de la señal del Enrutador

para reconectarse al servidor y subir los datos recopilados manualmente hacia la base de datos del servidor MySQL. Desde esta posición, se podrá acceder al listado de nodos registrados en la base de datos por el nodo coordinador de la red. Una vez que se realiza un click sobre uno de los nodos, se accederá a la supervisión de los parámetros ambientales instantáneos medidos por este.

2.3.2 Servidor

La aplicación del servidor ejecutándose sobre la Raspberry, es la encargada de la gestión de la conexión de los dispositivos Android, la conexión con el servidor local MySQL, el manejo de la base de datos de los dispositivos de campo, la administración de credenciales de usuarios y el procesamiento de los comandos recibidos desde los dispositivos que funcionan como clientes. Además, desde su aplicación en consola se puede acceder a un registro de accesos de usuarios y las acciones realizadas por estos.

En aras de la simplicidad y el alcance de este trabajo, la propuesta de servidor se ejecuta sobre la capa de transporte del modelo OSI, por tanto, el protocolo de comunicación empleado es TCP.

Seguridad

En aplicaciones y servicios de red relacionados con el comercio electrónico y el intercambio de datos privados, se utilizan los protocolos TLS/SSL. Para este proyecto se decidió el empleo del método de encriptación RSA para la conexión mediante el protocolo TLS/SSL.

En el caso de este proyecto, debido al uso del protocolo TCP para la comunicación cliente – servidor, TLS/SSL pasará a realizar un *wrapper* (envoltorio) de una conexión previamente establecida a través de *sockets*. Un socket de red es un punto final virtual donde las entidades pueden realizar la comunicación entre procesos. Por ejemplo, un proceso ejecutándose en una computadora intercambia datos con otro proceso que se encuentra en la misma u otra computadora. Normalmente se etiqueta el primer proceso que inicia la comunicación como el cliente y el último como el servidor.

Python

El servidor fue programado en lenguaje Python, debido a que Raspbian Stretch trae instalado nativamente el intérprete versión 3.0, necesario para ejecutar cualquier aplicación de Python. Además, es orientado a objetos y tiene licencia de código abierto, o sea, permite vender, usar o distribuir cualquier aplicación basada en este lenguaje, sin necesidad de un permiso adicional. El hecho de que Python se ejecuta en tantas plataformas implica que no es necesario escribir una aplicación con portabilidad limitada. Permite el desarrollo de código y experimentación en vivo, eliminando así el consumo de tiempo y compilación. Puede ser integrado en otras aplicaciones escritas en otros lenguajes (Oliphant, 2007).

Programación

El servidor consta de tres archivos de extensión **.py* programados en lenguaje Python utilizando el IDE Spyder ejecutándose en Ubuntu. Un archivo contiene la clase *Server*, la cual implementa todas las funciones encargadas de gestionar la conexión del servidor con los clientes a través de la red. Otro contiene la clase *DBManager*, la encargada de administrar la conexión al servidor local MySQL. Y el tercero se encarga de la creación de un objeto de tipo *Server* y constituye la aplicación necesaria a ejecutar para iniciar los servicios del servidor de datos.

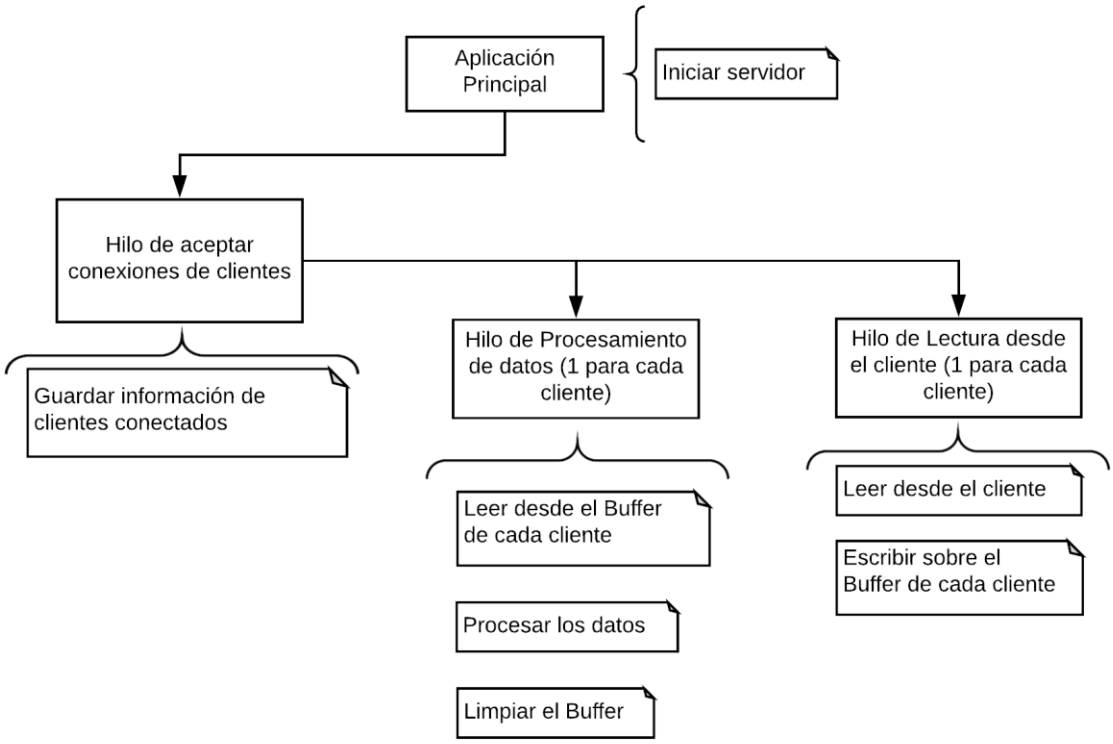


Figura 2.5. Hilos de ejecución del servidor y sus funciones.

En la Figura 2.5. se muestran los hilos de ejecución del servidor y las funciones que realiza. Una vez que la aplicación principal es ejecutada desde un Terminal de Raspbian, se inicia el servidor con la creación de un socket de la biblioteca de Python del mismo nombre. El socket se enlaza a un puerto determinado (una interfaz para comunicarse con un programa a través de una red) de los 65536 disponibles. Luego inicia un hilo de ejecución paralelo encargado de aceptar nuevas conexiones de clientes. El número de conexiones simultáneas está determinado por el parámetro que se pasa a la función *listen(<entero>)* perteneciente a la clase *sockets* de Python.

Cada vez que se acepta una nueva conexión, se inician dos hilos paralelos para cada cliente:

- El primero es el encargado de leer constantemente del socket en espera de mensajes enviados desde el dispositivo cliente que se conectó al servidor. También se encarga de guardar los datos recibidos en un *buffer*.
- El otro hilo procesa los datos almacenados en el *buffer* constantemente y realiza acciones en relación a los comandos y argumentos de comando enviados desde cada cliente.

Cada petición de muestra de un dato almacenado en el servidor MySQL, como la medición de la red de sensores, o la solicitud de guardar en la base de datos, un dato introducido desde la interfaz de la aplicación Android, es procesada mediante al uso de un objeto *DBManager*, el cual implementa la función *do_over_db(<sentencia SQL>)*.

Cada vez que el usuario administrador introduce un nuevo usuario en el sistema, los datos relacionados con las credenciales del mismo son guardadas en la base de datos local de la Raspberry. Por cuestiones de seguridad, las contraseñas de usuario se codifican antes de ser guardadas mediante el método de encriptación SHA-2. Cada vez que se realiza el acceso de un usuario con sus credenciales, el servidor encripta la contraseña enviada por el cliente y la compara con el valor encriptado que se encuentra almacenado en la base de datos.

2.3.3 Cliente

La aplicación Android fue desarrollada utilizando el IDE Android Studio (IDE oficial ofrecido por Google y de uso libre y gratuito), utilizando los SDKs correspondientes a todas las versiones del sistema operativo Android partiendo de la versión 4.4 (KitKat), garantizando que la aplicación desarrollada funcione en la mayor cantidad de dispositivos Android como sea posible, sin comprometer su rendimiento y capacidad de visualización al utilizarse como portador de la aplicación.

Programación

Android Studio emplea el lenguaje anidado XML para la programación de las vistas o interfaces visuales (frontend) de la aplicación, y el lenguaje orientado a objetos Java para la programación de la lógica interna de la aplicación (backend).

La aplicación consta de una única *Activity* (Actividad), cuya interfaz visual solamente sirve de contenedor para los distintos *Fragment* (Fragmentos) que constituyen las diferentes interfaces con las que el usuario interactúa. La Actividad principal contiene un *Navigation Drawer* (Panel lateral), en el cual se encuentra un menú mediante el cual se puede acceder a los diferentes Fragmentos (Registro, Red de Sensores, Levantamiento Fenotípico, Ajustes). Existe un Fragmento de tránsito llamado *MainPageFragment* (Página de inicio), la pantalla de bienvenida de la aplicación e informa al usuario del estado de la conexión con el servidor de datos. La comunicación entre Fragmentos y la Actividad principal se efectúa mediante una *Interface* (Interfaz), la cual es implementada en la Actividad principal establece el enlace entre esta y los diferentes Fragmentos.

La conexión con el servidor debe perdurar durante toda la existencia de la aplicación y hasta que esta se destruya, y debe poder estar accesible tanto desde los diferentes Fragmentos como desde la misma Actividad principal. Por tanto, la clase *TcpClient* se creó en la forma de *Singleton*, lo cual representa un tipo de objeto que, al ser creado una primera vez, puede ser accedido desde cualquier parte de la aplicación utilizando una de sus funciones públicas, implementada para devolver una instancia del mismo objeto ya creado.

