



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

+



Esteban Mauricio Riquelme Torres

Estudio de compatibilidad entre Raspberry pi y Arduino

Informe Proyecto de Título de Ingeniero Civil Electrónico



Escuela de Ingeniería Eléctrica



PONTIFICIA UNIVERSIDAD
CATOLICA
DE VALPARAISO

Estudio de compatibilidad entre Raspberry pi y Arduino

Esteban Mauricio Riquelme Torres

Informe Final para optar al título de Ingeniero Civil Electrónico,

aprobada por la comisión de la

Escuela de Ingeniería Eléctrica de la

Pontificia Universidad Católica de Valparaíso

conformada por

Sr. David Velasco López

Profesor Guía

Sr. Francisco Alonso Villalobos

Segundo Revisor

Sr. Sebastián Fingerhuth Massmann

Secretario Académico

Valparaíso, 20 de abril de 2017

Este trabajo está dedicado a mis padres, familia y amigos.

Agradecimientos

Quiero agradecer a mis padres, por su amor y entrega incondicional con nuestra familia, por todo su esfuerzo a lo largo de los años, que me llevó a esta instancia, gracias por todo lo que entregan, gracias por el hogar, gracias por el amor y gracias por la herencia más importante, esta educación.

Juan, mi padre, gracias por mostrarme que el trabajo y el esfuerzo, traen consigo satisfacciones más allá de lo material, por el cariño y la entrega a nuestra familia, porque la vida que vivo es gracias a ti.

Lucinda, mi madre, gracias por la vida, por tu amor, por siempre tratar de formarme como una mejor persona, especialmente quiero agradecerte por todos tus sacrificios y espero poder tener al menos una fracción de tu fuerza, porque sé que sería suficiente para una vida entera.

Agradecer también a mis hermanas por ser la mayor y primera amistad, ellas han sido compañeras incondicionales en esta vida. María Fernanda gracias por ser un ejemplo de esfuerzo y dedicación, por ser una guía en esta vida y mostrar el camino por recorrer a los hermanos menores. Ana Beatriz gracias por enseñarme día a día con tu optimismo, gracias por inspirarme a superarme académicamente al demostrarme tu esfuerzo, dedicación e inteligencia.

Quiero agradecer también a mis amigos de la universidad que con su alegría y energía hicieron de mi paso por la institución una época inolvidable, siempre les tendré un gran cariño y les deseo mucha suerte en su futura vida profesional. A mis amigos de la vida, quienes me apoyaron siempre, gracias por el cariño que entregan y por las alegrías que pasamos juntos.

A mi abuela Mercedes, quien fue una segunda madre, gracias por tu amor y ayuda en toda mi educación, por demostrarme la capacidad infinita de amar que se esconde en el alma de todo ser humano, por demostrarme en vivo ejemplo la meta de una vida feliz.

Finalmente, quiero agradecer a Francisco Espíndola Riquelme, por su consejo y apoyo en este trabajo, gracias por compartir toda tu experiencia de años en desarrollo de tecnología, también por tu apoyo material, anímico y por brindarme el cariño de un hermano.

Valparaíso, 20 de abril de 2017

E. R

Resumen

El presente documento ha sido escrito para ser presentado como informe final de la asignatura proyecto de titulación. El trabajo expuesto en este informe, consiste en el desarrollo de una red de sensores, construida en base a las placas de desarrollo Arduino y Raspberry Pi.

El primer capítulo es una exposición de los antecedentes generales del proyecto, explica el contexto en el que se desarrolla el proyecto. Se comentan las tecnologías usadas, también la problemática que se pretende resolver, con los objetivos necesarios para resolver la problemática. Finalmente, se muestra el estado del arte de la compatibilidad entre Raspberry Pi y Arduino, para un análisis de las tendencias y falencias.

El segundo capítulo es una introducción a las redes de sensores, entregando una reseña histórica del desarrollo, en este capítulo se pretende dar a conocer los elementos que componen a la red de sensores y el trabajo ejecutado por cada elemento.

En el tercer capítulo se comentará la estructura final de la red de sensores desarrollada, partiendo con detallar el funcionamiento y uso de cada placa de desarrollo utilizada. Seguido de una explicación de los medios de comunicación elegidos, luego se detallará el esquema de la red. Siguiendo en el tercer capítulo se comentan los roles cumplidos por Arduino y Raspberry Pi, detallando la programación y los dispositivos utilizados para cumplir este rol.

El cuarto capítulo muestra los resultados finales de la red de sensores, con imágenes de la interfaz web con los gráficos y el control de pines de los nodos sensores, también se comenta el desempeño de cada elemento de la red, evaluando los aspectos más importantes de su funcionamiento.

El último capítulo entrega conclusiones sobre el desarrollo de la red de sensores, comentando las fortalezas del proyecto, en conjunto con posibles aplicaciones y trabajos futuros que pueden mejorar el desempeño o las funcionalidades de la red de sensores.

Palabras claves: Arduino, Raspberry Pi, red de sensores, servidor web, computación física.

Abstract

The present document has been written to be presented as the final report of the subject project of titration. The work outlined in this report, consists in the development of a network of sensors, built on the development boards Arduino and Raspberry Pi.

The first chapter is an exhibition of the general background of the project, explains the context in which the project develops. Discusses the technologies used, also the problem that is designed to resolve with the goals needed to resolve the problem finally displays the state of the art of the compatibility between Raspberry Pi and Arduino, for an analysis of the trends and shortcomings.

The second chapter is an introduction to sensors networks, giving a historical overview of the development, then will discuss the basics of its structure and work methodology, in this chapter is intended to give to know the elements that make up the network of sensors and the work carried out by each item.

In the third chapter will discuss the final structure of the network of sensors developed, starting with detail the operation and use of each development board used. Followed by an explanation of the communication media chosen, then will explain the network diagram. In the third chapter discusses the roles fulfilled by Arduino and Raspberry Pi, detailing the programming and the devices used to fulfill this role.

The fourth chapter shows the final results of the sensors network, with images of the web interface with the graphics and the control of pins for the sensors nodes, also comments on the performance of each element of the network, assessing the most important aspects of its operation.

Last chapter delivery the conclusions on the development of sensors network, commenting on the strengths of the project, in conjunction with possible applications and future work that may improve the performance or functionality of the network of sensors.

Key words: Arduino, Raspberry Pi, network of sensors, web server, physic computing.

Índice general

Introducción	1
Objetivo general	3
1 Antecedentes generales	4
1.1 Contexto del proyecto	4
1.2 Problemática.....	5
1.3 Objetivo general	5
1.3.1 Objetivos específicos	5
1.4 Estado del arte de la compatibilidad entre Raspberry Pi y Arduino	6
1.4.1 Análisis del estado del arte	8
1.5 Solución propuesta para la compatibilidad.....	9
1.6 Conclusión del capítulo	9
2 Redes de sensores.....	10
2.1 Historia de las redes de sensores	10
2.2 Estructura de una red de sensores	11
2.2.1 Nodos sensores	11
2.2.2 Nodos de subida	12
2.2.3 Base de datos.....	12
2.2.4 Medios de comunicación para una red de sensores.....	12
3 Implementación del proyecto	13
3.1 Placas de desarrollo.....	13
3.2 Raspberry Pi	14
3.2.1 Breve historia de Raspberry Pi.....	14
3.2.2 Aspectos técnicos.....	15
3.2.3 Utilización	17
3.3 Arduino.....	18
3.3.1 Breve historia de Arduino.....	18
3.3.2 Aspectos técnicos.....	19
3.3.3 Utilización	21
3.4 Shields Arduino	22

3.5 Problemática de la compatibilidad entre Raspberry Pi y Arduino	23
3.6 Medio de comunicación de la red.....	24
3.6.1 Comunicación Cableada	25
3.6.2 Comunicación Inalámbrica	27
3.7 Esquema de la red	31
3.8 Arduino como nodo sensor	32
3.8.1 Nodo sensor inalámbrico	32
3.8.2 Software del nodo sensor inalámbrico:	35
3.8.3 Nodo sensor cableado	40
3.8.4 Software nodo sensor cableado	42
3.9 Raspberry Pi como nodo de subida	44
3.9.1 Comunicación con la red de sensores	45
3.9.2 Construcción de la base de datos	46
3.9.3 Programa Nodo de subida	47
3.9.4 Visualización de los datos	51
3.9.5 Página Web RSM	53
4 Resultados	56
Discusión y conclusiones	62
Acerca de las placas de desarrollo:.....	62
Acerca de la compatibilidad entre las placas	63
Acerca del trabajo	64
Posibles aplicaciones	64
Trabajos futuros:	65
Acerca de los Objetivos	66
Bibliografía.....	67

Índice de figuras

Figura 1-1: Raspberry Pi y Arduino UNO conexión USB (fuente: www.meccanismocomplesso.org)	6
Figura 1-2: Placa de compatibilidad Arduberry (fuente: www.Dexterindustries.com)	7
Figura 2-1: Esquema red de sensores (fuente: elaboración propia)	11
Figura 3-1: Raspberry Pi (fuente: www.raspberrypi.org)	14
Figura 3-2: Partes Raspberry pi (fuente: elaboración propia)	15
Figura 3-3: Escritorio Raspberry Pi (fuente: www.raspberrypi.org)	18
Figura 3-4: Arduino Uno (fuente: www.arduino.cc)	19
Figura 3-5: Arduino IDE (fuente: www.arduino.cc)	21
Figura 3-6: Shield GPS (fuente: www.sparfun.com)	23
Figura 3-7: Trama Ethernet (fuente: www.lostintransist.se)	25
Figura 3-8: Trama IEEE802.11 (fuente: www.ecovi.uagro.mx)	28
Figura 3-9: Esquema Red de sensores mixta (fuente: elaboración propia)	31
Figura 3-10: Modulo ESP8266 (fuente: www.maxelectronica.cl)	33
Figura 3-11: Regulador de Voltaje Step-Down (fuente: www.maxelectronica.cl)	34
Figura 3-12: Convertidor bidireccional de niveles lógicos (fuente: www.rambal.cl)	34
Figura 3-13: Esquema ESP8266 con Arduino (fuente: elaboración propia)	35
Figura 3-14: Diagrama programa nodo Wi-Fi (fuente: elaboración propia)	37
Figura 3-15: Modulo Ethernet ENC28J60 (fuente: www.internetdelasoscos.cl)	41
Figura 3-16: Esquema Código nodo sensor cableado (fuente: elaboración propia)	42
Figura 3-17: Tabla de datos sensores (fuente: interfaz phpmyadmin)	46
Figura 3-18: Tabla valores pines (fuente: interfaz phpmyadmin)	47
Figura 3-19: Diagrama funcionamiento del programa servidor (fuente: elaboración propia)	48
Figura 3-20: Esquema servidor LAMP (fuente: www.diy-makers.es)	52
Figura 4-1: Página principal del proyecto (fuente: elaboración propia)	58
Figura 4-2: Pagina de datos históricos (fuente: elaboración propia)	59
Figura 4-3: Calendario para selección de fecha (fuente: elaboración propia)	59
Figura 4-4: Pagina de datos en tiempo real (fuente: elaboración propia)	60
Figura 4-5: Interfaz de control pines Arduino (fuente: elaboración propia)	60

Indice de tablas

Tabla 3-1: Tabla comparativa de características Raspberry Pi	17
Tabla 3-2: Características Atmega328P Arduino UNO (fuente:www.atmel.com)	21
Tabla 3-3: Comparativa entre Raspberry Pi y Arduino	23
Tabla 3-4: Contenido de la trama Ethernet	26
Tabla 3-5: Tabla explicativa de la trama Wi-Fi.....	29
Tabla 3-6: Subtrama de Frame Control	30
Tabla 3-7: Ejemplos comandos AT.....	35
Tabla 3-8: Tabla de respuestas comandos AT.....	36
Tabla 3-9: Formato de Mensajes	39
Tabla 3-10: Formato mensaje de control de pines	39
Tabla 3-11: Conexión del ENC28J60 según librería	41

Glosario de terminos

GPIO: General Pin Input Output
IDE: Integrated Development Environment
IoT: Internet of Things
USB: Universal Serial Bus
LAN: Local Area Network
SOSUS: Sound Surveillance System
NOAA: National Oceanic and Atmospheric Administration
MEMS: MicroElectroMechanical Systems
UART: Universal Asynchronous Receiver-Transmitter
CAN: Controller Area Network
SoC: System on a Chip
HDMI: High-Definition Multimedia Interface
I2C: Inter-Integrated Circuit
SPI: Serial Peripheral Interface
SSH: SECURE SHELL
IDII: Interaction Desing Institute Ivrea
PIC: Programmable Interrupt Controller
MIT: Massachusetts Institute of Technology
ICSP: In Circuit Serial Programming
PC: Personal Computer
RAM: Random Access Memory
EEPROM: Electrically Erasable Programmable Read-Only Memory
CLP: Pesos chilenos
TTL: Transistor-Transistor Logic
CMOS: Complementary Metal-Oxide-Semiconductor
IEEE: Institute of Electrical and Electronics Engineers
UTP: Unshielded Twisted Pair
TCP: Transmission Control Protocol
UDP: User Datagram Protocol
MAC: Media Access Control
CSMA/CD: Carrier Sense Multiple Access Collision Detect
WLAN: Wireless Local Area Network

DSSS Direct- Sequence Spread Spectrum

QoS: Quality of Service

Orthogonal Frequency Division Multiplexing (OFDM)