A lo largo del desarrollo de la aplicación, fue necesario la utilización de operaciones necesarias a ejecutar en hilos paralelos al hilo principal de la interfaz de usuario (UI), y al mismo tiempo el uso de tareas asíncronas (*AsynkTask*), las cuales constituyen una especie de paralelismo en la ejecución, ya que evitan que la UI se bloquee en espera de la ejecución de una tarea de larga duración, y al finalizar su ejecución pueden llamar al método *onPostExecute()*, el cual puede modificar el UI con los resultados de la tarea.

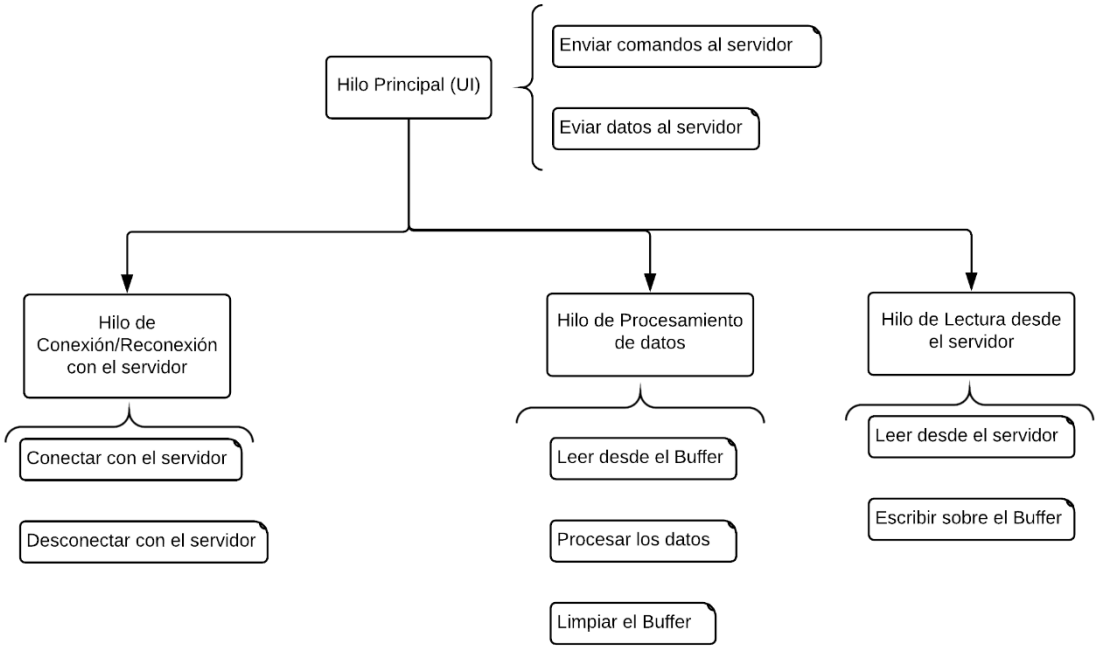


Figura 2.6. Hilos de ejecución fundamentales del cliente y sus funciones.

En la Figura 2.6. se muestran los principales hilos de ejecución paralelos al UI. Al iniciarse la aplicación, primeramente, se crea la Actividad principal y se crean tres hilos paralelos al hilo de ejecución del UI: el primero es el encargado de comprobar si hay o no conexión con el servidor de datos, y en caso de no estar conectado intenta conectarse; el segundo hilo es el encargado de leer los datos recibidos desde el servidor de datos y de guardarlos en el *buffer*; y el tercer hilo, se encarga del procesamiento de los datos almacenados en el *buffer*. Luego se crea el Fragmento Página inicial, la cual actualiza el estado de la conexión cada cierto período de tiempo. De esta forma comienza la aplicación su operación. Durante su uso, cada acción realizada por el usuario puede desencadenar la ejecución de hilos paralelos, tareas asíncronas, temporizadores, uso del *Singleton*, ejecución de funciones y funciones de la Interfaz, así como cambios en el UI.

2.4 Conclusiones del capítulo

La arquitectura propuesta consta de una Raspberry Pi 3 Modelo B funcionando como Gateway IoT, conectada a un Enrutador WiFi a través del puerto Ethernet. Además, se cuenta con una tableta con sistema operativo Android de versión 4.4 (KitKat) en adelante, en la cual se ejecuta la aplicación que funciona como cliente e interfaz de usuario para el operador en la conexión con el servidor de datos.

En la Raspberry se ejecutará el servidor de datos programado en lenguaje Python, que gestionará el flujo de datos de entrada y salida del servidor local MySQL, el envío, recepción, procesamiento de comandos y la seguridad en la conexión con los clientes Android. La aplicación móvil permitirá el muestreo de las variables medidas por la red de sensores instalada en las casas de cultivo y permitirá la realización del levantamiento fenotípico de las plantas. Una vez propuesta y desarrollada la arquitectura de hardware y software de acuerdo a las especificaciones brindadas por la empresa agroindustrial, se hace necesario la realización de pruebas de funcionamiento y estabilidad del sistema.

CAPÍTULO 3. PRUEBAS Y RESULTADOS

Los resultados arrojados por una investigación constituyen elementos de gran importancia, tanto para el objeto de estudio como para futuras investigaciones del mismo campo científico. En este capítulo se abordan diferentes pruebas realizadas a la aplicación Android y el servidor, enfocadas en el funcionamiento, la conectividad, la detección de errores, el rendimiento y la seguridad en la comunicación. También se realiza un análisis económico y de posibles impactos de la aplicación desarrollada.

3.1 Pruebas realizadas

3.1.1 Prueba manual de conectividad del servidor

Desde el comienzo del desarrollo de la arquitectura de software de este proyecto y luego de finalizada su versión inicial, se realizaron pruebas de funcionamiento de cada una de sus características, así como la evaluación de sus errores con el objetivo de corregirlos.

La prueba manual de los usuarios de un software desarrollado es la evidencia final de su correcto funcionamiento. Este tipo de prueba es de larga duración, debido a que se necesita que los usuarios exploten todas las capacidades de la aplicación en todas las condiciones de operación posibles. Aunque debido al alcance de tiempo de este proyecto, es necesario la realización de una prueba más rigurosa y de corta duración.

El siguiente experimento se realizó con el fin de probar los casos de uso del sistema desarrollado, detectar errores de funcionamiento, facilitar la realización de pruebas

paralelas de rendimiento del servidor y determinar un número máximo de conexiones simultáneas de dispositivos clientes al servidor.

Condiciones necesarias y garantizadas para la realización de la prueba:

- Los dispositivos móviles poseen conectividad disponible a través de WiFi.
- Tienen habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.
- Poseen pantalla táctil funcionando correctamente.
- Tienen instalada la aplicación cliente de Android (GHMonitor) respectivamente.
- Poseen pantalla táctil funcionando correctamente.
- La aplicación GHMonitor tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- El servidor se ejecuta sobre la Raspberry Pi 3 Modelo B permitiendo un máximo de 4 clientes conectados simultáneamente.
- En el servidor se encuentran registrados al menos 7 usuarios y sus credenciales. Las cuentas correspondientes a estos usuarios se encuentran habilitadas.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

Para la realización de la prueba se emplearon 7 teléfonos inteligentes con sistema operativo Android y sin ningún criterio específico de selección. Se conectó un dispositivo a la vez. Después de conectados todos los dispositivos, se intentó el registro simultáneo de todos los usuarios con sus credenciales y de peticiones a la base de datos MySQL de la Raspberry. Finalmente se inició un proceso de peticiones y desconexiones aleatorias. La prueba alcanzó una duración de 30 minutos.

Resultados

La prueba arrojó los siguientes resultados:

- Fueron probados manualmente todos los casos de uso de la aplicación GHMonitor. No se detectó ningún error o excepción no manejada en el funcionamiento de la aplicación.
- Mientras el número de dispositivos conectados crece, también aumenta probabilidad de que, al realizar una petición simultánea al servidor, la aplicación agote su tiempo en espera de respuesta en alguno de los dispositivos. Esto se debe al uso del ancho de banda de la conexión mediante la cual se comunican servidor y clientes. Por tanto, es importante la correcta selección del Enrutador WiFi a emplear en el montaje del proyecto.
- Debido a la carga de CPU provocada en el servidor por las peticiones de conexión, desconexión y procesamiento de datos al servidor, la realización de esta prueba permite la realización de pruebas paralelas de rendimiento del servidor.
- Esta prueba aporta un número probable de dispositivos conectados simultáneamente a ser configurado en la función *listen(<# de dispositivos>)* del servidor. Un número mayor a 7 dispositivos conectados al mismo tiempo, podría elevar demasiado el consumo del procesador de la Raspberry Pi 3, reduciendo su vida útil y comprometiendo la funcionalidad del sistema de supervisión y configuración desarrollado.

3.1.2 Prueba automatizada de conectividad del servidor

Una manera fiable de comprobar el funcionamiento de un software desarrollado y detectar errores en su programación, es su uso por parte de muchos usuarios, los cuales expresan las dificultades presentadas por esta. Pero usualmente este tipo de prueba no es posible debido al tiempo necesario para su realización y la cantidad de personas que involucra. En este contexto, es más fiable el empleo de pruebas automatizadas, las cuales en menor tiempo pueden detectar errores en el código.

Para la realización de esta prueba, fue desarrollada una aplicación Android adicional (GHMTester). Dicha aplicación consta de una única pantalla de interacción con el usuario que permite la introducción de los parámetros de la prueba a realizar, los cuales son: Dirección IP y Puerto del servidor, y el tiempo de duración de la prueba.