CRC: Cyclic Redundancy Check

P2P: Peer to Peer

AT: Attention

CD: Compact Disc

LDR: Light Dependent Resistor

AC/DC: Alternating Current / Direct Current

SQL: Structured Query Language

GUI: Graphical User Interface

PHP: Personal Home Pages

HTTP: Hypertext Transfer Protocol

LAMP: Linux Apache MySQL Php

CSS Cascading Style Sheets

DIY: Do It Yourself

AJAX: Asynchronous JavaScript And XML

DNS: Domain Name System

OECD: Organisation for Economic Co-Operation and Development

I+D: Investigación y Desarrollo

OEM: Onda Electro-Magnetica

Introducción

El desarrollo de la electrónica ya no está solamente asociado al área profesional, ya que muchas empresas, que producen herramientas para el desarrollo tecnológico, han adoptado la visión de negocios Open-Source [1]. Permitiendo el libre acceso a sus desarrollos de software, especificaciones técnicas y herramientas de desarrollo, fomentando el uso de sus productos, entre estudiantes y entusiastas de la electrónica, para programar y desarrollar proyectos electrónicos, sin necesariamente tener una formación profesional, pero se interesan y aprenden de forma autodidacta para desarrollar sus propios proyectos. Con la continua evolución de la tecnología, se han podido generar herramientas cada vez más poderosas y de bajo costo, lo que derivó en la aparición de las placas de desarrollo, son tarjetas de circuito impreso con capacidades que van desde leer señales análogas y digitales, a una tarjeta con la potencia de procesamiento de un computador de escritorio, las placas de desarrollo permiten generar proyectos con grandes requerimientos tecnológicos, sin la necesidad de tener todos los conocimientos para construirlo desde cero. El presente trabajo es desarrollado en base a Raspberry Pi y Arduino, dos placas de desarrollo que tienen la mayor cantidad de documentación y uso, en el medio de las placas de desarrollo.

La placa de desarrollo Raspberry Pi es un computador embebido, esto quiere decir que cuenta con todas las funcionalidades de un computador, pero, físicamente es de tamaño reducido, esto sin contar los periféricos que se asocian normalmente a un computador. Raspberry Pi es del tamaño de una tarjeta de crédito, en comparación a otras tarjetas de desarrollo, tiene un gran poder de procesamiento de datos, capaz de realizar tareas de reconocimiento de patrones o análisis de imágenes, al igual que su capacidad de almacenar información, que permite implementar distintos tipos de servidores y bases de datos. Raspberry Pi como cualquier computador utiliza sistemas operativos, por lo que tiene un ambiente de trabajo en el que se puede programar en varios lenguajes, si se cuenta con el compilador indicado. Raspberry Pi también cuenta con terminales de salida y entrada de señales digitales (GPIO), que permiten trabajar con electrónica externa en los rangos lógicos de 0[V] (nivel bajo) a 3,3[V] (nivel alto).

Arduino es una plataforma para trabajar con micro-controladores Atmel, esta plataforma es una placa de circuito impreso, que contiene al micro-controlador en conjunto con todos los circuitos necesarios para su correcto funcionamiento y programación. Arduino tiene una interfaz USB y un software denominado ambiente de programación integral (IDE), ambos

elementos, permiten programar el micro-controlador en un lenguaje de alto nivel llamado Arduino, desde cualquier computador con conexión USB. En comparación con otras placas de desarrollo, los micro-controladores tienen una capacidad de procesamiento y almacenamiento de datos muy baja, pero, suficiente para escribir rutinas de control de procesos. Arduino puede leer y emitir señales analógicas y digitales (Bajo 0[V], Alto 5[V]), con lo cual se puede tener acceso al uso de cientos de sensores y electrónica externa. Se puede resumir la fortaleza de Arduino en que permite implementar un código que interactúe con el ambiente, en diversas formas, de forma sencilla y a bajo costo.

Existen dispositivos electrónicos diseñados para ser usados con Arduino, se les denomina Shields, estos dispositivos cumplen con otorgar funcionalidades ausente en Arduino. Por ejemplo, un Shield que permite comunicar a la placa Arduino por Ethernet o Wi-Fi, un Shield con sensores, matrices de luces led o un Shield que permite el control de motores DC o paso a paso. Los Shields se conectan directamente sobre Arduino, la distribución física de los pines del Shield y Arduino es la misma, por lo que físicamente ocupa todos los pines, pero en el funcionamiento sólo se utilizan algunos.

La compatibilidad es una cualidad, de lo que puede existir o interactuar de forma armónica con otros elementos. En base a esa definición, la compatibilidad de Raspberry Pi con Arduino, implica una comunicación entre ambas placas, que permita un trabajo coordinado, para lograr un objetivo en conjunto.

Desde la perspectiva de los proyectos de computación física, las placas de desarrollo comentadas tienen características que se complementan. Por parte de Raspberry Pi, se tiene un gran poder de procesamiento, múltiples plataformas de programación y almacenamiento de datos. Por parte de Arduino, existe una gran cantidad de sensores y electrónicas externas compatibles, con esto se pueden crear proyectos con requerimientos en Hardware y Software que las placas por si solas no podrían cumplir. La compatibilidad entre Raspberry y Arduino es aplicada en varios trabajos y con distintos objetivos, por ejemplo el uso de las placas para proyectos de robótica [2] o domótica [3].

Existen impedimentos para la interacción entre las placas, por una parte están los rangos de voltaje utilizados por ambas, Raspberry Pi usa voltajes de máximo 3.3[V], cualquier señal de mayor voltaje podría quemar la tarjeta, Arduino utiliza señales de 5[V] por lo que quemaría la Raspberry Pi si le envía una señal digital, por otra parte, si Raspberry Pi le enviase una señal de 3.3[V] a Arduino este no la reconocería como un nivel lógico alto, por estas razones cualquier comunicación por los pines de entrada y salida digitales queda descartada.

La compatibilidad entre Raspberry Pi y Arduino generaría una poderosa plataforma para la generación de distintos tipos de proyectos, reduciendo los costos y dificultad de los mismos, sin escatimar en calidad ni capacidad de computación. Un campo de uso importante para la compatibilidad entre Raspberry Pi y Arduino es el área académica, estas herramientas de bajo costo son normalmente asociadas a los estudiantes y hobbistas en electrónica, la compatibilidad entre las placas de desarrollo pueden servir a muchos estudiantes, para realizar proyectos de mayor complejidad en un rango accesible de tiempo y precio, pudiendo incentivar conceptos

como la investigación, innovación y una mayor probabilidad de emprendimientos por parte de alumnos.

Actualmente, pueden encontrarse varias soluciones a la compatibilidad entre Raspberry Pi y Arduino, algunas son productos comerciales, otras ideas abiertamente publicadas, todas esas soluciones fueron investigadas para determinar un punto de referencia y base para la innovación en la compatibilidad de placas de desarrollo.

Desde una investigación de las tecnologías de compatibilidad, se llegó al desarrollo de las redes de sensores. Libros como [4] y [5] explican las redes de sensores como una colección de sensores comunicados por cable o por radio frecuencia, distribuidos lógicamente o geográficamente, para recolectar información del ambiente. La información recopilada por los sensores, es transmitida a un elemento que integra la información basado en alguna estrategia predefinida, para luego entregar la información a un usuario, que pueda inferir algo acerca del ambiente en el que fueron distribuidos los sensores. Las redes de sensores, están presentes en varios desarrollos tecnológicos actuales, como las Smart Cities y el IoT (Internet of Things).