Al iniciarse la prueba (presionando el botón *START TEST*), la aplicación comienza un proceso continuo de conexión, desconexión y envío de mensajes, donde la acción a realizar se decide de forma aleatoria (Ver Figura 3.1.).

Esta aplicación se diferencia de la aplicación cliente (GHMonitor) solamente en la parte visual y de interacción con el usuario, ya que en *backend* (código fuente) utiliza la misma clase *TcpClient* y los mismos servicios que utiliza la aplicación cliente GHMonitor, por lo que el servidor interpreta la conexión de ambas aplicaciones de la misma forma.

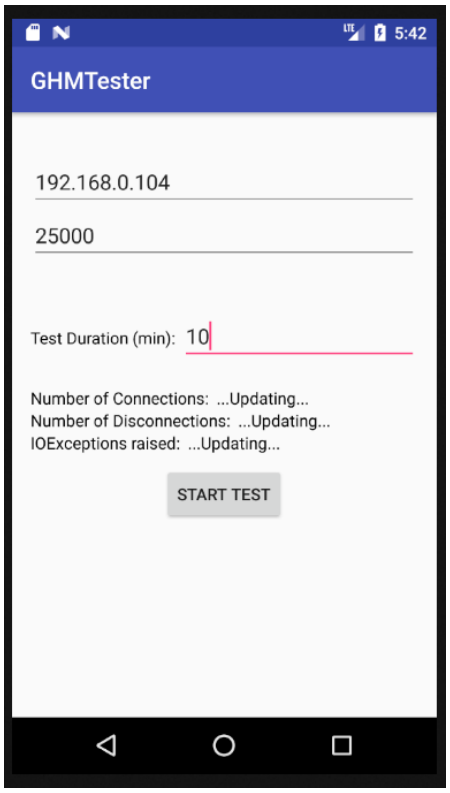


Figura 3.1. Pantalla de la aplicación de prueba automatizada GHMTester.

Condiciones necesarias y garantizadas para la realización de la prueba:

- Los dispositivos poseen conectividad disponible a través de WiFi.
- Tienen habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.

- Poseen pantalla táctil funcionando correctamente.
- Tienen instalada la aplicación prueba automatizada de Android (GHMTester) respectivamente.
- La aplicación GHMTester tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- El servidor se ejecuta sobre la Raspberry Pi 3 Modelo B permitiendo un máximo de 4 clientes conectados simultáneamente.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

El experimento se realizó mediante la configuración de una prueba de 1 hora de duración en cada uno de los 3 dispositivos móviles seleccionados para ejecutar la aplicación GHMTester. Luego de la conexión de los dispositivos a la misma red local a la que se encuentra conectada la Raspberry, se inician las pruebas.

Resultados

La prueba arrojó los siguientes resultados:

- Permitió la detección y posterior corrección de un error en la gestión de los hilos de ejecución del servidor.
- No fue lanzada ninguna excepción en la aplicación en los experimentos realizados. No se generan excepciones siempre que el dispositivo cliente sea enrutado hacia el servidor y le sean otorgados a la aplicación, los permisos necesarios dentro del sistema operativo Android.
- Si se realizan varias pruebas simultáneamente, el número de conexiones logradas por la aplicación de prueba es menor en cada dispositivo, que la cantidad lograda cuando se realiza una prueba en un solo dispositivo. Esto evidencia el uso del ancho de banda en la conexión multiusuarios, por lo que es necesario prestar atención a la selección del Enrutador WiFi a emplear en la arquitectura de hardware.
- Previó la posibilidad de que la aplicación GHMTester pudiera ser utilizada en pruebas posteriores de los servicios de red de la aplicación cliente en las diferentes versiones de Android para las que fue destinadas.

- Debido a la continuidad y simultaneidad de las peticiones de conexión, desconexión y procesamiento de datos al servidor, el uso de esta aplicación (GHMTester) se prevé para la realización de pruebas de rendimiento del servidor.

3.1.3 Pruebas de rendimiento del servidor

Paralelo a la realización de pruebas de conectividad del servidor, se efectuó un análisis del rendimiento del software del servidor basado en la carga generada sobre la CPU del Gateway IoT. Este análisis se efectúa con el propósito de determinar el número de conexiones simultáneas de clientes al servidor permitidas con el fin de proteger la funcionalidad del sistema y proteger el hardware utilizado ante sobrecalentamientos y sobreconsumo.

Para medir el uso de CPU de la Raspberry se empleó el monitor de rendimiento presente en el administrador de tareas del sistema operativo Raspbian. La prueba de rendimiento se efectuó durante la realización de la prueba de conectividad del servidor.

Condiciones necesarias y garantizadas para la realización de la prueba:

- La prueba se realiza simultáneamente a las pruebas anteriores relacionadas con la conectividad del servidor.
- Los dispositivos móviles poseen conectividad disponible a través de WiFi.
- Tienen habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.
- Tienen instalada la aplicación cliente de Android (GHMonitor) respectivamente.
- Poseen pantalla táctil funcionando correctamente.
- La aplicación GHMonitor tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

Resultados

La prueba arrojó los siguientes resultados:

- La carga de CPU generada por el servidor en el Gateway, en la prueba manual de conectividad, alcanza el 40% durante la conexión simultánea de los 7 dispositivos involucrados (Ver Figura 3.2.). Aunque, el uso de CPU pudiese ascender a más del 90%, se requiere que no sea constante. El valor que se desea es de un promedio menor al 50%.

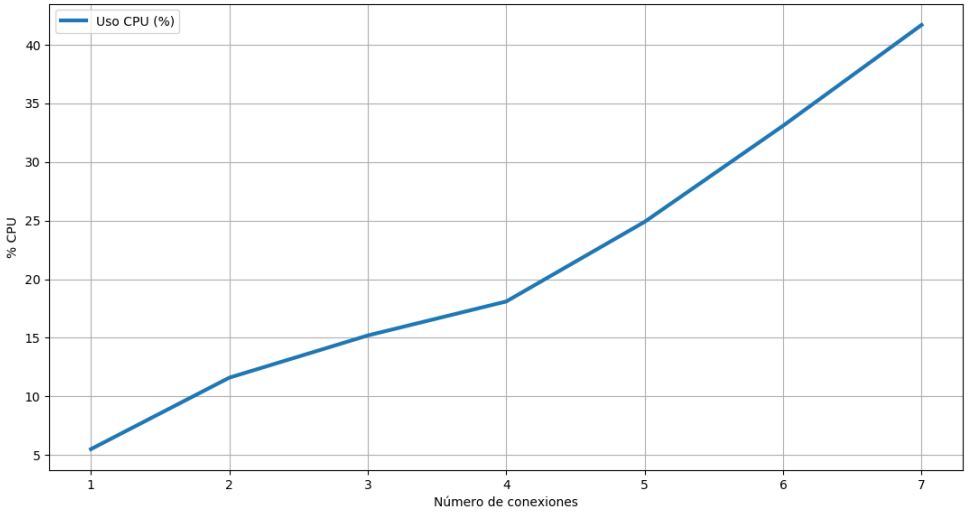


Figura 3.2. Uso de CPU en la Raspberry Pi 3 Modelo B ante la conexión de usuarios de GHMonitor.

- Durante la prueba automatizada de conectividad, ante una sobrecarga de operaciones de conexión, desconexión, lectura y procesamiento de datos, se observa que el uso de CPU ante 3 dispositivos empleados, alcanza el 30% (Ver Figura 3.3.).

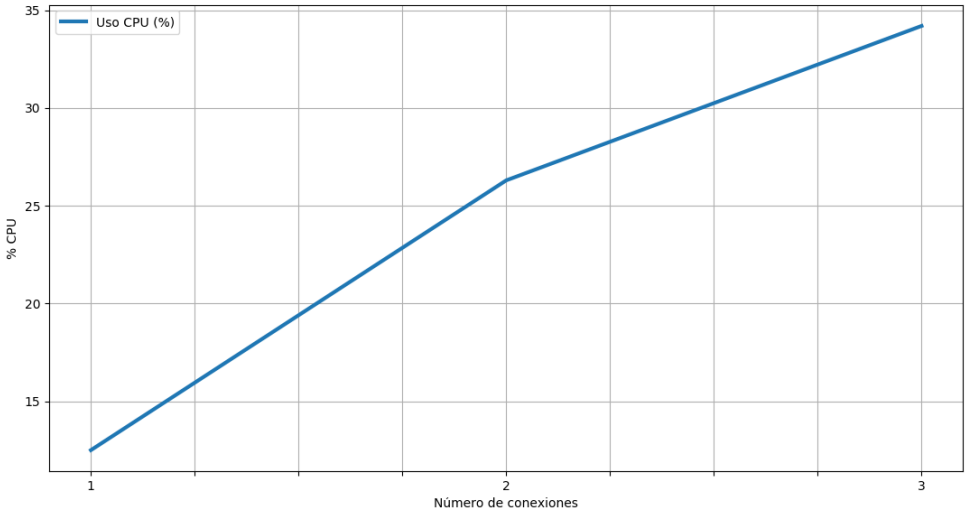


Figura 3.3. Uso de CPU en la Raspberry Pi 3 Modelo B ante la conexión de usuarios de GHMTester.

- Si se consideran condiciones normales a las descritas en las especificaciones brindadas por la empresa agroindustrial, se determina que la Figura 3.2. muestra tales condiciones con bastante similitud. Por tanto, se define el número máximo de usuarios conectados simultáneamente al servidor como 6. Aunque en la práctica, la cantidad de operadores del sistema en cada turno de trabajo es menor. Además, este proyecto incluye solo un dispositivo inteligente (Tablet) en el cual ejecutar la aplicación cliente GHMonitor.