Existen desarrollos de redes de sensores, implementados con Arduino y Raspberry Pi, en [6] y [7] se encontraron implementaciones de redes inalámbricas de bajo costo, en base a las placas de desarrollo mencionadas y placas Xbee para la comunicación por radiofrecuencia. Estos trabajos evidencian la posible compatibilidad entre Raspberry Pi y Arduino, ya que se logra una comunicación entre las placas, que por medio de un algoritmo predefinido coordinan los trabajos de ambas, para lograr un objetivo en conjunto.

Es en base a los trabajos comentados, que se sustenta el trabajo expuesto en este informe, el cual consiste en el diseño e implementación de una red de sensores con Raspberry Pi y Arduino. Se busca lograr la compatibilidad entre Raspberry Pi y Arduino, para generar un conocimiento de la compatibilidad entre placas de desarrollo, en conjunto con crear una herramienta que sirva como punto de partida para el desarrollo de proyectos electrónicos más complejos, a partir de Raspberry Pi y Arduino.

Parte del trabajo desarrollado, consiste en la construcción de una interfaz web, para la visualización de los datos obtenidos por la red de sensores. Trabajo que consiste en la creación de base de datos, servidores web y programación de páginas web, desarrollados con lenguajes HTML, CSS, PHP y Javascript, en conjunto con interfaces de administración de bases de datos como MySQL. Otra característica importante, del trabajo que se mostrará en este informe, es la implementación de un sistema de control de salidas digitales de Arduino, por medio de la interfaz web.

Objetivo general

- Lograr compatibilidad entre Raspberry Pi y Arduino.

1 Antecedentes generales

En este capítulo se entrega una introducción al proyecto, se comentará el contexto, en el cual se desarrolla el proyecto. Se expone la problemática que se quiere resolver, el objetivo de la compatibilidad entre las placas mencionadas, un estado del arte acerca de la compatibilidad entre placas de desarrollo, indicando las dificultades y beneficios de lograr la compatibilidad.

1.1 Contexto del proyecto

La palabra proyecto, proviene desde el latín proicere, sus raíces son pro (adelante) e iacere (lanzar) [8]. Esta definición etimológica, permite entender que el concepto de proyecto, es la idea o visión, de un elemento en el futuro. El desarrollo de proyectos, es la planificación y ejecución de acciones, con el fin de lograr el elemento buscado, con las características de la visión original.

En el campo de la electrónica, los proyectos buscan solucionar una problemática, mediante el desarrollo, diseño o implementación de tecnologías. Los proyectos electrónicos, se pueden dividir en dos grandes áreas, una es la construcción física de dispositivos electrónicos (Hardware), otra área es la creación de herramientas intangibles (Software), conformadas por algoritmos programados, que permiten manejar el comportamiento de una máquina.

Los desarrollos de proyectos electrónicos, tienen características que son influenciadas por el contexto en el que se desarrollan. El contexto de un proyecto electrónico, se puede definir en mayor parte, por el nivel de la tecnología a la cual se puede acceder y los requerimientos de la problemática.

Este proyecto, el cual es desarrollado en la Pontificia Universidad Católica de Valparaíso, está enfocado en lograr compatibilidad, entre las placas de desarrollo Raspberry Pi y Arduino, ya que son herramientas ampliamente utilizadas en la docencia, a nivel mundial. La tecnología utilizada para este proyecto, se limita a las compatibles con las placas de desarrollo utilizadas.

Raspberry Pi y Arduino, tienen características complementarias, que al combinarse permitirían generar una poderosa herramienta para el desarrollo de proyectos electrónicos, El concepto de compatibilidad entre dispositivos electrónicos, hace referencia a la capacidad de los dispositivos, para ejecutar sus procesos de trabajo en conjunto, coordinándose mediante una

forma de comunicación. Para este proyecto, la problemática es lograr la compatibilidad entre Raspberry Pi y Arduino, que implica generar una comunicación y coordinación, de los procesos ejecutados por las placas, para lograr utilizar las características de cada placa en conjunto.

1.2 Problemática

Raspberry Pi y Arduino, son dos de las placas de desarrollo más usadas a nivel mundial, se emplean en la docencia de robótica, informática y computación física, también son utilizadas para la investigación y desarrollo de proyectos electrónicos y computacionales.

Raspberry Pi, es un computador de placa única. Al tener la estructura de un computador, se tiene una gran versatilidad de programación, soportando varios lenguajes, aplicaciones e interfaces de gran complejidad, también una fácil conectividad a redes de internet. Esto hace de Raspberry Pi, una herramienta para la implementación de casi cualquier proyecto, de electrónica e informática. Arduino, es un módulo para el uso y programación de micro-controladores, se destaca por ser compatible con una gran cantidad de sensores y dispositivos electrónicos, también por tener una amplia documentación en el uso de la electrónica externa compatible.

Se pueden combinar las características mencionadas, para obtener una herramienta con un gran potencial para el desarrollo de proyectos electrónicos, tanto en Hardware como en Software. La idea principal, es utilizar todo el poder de procesamiento y versatilidad de Raspberry Pi, con la gran gama de sensores y dispositivos compatibles con Arduino.

En primera instancia, los dispositivos compatibles con Arduino, no pueden ser utilizados en Raspberry Pi, ya que los rangos de voltajes utilizados por las placas no son compatibles. Este problema, puede ser superado, al generar una compatibilidad entre Raspberry Pi y Arduino, esto permitiría usar la electrónica y sensores compatibles de Arduino, desde Raspberry Pi de forma indirecta.

Una segunda condición, es lograr la compatibilidad entre las placas mencionadas, de una forma diferente a las soluciones existentes, ya que la compatibilidad entre Raspberry Pi y Arduino, ha sido lograda por otros proyectos. La solución propuesta, debe innovar en el campo de la compatibilidad entre placas de desarrollo.

1.3 Objetivo general

El objetivo principal del proyecto, es generar una plataforma de compatibilidad, entre las placas de desarrollo Raspberry Pi y Arduino. Con este objetivo, se busca unir las capacidades de Raspberry Pi en procesamiento, almacenamiento y versatilidad, con las capacidades de Arduino, pudiendo utilizar una gran gama de sensores y electrónica externa.

1.3.1 Objetivos específicos

La compatibilidad de dispositivos electrónicos, se forma mediante una comunicación y coordinación de los procesos, por lo que conseguir la compatibilidad, implica construir una

comunicación compatible con ambas placas, en conjunto con un algoritmo de coordinación de los procesos ejecutados por ambas. A continuación, se listan los objetivos específicos del proyecto, que buscan lograr las afirmaciones comentadas previamente.

- Lograr la comunicación y transmisión de datos entre las placas de desarrollo, por medio de un algoritmo de comunicación definido para ambas placas, en síntesis un objetivo específico del proyecto es la generación de una red de comunicación.
- Otro objetivo específico que está estrechamente relacionado con el uso de la red, es la implementación de una base de datos local en Raspberry Pi, para la gestión de la información obtenida por ambas placas.
- Parte importante del proyecto es poder tener una plataforma para ver la información adquirida por las placas, un objetivo específico del proyecto es lograr implementar una plataforma de visualización de los datos obtenidos.

1.4 Estado del arte de la compatibilidad entre Raspberry Pi y Arduino

Como se había comentado, existen trabajos previos, que lograron la compatibilidad entre Raspberry Pi y Arduino. Este subcapítulo, es un análisis de los trabajos recopilados, para obtener una visión de las tendencias, fortalezas y falencias en el campo de la compatibilidad. Con el fin de desarrollar una propuesta propia, con un desarrollo o implementación de tecnología para la compatibilidad.

La forma más básica de compatibilidad, entre Raspberry Pi y Arduino, es la conexión USB. En ambas placas de desarrollo, se dispone de un puerto USB (ver figura 1-1), también se pueden programar algoritmos de lectura, análisis de acción y escritura de datos por medio del bus USB [9]. Al comunicar ambas placas por este puerto, se puede generar la transferencia de datos, para la coordinación y comunicación de los procesos ejecutados. La comunicación por USB, sirve como base para nuevos proyectos de compatibilidad, entre las tecnologías de Raspberry Pi y Arduino.

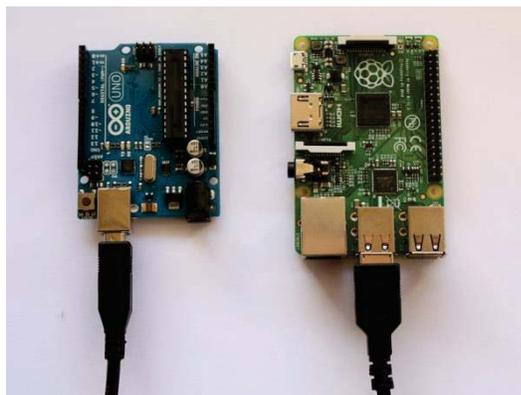


Figura 1-1: Raspberry Pi y Arduino UNO conexión USB (fuente: www.meccanismocomplesso.org)

Durante la investigación, se encontraron proyectos denominados placas de compatibilidad (ver figura 1-2) [10] [11] [12] [3], cumplen la función de integrar Raspberry Pi con Shields y sensores de Arduino.

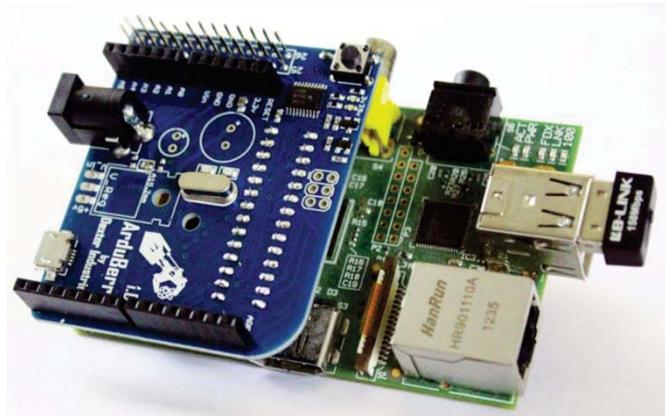


Figura 1-2: Placa de compatibilidad ArduBerry (fuente: [www. Dexterindustries.com](http://www.Dexterindustries.com))

Estas placas, se montan sobre los pines de Raspberry Pi, incluyen un microcontrolador Atmega328P, microcontrolador del Arduino UNO, el cual se programa con el software IDE de Arduino (desde Raspberry Pi u otro computador), para controlar los Shields y sensores conectados. La placa de compatibilidad sirve de intermediario entre Raspberry Pi y los productos de Arduino, ya que se puede programar la misma comunicación USB descrita en [9], para enviar los datos de los sensores y Shields a Raspberry Pi, también para recibir datos y coordinar los procesos de ambas placas.

La compatibilidad, en los trabajos comentados, se desarrolla en una relación de uno es a uno, o sea entre una Raspberry Pi y un Arduino. Existen trabajos más recientes, que llevan la compatibilidad a una relación entre múltiples agentes, a esta estructura de comunicación se les denomina redes de sensores, a cada elemento o conjunto de elementos, que componen la red, se les denomina nodo.

Existe un desarrollo de red, entre una Raspberry Pi y varios Arduino, por medio de comunicación inalámbrica [13], con fines de monitoreo ambiental, la comunicación inalámbrica se logra con placas Xbee [14], compatibles con Arduino.

Cada Arduino, se conectan con un sensor y una placa Xbee. El sensor extrae información del ambiente, la cual se transmite por la placa Xbee a la red inalámbrica, a este conjunto de elementos descritos, se les denominó como nodo sensor.

Raspberry Pi, se conecta por USB con un Arduino, el cual se conecta con una placa Xbee. La placa Xbee, recibe la señal inalámbrica de todos los nodos sensores, pasando esa información al Arduino, el cual transmite los datos a Raspberry Pi, para que se procesen y almacenen los datos, a este conjunto de elementos, se les denomina nodo de datos.

Una base de datos local, en conjunto con una interfaz de visualización web, se programaron en Raspberry Pi. La información, obtenida por los sensores de Arduino, se puede visualizar y administrar, en la página web creada en Raspberry Pi. La página creada, puede ser accedida desde cualquier dispositivo con navegador web, que esté conectado a la misma red LAN de Raspberry Pi.

Un libro [15], enseña sobre el uso de Raspberry Pi, Arduino y placas Xbee, para la implementación de redes inalámbricas. En el documento mencionado, no se trata la compatibilidad entre las placas de desarrollo, pero si da indicaciones de cómo utilizar cada elemento, para construir una red de sensores.

1.4.1 Análisis del estado del arte

La compatibilidad, entre Raspberry Pi y Arduino, ha sido lograda de diversas formas. Según lo visto, en el estado del arte, se puede apreciar una evolución de los métodos para lograr la compatibilidad.

Los primeros métodos, se basan en una comunicación cableada USB, con un estándar de comunicación muy confiable, compatible con las placas de desarrollo, de fácil implementación y alta velocidad.

Las placas de compatibilidad, son fácilmente utilizables, al ser técnicamente igual a un Arduino, se pueden programar con todas esas funcionalidades, pero obligan a tener una compatibilidad, entre sólo una Raspberry Pi y un Arduino. Teniendo en cuenta, la capacidad de Raspberry Pi, para procesar y almacenar datos, se puede considerar que su capacidad es subutilizada en este método de compatibilidad.

La conexión por cable USB, entre Raspberry Pi y una placa Arduino, puede usarse en los 4 puertos USB de Raspberry Pi, pero existiría la restricción de un largo máximo de 5[m] [16], sumado a la cantidad limitada de puertos, se pone un límite a las posibilidades de un sistema con mayor cobertura de área y escalabilidad.

Las redes de sensores, son un método de compatibilidad altamente escalable, dinámico y que aprovecha de mejor forma, las capacidades de ambas placas de desarrollo. El hecho, de trabajar con una interfaz web, hace que el proyecto se pueda usar desde cualquier dispositivo de uso diario, como computadores o celulares modernos.