3.1.4 Prueba de versiones del sistema operativo Android

Un elemento de vital importancia que caracteriza a una aplicación es su portabilidad. En el caso de la aplicación servidor, debido a que está programada en lenguaje Python, es capaz de ejecutarse y funcionar en cualquier computadora con capacidades de red y cualquier sistema operativo que tenga instalado el intérprete de Python. La aplicación cliente, por su parte, se ejecuta sobre dispositivos inteligentes de la plataforma Android desde su versión 4.4 hasta la 7.0, permitiendo que la aplicación funcione en la mayor cantidad de dispositivos posible.

La siguiente prueba se realiza para comprobar el correcto funcionamiento de la aplicación cliente en dispositivos móviles de diferentes versiones del sistema operativo Android (partiendo de la versión 4.4 hasta la versión 7.0).

Las características de los dispositivos móviles involucrados en la prueba se muestran en la Tabla 3.1.

Tabla 3.1. Dispositivos móviles empleados en la prueba de versiones de Android.

Modelo	**Procesador**	**RAM**	**Pantalla**	**Versión de SO**
Samsung Galaxy S4	Qualcomm Snapdragon 600 Quad Core 1,6 GHz	2 GB	4,99 pulgadas	4.4
Motorola Moto E (2nd gen) LTE	Qualcomm Snapdragon 200 Quad Core 1,2 GHz	1 GB	4,5 pulgadas	5.0
LG K8	MediaTek MT6735 Quad Core 1,3 GHz	1 GB	5 pulgadas	6.0
HTC Bolt 2	Qualcomm Snapdragon 810 Octacore 2 GHz	3 GB	5,5 pulgadas	7.0

Condiciones necesarias y garantizadas para la realización de la prueba:

- Los dispositivos poseen conectividad disponible a través de WiFi.

- Tienen habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.
- Tienen instalada la aplicación cliente de Android (GHMonitor) respectivamente.
- Poseen pantalla táctil funcionando correctamente.
- La aplicación GHMonitor tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- El servidor se ejecuta sobre la Raspberry Pi 3 Modelo B permitiendo un máximo de 4 clientes conectados simultáneamente.
- En el servidor se encuentran registrados al menos 4 usuarios y sus credenciales. Las cuentas correspondientes a estos usuarios se encuentran habilitadas.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

El experimento se realizó probando manualmente los casos de uso de la aplicación Android en cada uno de los dispositivos móviles involucrados. La prueba se ejecutó durante todo el período de desarrollo de la aplicación Android y después de terminada. Adicionalmente se probó la aplicación de prueba automática desarrollada (GHMTester) para comprobar la conectividad de los servicios de red utilizados por ambas aplicaciones en un período de tiempo prolongado de 8 horas y de forma automatizada.

Resultados

La prueba arrojó los siguientes resultados:

- La aplicación funcionó correctamente en todos los dispositivos probados, lo cual corrobora los ajustes realizados en el proyecto de Android Studio. En la versión 7.0, debido a que la API del sistema incorpora muchos cambios desde la versión 4.4 de Android, la funcionalidad de envío de los comandos de Acceso de usuario no funcionó correctamente. El error detectado fue corregido mediante el uso de Tareas Asíncronas en la pantalla de Registro.

- A medida que la versión de Android disminuye, aumenta la tendencia al tamaño de pantalla reducida y una menor resolución de pantalla, lo cual dificulta la visualización dentro de la aplicación.
- Se selecciona la versión 6.0 de Android para el Tablet del proyecto de CEDAI ya que es la versión más utilizada, con un 31.8% de los dispositivos según las estadísticas (“Android Threat Profile,” 2017).
- La prueba automatizada mediante la aplicación GHMTester comprobó la conectividad de servicios de red utilizados por ambas aplicaciones en cada uno de los dispositivos. Se demuestra la utilidad de la aplicación de prueba para la evaluación de la conectividad. La funcionalidad de esta aplicación pudiera ser agregada a la aplicación GHMonitor para la prueba del servidor y la conectividad al mismo una vez introducida la configuración de conexión.
- Las pruebas realizadas en cada dispositivo involucrado permiten la realización de pruebas paralelas de rendimiento en los mismos.

3.1.5 Pruebas de rendimiento de la aplicación cliente de Android

Además de las pruebas realizadas en dispositivos con diferentes versiones de Android, es necesario realizar un análisis del rendimiento de la aplicación desarrollada. Este se efectúa con el propósito de conocer los requerimientos mínimos que debe tener un dispositivo móvil para que sea capaz de utilizar la aplicación.

Existen muchas aplicaciones con las cuales se puede conocer el rendimiento de un dispositivo móvil, entre ellas, se encuentra la aplicación CPU-Z, la cual nos brinda un informe con las características u uso de CPU, información de temperatura, batería, RAM y los diferentes sensores integrados (“CPUID,” 2018).

Otra aplicación es App Tune-up Kit, desarrollada por Qualcomm Innovation Center, Inc., fabricante de procesadores de la plataforma móvil. Esta nos permite la selección de otra aplicación instalada en el dispositivo para la confección de su perfil de uso de recursos. Tiene la limitante de que solo es compatible con dispositivos que utilicen CPU de Qualcomm (“Wireless Technology & Innovation | Mobile Technology,” 2017).

La prueba de rendimiento se efectuó utilizando la aplicación CPU-Z en los dispositivos móviles en los cuales se empleó la aplicación cliente GHMonitor durante la realización de la prueba de versiones de Android. Además, se utilizó la aplicación App Tune-up Kit en los dispositivos con CPU Qualcomm, corroborando la lectura realizada con la aplicación CPU-Z y aprovechando la funcionalidad de perfilar el consumo de CPU que brinda App Tune-up Kit.

Condiciones necesarias y garantizadas para la realización de la prueba:

- La prueba se realiza simultáneamente a las pruebas anteriores de la aplicación GHMonitor.
- Los dispositivos poseen conectividad disponible a través de WiFi.
- Tienen habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.
- Tienen instalada la aplicación cliente de Android (GHMonitor) respectivamente.
- Tienen instalada la aplicación CPU-Z respectivamente.
- Si utilizan CPU de Qualcomm, tienen instalada la aplicación App Tune-up Kit.
- Poseen pantalla táctil funcionando correctamente.
- La aplicación GHMonitor tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

Resultados

La prueba arrojó los siguientes resultados:

- La carga de CPU que representa la aplicación GHMonitor para los dispositivos involucrados en la prueba es menor al 25% (Ver Figura 3.5.).

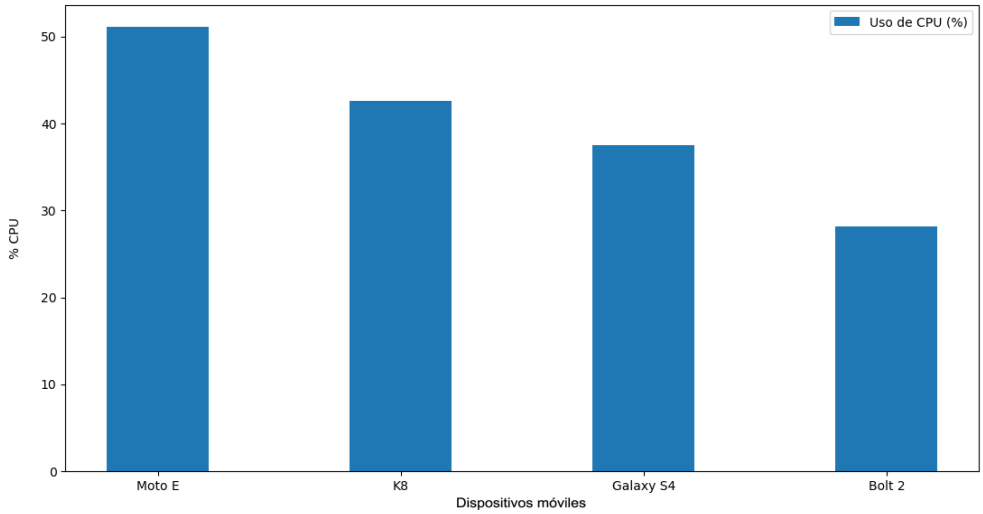


Figura 3.4. Uso total de CPU en dispositivos móviles con distintas prestaciones durante la prueba de rendimiento de la aplicación GHMonitor.

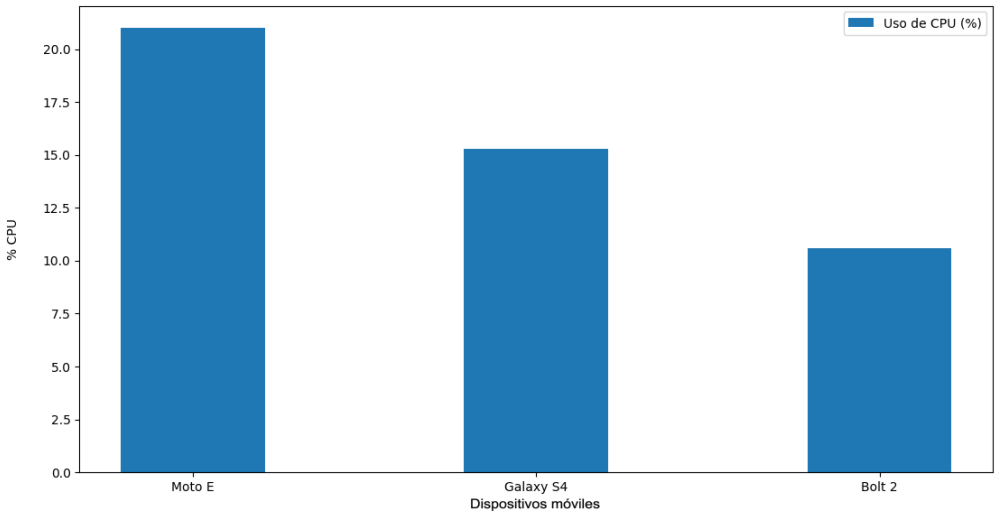


Figura 3.5. Uso de CPU de la aplicación GHMonitor en dispositivos móviles con procesador Qualcomm.