El trabajo mencionado, presenta algunas falencias, ya que se utiliza sólo un sensor por cada placa Arduino, lo que se puede entender como un error de aplicación, no de estructura. Otra falencia, es la comunicación simplex, o sea que sólo hay intercambio de información en un solo sentido, desde los Arduinos a Raspberry Pi, lo cual impide una mayor interacción entre los procesos de cada placa.

1.5 Solución propuesta para la compatibilidad

Desde la investigación de compatibilidad, entre placas de desarrollo, se encontraron soluciones a la problemática del proyecto, pero, no aprovechaban la potencialidad de la compatibilidad, ignorando las características de procesamiento de Raspberry Pi.

Para solucionar la problemática de la compatibilidad entre Raspberry Pi y Arduino, se propone la implementación de una red de sensores. Con esta estructura, se puede aprovechar de mejor forma las capacidades de Raspberry Pi, tanto en almacenamiento como en procesamiento, al igual que las de Arduino, con el uso de varios sensores y actuadores. Este proyecto, pretende utilizar recursos, obtenidos desde el estudio del estado del arte, pero también aportar con ideas propias en el campo de la compatibilidad.

Se pretende crear una red de sensores, la cual sea accesible por una interfaz web, con una base de datos local en Raspberry Pi. Alguno de los cambios realizados, es la implementación de una comunicación Half-Duplex, entre Raspberry Pi y las placas Arduino, este cambio permitirá activar y desactivar salidas digitales de Arduino, de forma remota, desde una página web de autoría propia. Otro cambio, a la estructura expuesta en el estado del arte, es el medio de comunicación usado, en lugar de usar placas Xbee, se utilizarán conexiones Ethernet y Wi-Fi, para poder tener una red de sensores en una LAN, que es la red de uso más común en hogares y oficinas.

1.6 Conclusión del capítulo

Desde el análisis a las características de Raspberry Pi y Arduino, se concluyó que Raspberry Pi y Arduino presentan fortalezas que son complementarias, en base a eso, se pensó que la compatibilidad debería aprovechar de la mejor forma posible las capacidades de ambas placas de desarrollo. Finalmente, se encontró una estructura que cumple con los requerimientos comentados, la red de sensores puede ser utilizada para aprovechar la capacidad de procesamiento y almacenamiento de datos de Raspberry Pi, en conjunto con una utilización de Arduino, que le exige el control de datos y procesamiento para la comunicación con la red.

2 Redes de sensores

En este capítulo, se explica de forma más detallada la estructura de una red de sensores. Primero, se entrega una breve reseña histórica del desarrollo de las redes de sensores, luego se explican los elementos que conforman la red, explicando los roles que cumple cada parte. Finalmente, se define el cómo interactúan los elementos de la red, presentando los distintos medios de comunicación disponibles.

2.1 Historia de las redes de sensores

La tecnología de las redes de sensores partió por aplicaciones de defensa en Estados Unidos, como medio para detectar actividad enemiga submarina. Durante la guerra fría, se creó el sistema de sonido para la supervivencia SOSUS (Sound Surveillance System) [17], un arreglo de hidrófonos ubicados en el fondo marino detectaba y rastreaba submarinos soviéticos. Actualmente SOSUS es usado por la administración oceánica y atmosférica nacional (NOAA), para el monitoreo de animales y eventos sísmicos en el océano [18].

Durante la guerra fría, redes de radares de defensa aérea eran utilizados para la defensa de suelo estadounidense y canadiense [17]. Las primeras apariciones de redes de sensores, tenían como parte clave la interpretación de las mediciones por parte de personas entrenadas, por lo que identificar e interpretar quedaron como puntos de evolución para las redes de sensores modernas [17].

El avance de la tecnología, ha permitido construir redes de sensores con dispositivos cada vez más pequeños y económicos, basados en sistemas micro-electromecánicos (MEMS). Cuando las tecnologías relacionadas a las redes de sensores avanzan, el costo de las redes baja y las aplicaciones toman parte en los campos industriales y comerciales [19].

Las redes de sensores, han sido seleccionadas dentro de las 21 tecnologías más importantes para el siglo 21 [20], una aseveración hecha en 1999 que se ve comprobada en nuestros días. Actualmente, el desarrollo de las redes de sensores inalámbricas es la pieza clave en las tendencias tecnológicas más importantes a nivel mundial, como la Internet de las cosas y las Smart city. En una Smart city, las redes de sensores, cámaras, aparatos inalámbricos y centro de datos forman la infraestructura clave para que las autoridades puedan distribuir los recursos de forma más rápida y eficiente [21].

2.2 Estructura de una red de sensores

Una red de sensores, puede ser descrita como un grupo de nodos que cooperan en la medición y control del ambiente, permitiendo la interacción entre personas o computadores y el ambiente que los rodea [22]. Las redes de sensores actuales constan de nodos sensores, nodos de subida, base de datos y clientes.

Un grupo de nodos sensores, se distribuye para el monitoreo de procesos de interés, para luego traducir y enviar la información a un nodo de subida, el cual administrará y entregará la información a la base de datos y al usuario final, que puede ser una persona o máquina (ver figura 2-1).

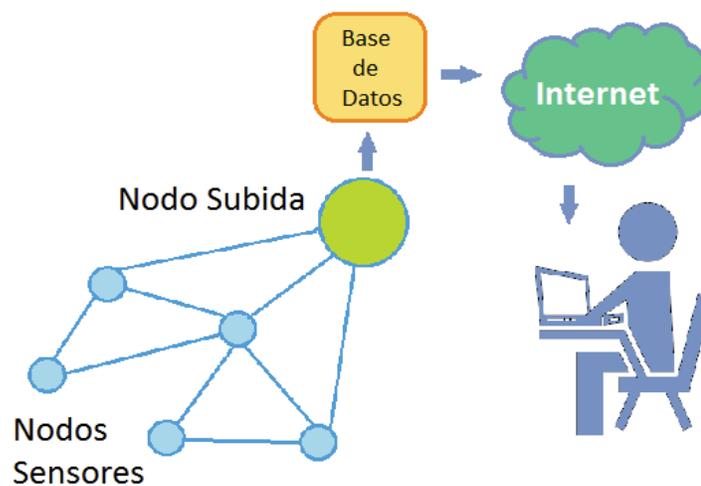


Figura 2-1: Esquema red de sensores (fuente: elaboración propia)

A continuación, se comentarán más en detalle la funcionalidad de cada una de las partes que componen una red de sensores. Trabajos como [15], entregan una visión básica sobre la estructura de una red de sensores.

2.2.1 Nodos sensores

Son los encargados de extraer información o actuar sobre el ambiente, también de transmitir la información obtenida a la red. Los nodos sensores se componen de 3 partes básicas, un sensor, un controlador y un comunicador.

El sensor es un transductor, convierte las variaciones y magnitudes de fenómenos físicos en señales eléctricas, estas señales son recibidas por el controlador, quien las interpreta y convierte a una magnitud entendible y acorde al proceso censado. Finalmente, el comunicador modula la información, a un formato adecuado para el medio de transmisión de la red.