- El desempeño de la aplicación GHMonitor y el rendimiento general (Ver Figura 3.4.) en dispositivos móviles con diferentes características de

hardware y software permite la posterior determinación de los requerimientos mínimos y recomendados para la utilización de la misma (Ver Anexo III).

3.1.6 Prueba de seguridad de la comunicación servidor – cliente

Debido a la sensibilidad de los datos trasmitidos en la comunicación servidor – cliente de la aplicación desarrollada, es necesario comprobar la seguridad en la conexión de red. Las credenciales de usuarios de la aplicación y datos de cosechas son datos que deben ser protegidos ante ataques del tipo “hombre en el medio”.

La prueba siguiente fue realizada utilizando la Raspberry Pi 3 Modelo B con el servidor programado en Python ejecutándose en ella y un teléfono inteligente como dispositivo cliente. Como analizador de red se utilizó la herramienta Wireshark.

Condiciones necesarias y garantizadas para la realización de la prueba:

- El dispositivo móvil posee conectividad disponible a través de WiFi.
- Tiene habilitada la opción de Fuentes desconocidas en las configuraciones de Seguridad.
- Tiene instalada la aplicación cliente de Android (GHMonitor) respectivamente.
- Posee pantalla táctil funcionando correctamente.
- La aplicación GHMonitor tiene el permiso de uso de Internet otorgado por el usuario en el sistema operativo Android.
- El servidor se ejecuta sobre la Raspberry Pi 3 Modelo B permitiendo un máximo de 4 clientes conectados simultáneamente.
- La herramienta Wireshark está instalada en la Raspberry.
- Existe un Enrutador WiFi intermedio en la conexión funcionando correctamente.

El experimento se realizó ejecutando la herramienta análisis una vez que el servidor ha sido iniciado. En la interfaz de Wireshark se configura un filtro de análisis para el puerto 5002 (puerto de escucha fijado para esta prueba en la configuración del servidor). Se inicia el análisis de los paquetes. No ocurre nada mientras no haya dispositivos conectados al servidor. Una vez que se conecta el dispositivo móvil al

servidor, comienza el registro de los paquetes trasmitidos en la comunicación (Ver Figura 3.6.).

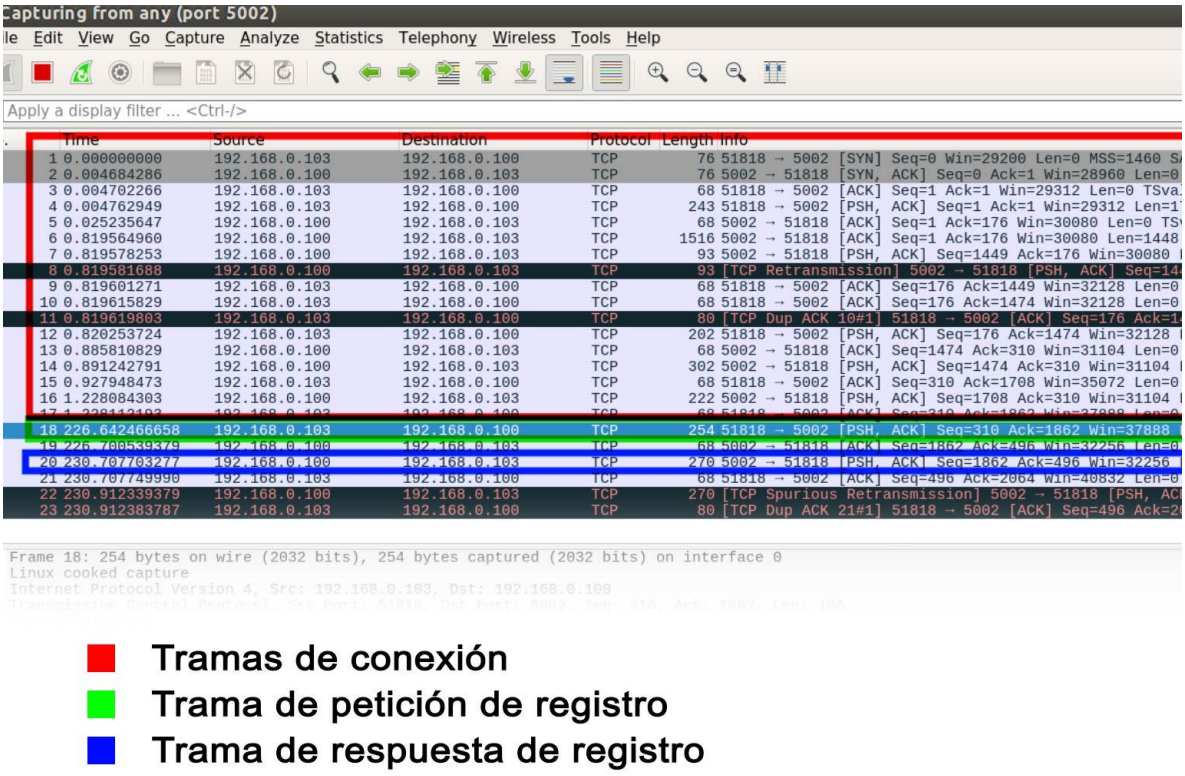


Figura 3.6. Paquetes transmitidos durante la conexión y acceso de usuario en la comunicación servidor – cliente, capturados con el analizador de red Wireshark.

Resultados

La prueba arrojó los siguientes resultados:

- La comunicación entre la aplicación cliente de Android GHMonitor y el servidor de datos ejecutándose en la Raspberry es menos vulnerable a ataques del tipo “hombre en el medio”.
- Todos los datos son ilegibles al analizar los paquetes transmitidos con la herramienta Wireshark (Ver Anexo IV).

3.1.7 Pruebas de desconexión

Teniendo en cuenta los casos de uso de la aplicación de Android, existen dos tipos de desconexión previstos: el primero, sucede cuando se cierra la aplicación de forma manual en el dispositivo móvil; y la segunda, cuando al realizar el levantamiento fenotípico de las cosechas, el operador se aleja del radio de conexión del Enrutador WiFi en busca de las casas de cultivo.

Para manejar el primer tipo de desconexión, fue necesario la espera de algún error de conexión en la operación de lectura en el servidor debido a la pérdida de la comunicación. Este error es de fácil detección, ya que al cerrar manualmente la aplicación Android, se ejecuta automáticamente la función *onDestroy()* de la misma, la cual cierra la conexión, permitiendo que el servidor detecte la ausencia del usuario.

El segundo tipo de desconexión es más complicado, ya que ocurre por una desconexión abrupta donde ninguno de los pares en la comunicación logra avisar al otro de que la conexión ha finalizado. En la aplicación Android, este problema se resuelve al igual que el primer tipo de desconexión en el servidor: mediante la espera de algún error de conexión obtenido por la operación de lectura, esta vez del cliente. Pero en el caso del servidor, el caso es más complejo. El servidor utiliza *sockets* que emplean el método de bloqueo en las operaciones de conexión, y ante una conexión abrupta no perciben la desconexión por parte de la pareja.

Para resolver este problema se agregó un hilo de ejecución paralelo en el servidor, encargado del envío periódico de una pregunta al cliente de si aún sigue conectado. Si transcurrido un tiempo predefinido, no se ha recibido ninguna respuesta del cliente, el servidor cierra el *socket* que le correspondía a este y continúa su ejecución.

Después de diversas pruebas, ambos mecanismos de detección de desconexiones funcionaron adecuadamente. El tiempo de espera ante una desconexión fue fijado en 1 minuto.

3.2 Propuesta de implementación del sistema

Con el propósito de implementar los resultados de la investigación realizada, se propusieron los pasos a llevar a cabo para el montaje del sistema diseñado en la UEB de Cultivos Protegidos “Valle del Yabú”. Los detalles de la propuesta se muestran en el Anexo II.

3.3 Análisis económico y medioambiental del sistema desarrollado

La agricultura de precisión es una tecnología que permite la combinación suelo-ecosistema-cultivos-ingeniería, para aplicar el ferti-riego a nivel de parcela y optimizar el costo y rendimiento de producción. Esto es posible mediante el empleo de las más nuevas tecnologías de las comunicaciones y el Internet de las Cosas.

La agricultura de precisión puede garantizar un uso más racional de los fertilizantes y también reducir el uso de pesticidas. Desde el punto de vista medioambiental, aplicar la cantidad correcta de insumos en el lugar correcto y en el momento adecuado, beneficia a los cultivos, los suelos y las aguas subterráneas. En consecuencia, la agricultura de precisión se ha convertido en una piedra angular de la agricultura sostenible, ya que respeta los cultivos, los suelos y los agricultores. La agricultura sostenible busca asegurar un suministro continuo de alimentos dentro de los límites ecológicos, económicos y sociales requeridos para sostener la producción en el largo plazo. Por lo tanto, la agricultura de precisión busca utilizar sistemas de alta tecnología para alcanzar este objetivo.