Otra función del nodo sensor, es la de actuar sobre el medio, según las instrucciones dadas por el usuario. En ese proceso, la información viaja desde el usuario al nodo sensor, de forma que el

comunicador recibe y traduce la información para el controlador, el cual debe interpretarla para luego actuar sobre algún elemento de control que alterará el ambiente.

2.2.2 Nodos de subida

Son los encargados de la distribución y administración de la información en la red, los nodos de subida constan de partes de comunicación, subida y almacenamiento. Normalmente, estos nodos tienen un controlador más avanzado que los nodos sensores, ya que las tareas que debe realizar son mucho más complejas.

Toda la información obtenida por los nodos sensores, es enviada a los nodos de subida, en esta etapa de la red la información es almacenada, en una base de datos de forma local o remota, identificando cada dato con su procedencia y tiempo. Otra tarea del nodo de subida, es enviar la información al usuario final, a través de un medio que permita una lectura comprensible.

2.2.3 Base de datos

En la base de datos se almacena toda la información adquirida por la red de sensores, se acostumbra a tener una base de datos, para poder tener un respaldo de la información, trabajar con datos históricos y asegurar una red robusta en caso de errores. La base de datos, puede ser local o remota, pero debe cumplir con la condición de organizar la información de forma ordenada, para identificar el origen y validar cada dato.

2.2.4 Medios de comunicación para una red de sensores

Las redes de sensores necesitan un medio físico de comunicación para vincular todos los nodos, la transmisión de la información puede ser realizada por comunicación cableada, inalámbrica o híbrida.

- Cableada: tienen la ventaja de ser más confiables contra las pérdidas de datos y son relativamente más fáciles de implementar, la mayor desventaja es que las redes cableadas tienen una menor flexibilidad, ante cambios estructurales. El cableado Ethernet con conectores RJ45 son los más usados para estas redes, se encontraron desarrollos con comunicación serial UART, otro método usado es el estándar de comunicación RS485.
- Inalámbrica: es el medio de comunicación más popular y versátil en el desarrollo de redes de sensores, las tecnologías más usadas son Wi-Fi y Bluetooth para comunicación cercana. Para comunicaciones de mayor distancia se utilizan emisores y receptores de radio frecuencia, otro método es usando los protocolos Zigbee por medio de la placa de comunicación Xbee.
- Híbrida: mezcla en tramos ambos tipos de comunicación, usado en redes de sensores más complejas, un ejemplo de esta implementación sería conectar los nodos sensores por Xbee a los nodos de datos y estos por cable Ethernet al nodo de subida. Toda configuración depende de las necesidades que debe resolver la red.

3 Implementación del proyecto

En este capítulo, se expone el trabajo realizado durante el desarrollo del proyecto, explicando las características principales de las placas de desarrollo utilizadas. Se define la composición del medio de comunicación, los nodos sensores, nodos de datos, la base de datos, también se define el método de visualización para los datos de la red. Se explica el cómo se utilizarán Arduino y Raspberry en la red de sensores en conjunto con todos los elementos y programación necesarios para hacerlo.

3.1 Placas de desarrollo

Actualmente, el desarrollo de proyectos electrónicos se ha popularizado, en parte, por la aparición de las placas de desarrollo. Las placas de desarrollo, son herramientas para la realización de proyectos electrónicos. Éstas se construyen en base a un conjunto de circuitos y sistemas electrónicos, esenciales para crear sistemas de control, sensores, robótica o automatización.

Existe una gran variedad de placas de desarrollo disponibles en el mercado, desde placas con las características de un micro-controlador, hasta placas que tienen toda la arquitectura y funcionalidades de un computador moderno.

Una característica común en las placas de desarrollo, son los puertos de entrada y salida para señales digitales, llamados pines. Estos puertos, permiten la interacción de las placas con dispositivos electrónicos externos, como sensores, relés, controladores de motores, dispositivos de comunicación (Wi-Fi, Radiofrecuencia, Ethernet, CAN BUS, etc.), que cumplen funciones irrealizables para la placa de desarrollo, pudiendo así formar un sistema más complejo, controlado por la placa.

Debido a la simpleza de las placas de desarrollo, sus capacidades son a veces superadas por las exigencias de los desarrolladores. Paulatinamente, se han ido creando dispositivos compatibles para dar nuevas funcionalidades a las placas, desde puertos Ethernet y Wi-Fi a controladores de grandes motores y sensores. Con las tarjetas de desarrollo y todas sus herramientas integradas, se puede generar un proyecto electrónico complejo sin la necesidad de desarrollar todas y cada una de las partes que lo componen.

Las placas de desarrollo utilizadas en este proyecto son Raspberry Pi y Arduino, ambas se consideran de las más populares, cientos de libros, tutoriales en Internet, proyectos de titulación y proyectos profesionales se han construido con estas placas.

3.2 Raspberry Pi

3.2.1 Breve historia de Raspberry Pi

Raspberry Pi (ver figura 3-1), es un computador de placa reducida, fue ideada en el laboratorio de computadores de la Universidad de Cambridge, Inglaterra. El gestor principal de la idea fue Eben Upton. Él y sus compañeros notaron que los estudiantes de ciencias de la computación más jóvenes, no tenían las mismas capacidades que tenían ellos en la década de los 90 [23].

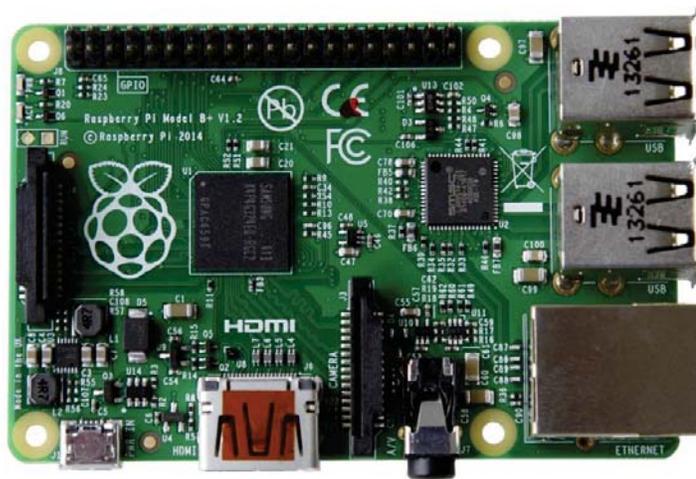


Figura 3-1: Raspberry Pi (fuente: www.raspberrypi.org)

Atribuyeron este fenómeno al hecho de que en la década de los 80, los equipos como Commodore 64, Amiga, BBC Micros y Spectrum ZX eran hechos para ser programados por el usuario final, lo que incentivo el aprendizaje en las ciencias de computadores. Los computadores de escritorio y consolas de juegos modernas, se volvieron un instrumento invaluable en cada hogar, por lo que se restringía el uso para preservar su integridad, de ahí la diferencia en el desempeño de las generaciones [23].

Los procesadores de celulares y tablets se volvieron más poderosos y económicos, lo que permitió construir un computador de placa reducida a un bajo costo, para fomentar el aprendizaje de la ciencia de la computación entre los jóvenes [23].