Con la aplicación de este trabajo en la UEB de Cultivos Protegidos “Valle del Yabú”, como parte del proyecto llevado a cabo por CEDAI, se espera un crecimiento en la producción agrícola y a la vez un crecimiento económico del país. Debido a que, en la medida en que se obtengan mejores resultados en esta empresa agroindustrial, el resultado de este proyecto será aplicado en otras empresas a lo largo del país. Desde el punto de vista de la economía agrícola, se espera la viabilidad del proyecto, lo cual significa que, junto con la adaptación del proceso, el capital invertido volverá como parte de los ingresos.

Esta empresa ha sido la de mejores resultados en todo el país. Pero no cuenta con los medios técnico necesarios para identificar las condiciones bajo las cuales ha logrado sus resultados. Por tanto, necesita un sistema de supervisión que registre los parámetros idóneos de su producción. En la Tabla 3.2. se muestran los componentes de la arquitectura de hardware utilizados en las pruebas realizadas, así como su costo en el mercado.

Tabla 3.2. Listado de componentes de la arquitectura de hardware propuesta y sus precios en el mercado.

Componentes	**Precios (USD)**	**Vendedor**
Raspberry Pi 3 Modelo B	35	Adafruit.com
Enrutador WiFi TP-Link Archer C2 AC750	49.99	Amazon.com
Tablet Lenovo (Quad Core 1.3 GHz, 2 GB RAM, Android 6.0)	126.20	Amazon.com
	211.19	

El costo total del proyecto es de 211.19 USD, lo cual representa un valor relativamente bajo con respecto a la tecnología especializada disponible en el mercado. Si se tiene en cuenta que la Raspberry Pi es un hardware libre, o sea, que cualquier aplicación que la emplee puede ser vendida a terceros, el proyecto pudiera utilizarse como un producto comercializable. Además, los softwares del servidor y del cliente fueron desarrollados para plataformas libres (Raspbian y Android respectivamente). Y la programación se desarrolló en entornos de desarrollo de uso gratuito.

Este proyecto, apoyando a proyectos paralelos de CEDAI, como redes de sensores inalámbricos e inteligencia artificial aplicada a las cosechas, pudiera tener un

impacto muy favorable sobre la producción agrícola, así como en la visión y estrategias de los trabajadores y especialistas de la agricultura en Cuba.

3.4 Conclusiones del capítulo

En este capítulo se realizaron distintas pruebas al sistema de supervisión y configuración desarrollado, enfocadas en la funcionalidad, portabilidad, los casos de uso, la conectividad, la seguridad y el rendimiento. Fueron detectados errores de la programación de los hilos de ejecución y la desconexión por pérdida de la señal WiFi en el servidor. Ambos errores fueron corregidos.

Se determinó que el número máximo de dispositivos móviles conectados permitido por el servidor debía ser 6, evitando sobrecarga de la red y alargando la vida útil del hardware. La versión de Android recomendada para el dispositivo móvil a emplear como cliente es la 6.0. La comunicación entre el servidor y el cliente posee un nivel de seguridad aceptable debido a la encriptación de los paquetes de red, lo cual la hace menos vulnerable a ataques del tipo “hombre en el medio”. Además, se desarrolló una herramienta de Android para probar del servidor.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Fue llevado a cabo el diseño de un sistema de supervisión y configuración local basado en IoT para casas de cultivo en la UEB de Cultivos Protegidos “Valle del Yabú”, de acuerdo con sus requerimientos y especificaciones. Esto fue logrado mediante la identificación de las variables a supervisar y configurar dentro del proceso productivo, la selección del Gateway IoT y de la plataforma móvil del sistema. Además, fue realizado el diseño de la arquitectura de software del servidor y la aplicación móvil. Lo cual hizo posible la propuesta de implementación del diseño para la supervisión y configuración de 3 casas de cultivo en la empresa. Esto permitió la realización de pruebas enfocadas en la valoración de su funcionalidad. Al término de estas pruebas:

1. Un sistema de supervisión y configuración local basado en IoT permite el control de acceso de los operarios, la visualización de parámetros ambientales y el levantamiento fenotípico de las cosechas en casas de cultivo.
2. La aplicación móvil diseñada como interfaz de operador, eleva las prestaciones de la interacción con el sistema y contribuye con la informatización de la agricultura.
3. El diseño de la arquitectura del Gateway IoT basado en Raspberry Pi y el empleo del sistema operativo Android como plataforma móvil demuestra la factibilidad del uso de hardware y software libres en el desarrollo de sistemas automatizados de bajo costo, como vía de alcanzar la independencia tecnológica del país.

4. El desempeño del sistema cumple con los requerimientos y especificaciones de diseño, corroborado a través de las pruebas de funcionamiento realizadas.

Recomendaciones

1. Al término de este trabajo se recomienda a la UEB de Cultivos Protegidos “Valle del Yabú”, la finalización del proyecto colaborativo entre CEDAI y la Universidad Central, mediante la implementación de la propuesta diseñada de sistema de supervisión y configuración local basado en IoT.
2. Se recomienda, además, la continuación por parte del Grupo de Internet de las Cosas, Automatización e Inteligencia Artificial, de investigaciones relacionadas con el desarrollo de aplicaciones móviles y el uso de plataformas de hardware y software libres, que puedan ser aplicadas al campo de la Ingeniería Automática.

REFERENCIAS BIBLIOGRÁFICAS

3.6.5 Documentation [WWW Document], 2017. URL https://docs.python.org/3/ (accessed 6.5.18).

Adhya, S., Saha, D., Das, A., Jana, J., Saha, H., 2016. An IoT based smart solar photovoltaic remote monitoring and control unit, in: Control, Instrumentation, Energy & Communication (CIEC), 2016 2nd International Conference On. IEEE, pp. 432–436.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., 2015. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 17, 2347–2376.

Al-Sarawi, S., Anbar, M., Alieyan, K., Alzubaidi, M., 2017. Internet of Things (IoT) communication protocols: Review, in: 2017 8th International Conference on Information Technology (ICIT). Presented at the 2017 8th International Conference on Information Technology (ICIT), pp. 685–690. https://doi.org/10.1109/ICITECH.2017.8079928

Android Developers [WWW Document], 2018. . Android Dev. URL / (accessed 5.10.18).

Android Threat Profile [WWW Document], 2017. . NJCCIC. URL https://www.cyber.nj.gov/threat-profiles/android/ (accessed 5.10.18).

Apple Developer [WWW Document], 2017. URL https://developer.apple.com/ (accessed 6.13.18).

Arduino [WWW Document], 2017. URL https://www.arduino.cc/ (accessed 6.13.18).

Atzori, L., Iera, A., Morabito, G., 2010. The internet of things: A survey. Comput. Netw. 54, 2787–2805.

Bawane, N., Gautam, P.K., Dewase, A., Mishra, V., Golait, S.S., 2017. Automation of Irrigation System using Android Technology. Int. J. Eng. Sci. 5580.

BeagleBoard.org [WWW Document], 2017. URL https://beagleboard.org/ (accessed 6.12.18).

Bulbul, H.I., Batmaz, I., Ozel, M., 2008. Wireless network security: comparison of wep (wired equivalent privacy) mechanism, wpa (wi-fi protected access) and rsn (robust security network) security protocols, in: Proceedings of the 1st

International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia and Workshop. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), p. 9.

Castro, S.A., Medina, B., Camargo, L.L., 2016. Supervisión y Control Industrial a través de Teléfonos Inteligentes usando un Computador de Placa Única Raspberry Pi. Inf. Tecnológica 27, 121–130.

Chawla, S., Kapoor, A., Sharma, S., Shukla, B., Gupta, M., Kaushik, P., Pushkar, S., 2016. App based Garden Bot for Regulation of Water Level in plants. Int. Res. J. Eng. Technol. 3, 1208–1212.

Chen, H., Jia, X., Li, H., 2011. A brief introduction to IoT gateway, in: Communication Technology and Application (ICCTA 2011), IET International Conference On. IET, pp. 610–613.

Cimino, D., Ferrero, A., Queirolo, L., Bellotti, F., Berta, R., De Gloria, A., 2016. A low-cost, open-source cyber physical system for automated, remotely controlled precision agriculture, in: International Conference on Applications in Electronics Pervading Industry, Environment and Society. Springer, pp. 191–203.

Collina, M., Bartolucci, M., Vanelli-Coralli, A., Corazza, G.E., 2014. Internet of Things application layer protocol analysis over error and delay prone links, in: Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014 7th. IEEE, pp. 398–404.

CPUID [WWW Document], 2018. URL https://www.cpuid.com/ (accessed 6.16.18).

Crow, B.P., Widjaja, I., Kim, J.G., Sakai, P.T., 1997. IEEE 802.11 wireless local area networks. IEEE Commun. Mag. 35, 116–126.

Da Xu, L., He, W., Li, S., 2014. Internet of things in industries: A survey. IEEE Trans. Ind. Inform. 10, 2233–2243.

Dheivamanogari, M., Uma, M., 2016. CONTROLLING THE TELE ROBOT USING ANDROID APP.

Dierks, T., 2008. The transport layer security (TLS) protocol version 1.2.

ElShafee, A., Hamed, K.A., 2012. Design and implementation of a WIFI based home automation system. World Acad. Sci. Eng. Technol. 68, 2177–2180.

Fernández Barcell, M., 2014. Protocolo TCP/IP.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., 1999. Hypertext transfer protocol--HTTP/1.1.

Gartner [WWW Document], 2017. URL https://www.gartner.com/newsroom/id/3859963 (accessed 6.15.18).

Goadrich, M.H., Rogers, M.P., 2011. Smart smartphone development: iOS versus Android, in: Proceedings of the 42nd ACM Technical Symposium on Computer Science Education. ACM, pp. 607–612.

Gronli, T.-M., Hansen, J., Ghinea, G., Younas, M., 2014. Mobile application platform heterogeneity: Android vs Windows Phone vs iOS vs Firefox OS, in: Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference On. IEEE, pp. 635–641.

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Gener. Comput. Syst. 29, 1645–1660.

Gupta, A., Jain, R., Joshi, R., Saxena, R., 2017. Real time remote solar monitoring system, in: Advances in Computing, Communication & Automation (ICACCA)(Fall), 2017 3rd International Conference On. IEEE, pp. 1–5.

He, N., Qian, Y., Huang, H., 2016. Experience of teaching embedded systems design with BeagleBone Black board, in: Electro Information Technology (EIT), 2016 IEEE International Conference On. IEEE, pp. 0217–0220.

Hegde, A., Gopi Kiran, T.S., Deepthi, D., Nagabhushan, T.N., Prakash, S.S., Ulle, A.R.S., 2016. Automated Water flow Control System, in: National Conference on Product Design (NCPD 2016).

Hodges, S., Taylor, S., Villar, N., Scott, J., Bial, D., Fischer, P.T., 2013. Prototyping connected devices for the internet of things. Computer 46, 26–34.

Izaguirre, E., 2012. Sistemas de Automatización [WWW Document]. URL https://es.scribd.com/document/309713744/Sistemas-de-Automatizacion-Eduardo-Izaguirre (accessed 5.10.18).

Jia, B., Yang, Y., 2011. The design of food quality supervision platform based on the Internet of Things, in: Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference On. IEEE, pp. 263–266.

Jobe, W., 2013. Native apps vs. mobile web apps. Int. J. Interact. Mob. Technol. IJIM 7, 27–32.

Kehagias, D., Nini, D., 2015. Home Automation Based on an Android and a Web Application Using Raspberry Pi.

Kim, D., Lee, G.I., Gang, D.H., Kim, Y.H., Lee, H.J., Lee, S.Y., Heo, J.W., 2012. Implementation of Greenhouse Monitoring and Control System Using a Smartphone and a Computer with Internet, in: 2012 Dallas, Texas, July 29-August 1, 2012. American Society of Agricultural and Biological Engineers, p. 1.

Lee, S.K., Bae, M., Kim, H., 2017. Future of IoT Networks: A Survey. Appl. Sci. 7, 1072.

Li, G., Zhang, W., Zhang, Y., 2014. A design of the IOT gateway for agricultural greenhouse. Sens. Transducers 172, 75.

Li, L., Maohua, W., ZHANG, M., Minzan, L.I., Sigrimis, N., Anastasiou, A., 2012. Application of IoT technology in greenhouse management, in: 2012 Dallas, Texas, July 29-August 1, 2012. American Society of Agricultural and Biological Engineers, p. 1.

Lukman, A., 2017. WIRELESS SENSOR NETWORKS BASED-INTERNET OF THING FOR AGRO-CLIMATIC PARAMETERS...

Maksimović, M., Vujović, V., Davidović, N., Milošević, V., Perišić, B., 2014. Raspberry Pi as Internet of things hardware: performances and constraints. Des. Issues 3, 8.

McBratney, A., Whelan, B., Ancev, T., Bouma, J., 2005. Future directions of precision agriculture. Precis. Agric. 6, 7–23.

Murikipudi, A., Prakash, V., Vigneswaran, T., 2015. Performance analysis of real time operating system with general purpose operating system for mobile robotic system. Indian J. Sci. Technol. 8.

Navulur, S., Sastry, A., Prasad, M.G., Prasad, G., 2017. Agricultural management through wireless sensors and internet of things. Int. J. Electr. Comput. Eng. IJECE 7, 3492–3499.

Oliphant, T.E., 2007. Python for scientific computing. Comput. Sci. Eng. 9.

Ortega, F., González Ispierto, B., Pérez Peláez, M.E., 2015. Audiencias en revolución, usos y consumos de las aplicaciones de los medios de comunicación en tabletas y teléfonos inteligentes. Rev. Lat. Comun. Soc.

Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [WWW Document], 2018. . Raspberry Pi. URL https://www.raspberrypi.org/ (accessed 5.9.18).

Rescorla, E., 2001. SSL and TLS: designing and building secure systems. Addison-Wesley Reading.

RIMT-MAEC, M.G., 2010. Comparative throughput of wifi & ethernet lans using opnet modeler. IJCST 1.

Rojas Pérez, M.I., 2015. Sistema electrónico para el monitoreo y control de cultivos utilizando tecnología inalámbrica en la Comunidad La Unión del Cantón Quero mediante software libre. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones.

Santamaría, G., 2016. Diseño de una red de monitorización del entorno basada en Arduino, Raspberry Pi y XBee.

Schmuller, J., 2000. Aprendiendo UML en 24 horas. Pearson educación.

SECTOR, S., ITU, O., 2012. SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS Next Generation Networks–Frameworks and functional architecture models. Int. Telecommun. Union Geneva Switz. Recomm. ITU-T 2060.

Seifert, R., 1998. Gigabit Ethernet: technology and applications for high-speed LANs. Addison-Wesley Longman Publishing Co., Inc.

Shekhar, Y., Dagur, E., Mishra, S., Sankaranarayanan, S., 2017. Intelligent IoT Based Automated Irrigation System. Int. J. Appl. Eng. Res. 12, 7306–7320.

Singh, E.S., Singh, R., 2015. Design of Environment Monitoring and Control System. Int. J. Eng. Comput. Sci. 4.

Singh, G., 2013. A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Int. J. Comput. Appl. 67.

Singh, M., Rajan, M.A., Shivraj, V.L., Balamuralidhar, P., 2015. Secure mqtt for internet of things (iot), in: Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference On. IEEE, pp. 746–751.

Wan, L., Sun, D., Deng, J., 2010. Application of IOT in building energy consumption supervision, in: Anti-Counterfeiting Security and Identification in Communication (ASID), 2010 International Conference On. IEEE, pp. 169–172.

Wireless Technology & Innovation | Mobile Technology [WWW Document], 2017. . Qualcomm. URL https://www.qualcomm.com/home (accessed 6.16.18).

Ye, J., Chen, B., Liu, Q., Fang, Y., 2013. A precision agriculture management system based on Internet of Things and WebGIS, in: Geoinformatics (GEOINFORMATICS), 2013 21st International Conference On. IEEE, pp. 1–5.

Yeo, K.S., Chian, M.C., Ng, T.C.W., 2014. Internet of Things: Trends, challenges and applications, in: Integrated Circuits (ISIC), 2014 14th International Symposium On. IEEE, pp. 568–571.

Zhao, C.W., Jegatheesan, J., Loon, S.C., 2015. Exploring iot application using raspberry pi. Int. J. Comput. Netw. Appl. 2, 27–34.

Zhao, J., Zhang, J., Feng, Y., Guo, J., 2010. The study and application of the IOT technology in agriculture, in: Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference On. IEEE, pp. 462–465.

Zhong, D., Lv, H., Han, J., Wei, Q., 2014. A practical application combining wireless sensor networks and internet of things: Safety management system for tower crane groups. Sensors 14, 13794–13814.

Zhou, C., Damiano, N., Whisner, B., Reyes, M., 2017. Industrial Internet of Things:(IIoT) applications in underground coal mines. Min. Eng. 69, 50.

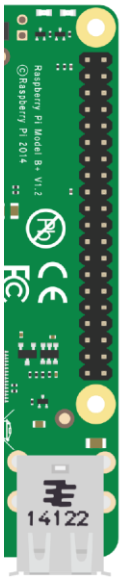
Zubizarreta Luján, D., 2017. Diseño de red de sensores inalámbricos para aplicaciones de riego agrícola. Universidad Central" Marta Abreu" de Las Villas. Facultad de Ingeniería Eléctrica. Departamento de Automática y Sistemas Computacionales.

ANEXOS

Anexo I Raspberry Pi 3 Modelo B

Características

- Broadcom BCM2837 64bit ARMv7 Quad Core con procesador de una sola placa y funcionando a 1.2 GHz.
- 1GB RAM.
- BCM43143 WiFi integrada.
- Bluetooth integrado.
- GPIO de 40 pines.
- 4 puertos USB.
- Salida estéreo de 4 polos y puerto de video compuesto.
- HDMI
- Puerto de cámara CSI para conectar la cámara de la Raspberry Pi.
- Puerto de pantalla DSI para conectar la pantalla táctil de la Raspberry Pi.
- Puerto Micro SD para cargar su sistema operativo y almacenar datos.
- Fuente de alimentación Micro USB conmutada mejorada (admite hasta 2.4 A).
- Sistema Operativo oficial: Raspbian.



Peripherals	GPIO	Particle	Pin #			Pin #	Particle	GPIO	Peripherals
3.3V			1	X	X	2	5V		
I2C	GPIO2	SDA	3	X	X	4	5V		
	GPIO3	SCL	5	X	X	6	GND		
Digital I/O	GPIO4	D0	7	X	X	8	TX	GPIO14	UART
GND			9	X	X	10	RX	GPIO15	Serial 1
Digital I/O	GPIO17	D1	11	X	X	12	D9/A0	GPIO18	PWM 1
Digital I/O	GPIO27	D2	13	X	X	14	GND		
Digital I/O	GPIO22	D3	15	X	X	16	D10/A1	GPIO23	Digital I/O
3.3V			17	X	X	18	D11/A2	GPIO24	Digital I/O
SPI	GPIO10	MOSI	19	X	X	20	GND		
	GPIO9	MISO	21	X	X	22	D12/A3	GPIO25	Digital I/O
	GPIO11	SCK	23	X	X	24	CE0	GPIO8	SPI (chip enable)
GND			25	X	X	26	CE1	GPIO7	
DO NOT USE	ID_SD	DO NOT USE	27	X	X	28	DO NOT USE	ID_SC	DO NOT USE
Digital I/O	GPIO5	D4	29	X	X	30	GND		
Digital I/O	GPIO6	D5	31	X	X	32	D13/A4	GPIO12	Digital I/O
PWM 2	GPIO13	D6	33	X	X	34	GND		
PWM 2	GPIO19	D7	35	X	X	36	D14/A5	GPIO16	PWM 1
Digital I/O	GPIO26	D8	37	X	X	38	D15/A6	GPIO20	Digital I/O
GND			39	X	X	40	D16/A7	GPIO21	Digital I/O

Figura 0.1. Diagrama de pines de la Raspberry Pi 3 Modelo B.

Anexo II Propuesta de implementación del sistema de supervisión y configuración local basado en IoT para casas de cultivo

Hardware utilizado

Para la implementación de este proyecto en 3 de las casas de cultivo de la UEB de Cultivos Protegidos “Valle del Yabú”, se requiere del montaje del siguiente hardware:

- Raspberry Pi 3 Modelo B (Ver Anexo I).
- Enrutador WiFi TP-Link Archer C2 AC750 Dual Band.

Figura 0.2. Vista frontal del Enrutador WiFi TP-Link Archer C2 AC750 Dual Band.

Figura 0.3. Vista trasera del Enrutador WiFi TP-Link Archer C2 AC750 Dual Band.

- Tablet Android Lenovo.

Tabla 0.1. Características principales del Tablet Lenovo.

Tamaño de la pantalla	10 pulgadas
Resolución máxima de pantalla	1280x800 píxeles
Procesador	Qualcomm Snapdragon Quad Core 1.3 GHz
Núcleos	4
RAM	2 GB
Sistema operativo	Android 6.0 Marshmallow
Tamaño de memoria flash	16 GB

Montaje

El montaje del sistema de supervisión y configuración local debe realizarse en el panel de control de la caseta de ferti-riego cercana a las 3 casas de cultivo donde se instaló la red de sensores inalámbricos basados en Zolertia Z1, la cual mide las variables medioambientales de temperatura, humedad relativa, humedad del suelo e intensidad luminosa.

El software del servidor se debe ejecutar mediante el archivo *ClientApp.py*, desde el Terminal del sistema operativo Raspbian instalado en la Raspberry Pi 3. Mientras que la aplicación Android GHMonitor fue instalada debidamente en el Tablet Lenovo.

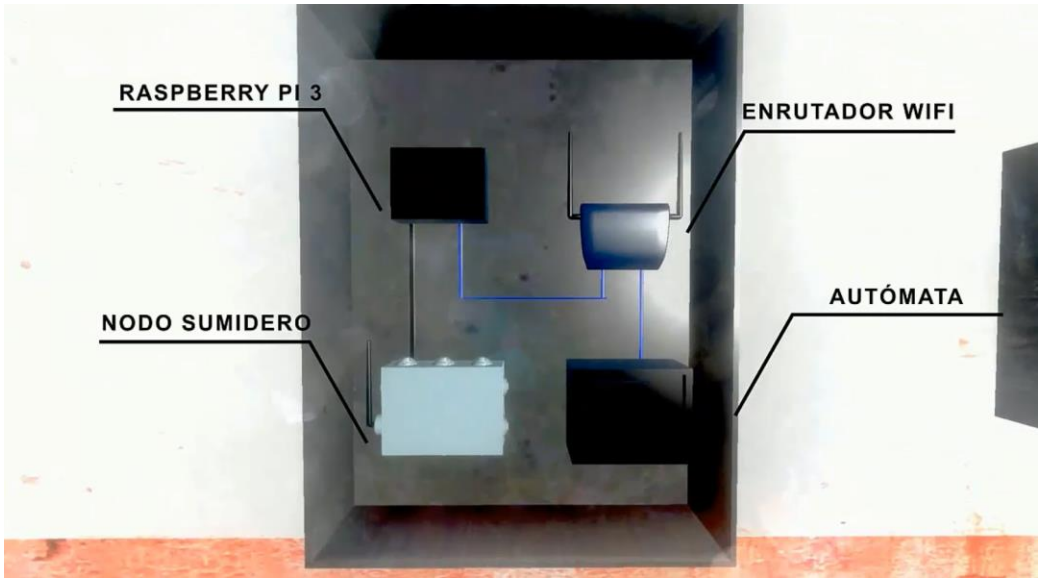


Figura 0.4. Simulación del montaje del panel de control de las casas de cultivo.

Figura 0.5. Caseta de ferti-riego.

Figura 0.6. Interior de las casas de cultivo.

Anexo III Requerimientos mínimos de la aplicación Android GHMonitor

Tabla 0.2. Requerimientos mínimos de un dispositivo móvil para el uso de la aplicación GHMonitor.

Procesador	1 GHz
RAM	1 GB
Tamaño de Pantalla	4,5 pulgadas
Versión del SO	4.4

Tabla 0.3. Requerimientos recomendados de un dispositivo móvil para el uso de la aplicación GHMonitor.

Procesador	2 GHz
RAM	2 GB
Tamaño de Pantalla	10 pulgadas
Versión del SO	6.0

Anexo IV Paquetes encriptados capturados utilizando el software Wireshark

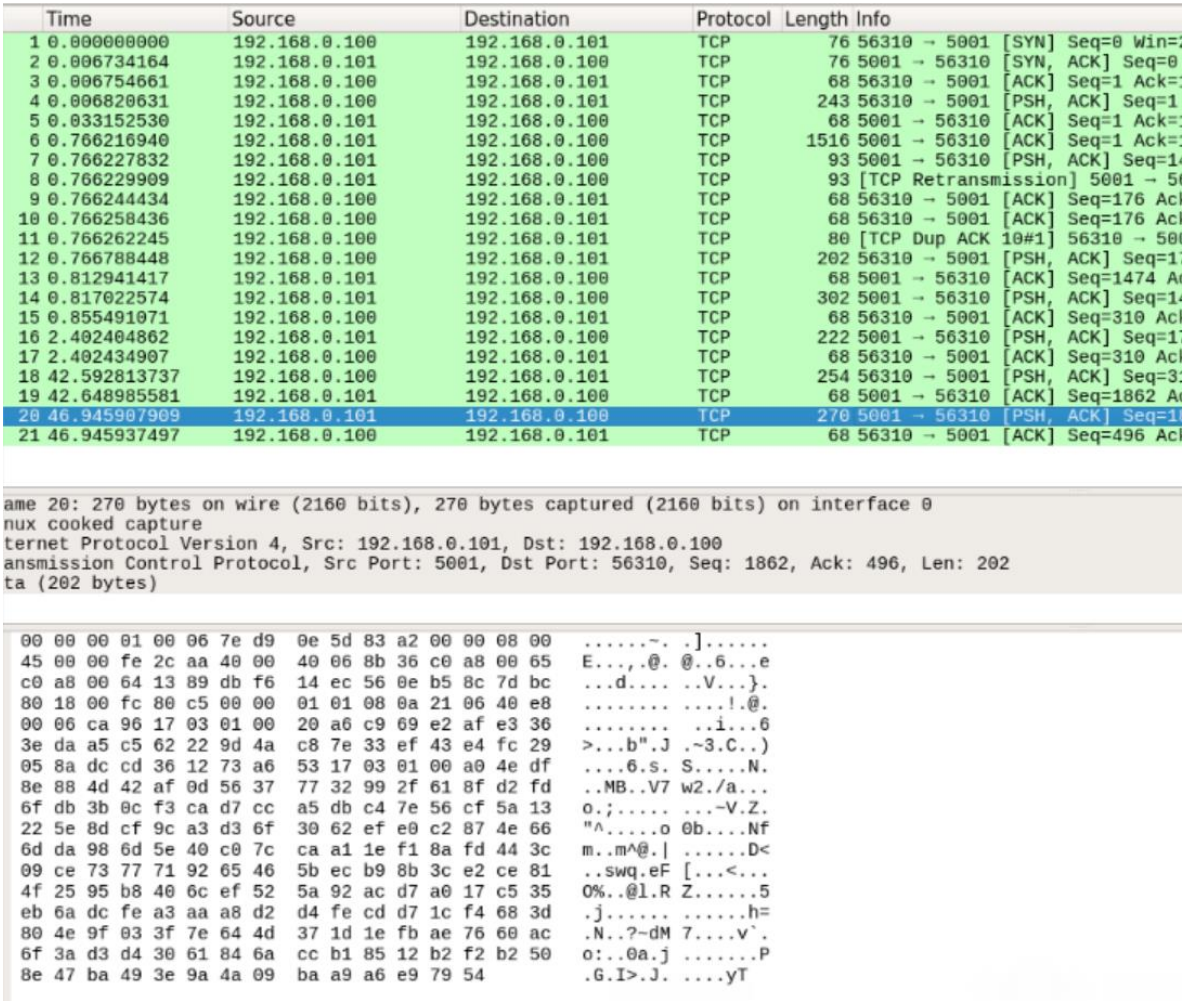


Figura 0.7. Paquetes encriptados capturados utilizando el software Wireshark.