3.2.2 Aspectos técnicos

Existen 5 modelos de Raspberry Pi que siguen siendo comercializados de forma oficial [24], sin contar los Compute Module, versiones industriales con estructura más compacta. A continuación, se comentará los componentes físicos de una Raspberry Pi (ver figura 3-2). En la tabla 3-1 se realiza una comparación entre algunos modelos de Raspberry Pi.

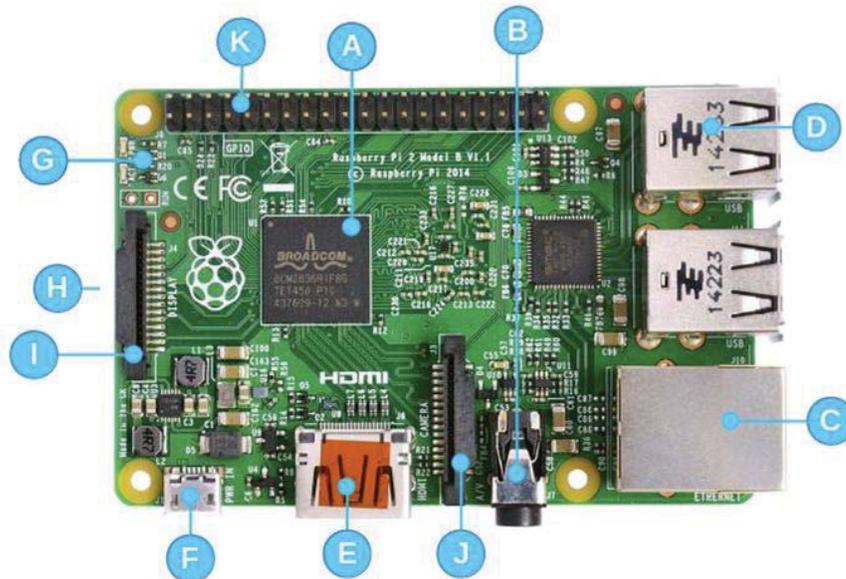


Figura 3-2: Partes Raspberry pi (fuente: elaboración propia)

- A) System on a Chip (SoC): El núcleo de Raspberry Pi está embebido en un SoC, que integra todo el hardware común de un computador en un chip [25], el SoC usado es de la marca Broadcom, vienen con la nomenclatura BCM28xx. Los procesadores de Raspberry Pi son ARM [23], con varias arquitecturas dependiendo del modelo. Todos los periféricos a los cuales accede el procesador están contenidos dentro del SoC [26].
- B) Salida de video y audio: Consiste de un conector jack hembra de 3.5mm con 4 polos, se utiliza con un cable jack de las mismas características con una salida de tres conectores, un par RCA de audio (rojo y blanco) y uno de video NTSC o PAL (amarillo) [23].
- C) Puerto Ethernet: Un puerto Ethernet para conector RJ45, soporta una velocidad de 10 a 100 [Mbps], se conecta directo a la red local de Internet [23].
- D) Puertos USB: 4 Puertos USB 2.0, si se utilizan varios dispositivos USB o de alto consumo eléctrico, se recomienda utilizar un hub USB con su propia fuente de energía [23].
- E) Puerto HDMI: Conector HDMI (High-Definition Multimedia Interface), es la conexión utilizada por los monitores modernos de alta definición [25].

F) Puerto Micro USB: Conector Micro USB por el cual se energiza la Raspberry Pi, el voltaje de entrada debe ser de 5[V], la corriente de consumo varía según la utilización de la placa y los periféricos conectados, documentación oficial indica que una fuente de 2,5[A] cubre prácticamente todos los requerimientos de energía de Raspberry Pi [26].

G) LED indicadores: Dos luces led (PWR rojo y ACT verde) que indican el estado de la Raspberry Pi, led PWR prendido indica que hay energía, el led ACT prendido indica que se accedió a la tarjeta SD [23].

H) Puerto micro SD: puerto para tarjeta micro SD, Raspberry Pi utiliza estas tarjetas como disco duro, almacenando toda la información del sistema operativo y la creada por el usuario.

I) Display Serial Interface: Conector de cable plano de 15 pines, para pantallas normales y táctiles.

J) Camera Serial Interface: Conector de cable plano, para un módulo de cámara oficial de la fundación Raspberry Pi.

K) Pines GPIO (General Pin Input Output): Pines por los cuales el SoC interactúa con electrónica externa, a través de señales digitales. Como indica el nombre pueden ser salidas o entradas digitales, la documentación oficial [26] indica que los voltajes a los que se conecta cualquier GPIO debe ser entre 3,3[V] y 0[V], de otro modo se puede dañar el SoC, esto deja inutilizable la Raspberry Pi. Las señales de los puertos GPIO son leídas o escritas desde los códigos escritos por el usuario.

Buses de comunicación soportados por Raspberry Pi, se incluye el número del pin:

- Inter-Integrated Circuit I2C:
 - 3 SDA Datos
 - 5 SCL Reloj
- Serial Peripheral Interface SPI:
 - 19 MOSI salida maestro, entrada esclavo
 - 21 MISO entrada maestro, salida esclavo
 - 23 SCLK reloj serial
 - 24 CE0 habilitar canal 0, selector de esclavo SS
 - 26 CE1 habilitar canal 1, selector de esclavo SS

- Universal Asynchronous Receiver/Transmitter UART:
 - 8 TX transmisor
 - 10 RX receptor

Actualmente hay 5 modelos de Raspberry Pi, siendo la versión 3B la última en salir al mercado, la tabla 3-1 hace una comparación entre las versiones, información obtenida desde [27].

Tabla 3-1: Tabla comparativa de características Raspberry Pi

Modelo	Zero	A+	2 B+	3B
SOC	BCM2835	BCM2835	BCM2836	BCM2837
CPU	ARM11	ARM11v6	ARM11v7	ARMv8
	1[GHz]	700[MHz]	4 núcleos 900[MHz]	4 núcleos 1.2[GHz]
RAM	512[MB]	256 [MB]	1[GB]	1 [GB]
	LPDDR2	LPDDR	LPDDR2	LPDDR2
	SDRAM	SDRAM	SDRAM	SDRAM
USB	2 micro USB	1	4	4
Vídeo	Mini HDMI	HDMI 1.4	HDMI 1.4	HDMI 1.4
Ethernet	No	Si	Si	Si
Wi-Fi	No	No	No	Si
Bluetooth	No	No	No	Si

3.2.3 Utilización

Raspberry Pi, como cualquier otro computador, necesita de un sistema operativo para funcionar. El disco duro de Raspberry Pi es una tarjeta microSD, en esta tarjeta se instalará el sistema operativo [28].

Raspberry Pi soporta sistemas operativos basados en Linux, al ser una placa basada en un procesador de dispositivos móviles (celulares), el tipo de sistema operativo no es igual al de un PC. Los sistemas operativos para Raspberry están optimizados para su arquitectura y capacidades [29].

Una vez instalado el sistema operativo, se deben conectar tarjeta, pantalla, mouse, teclado y fuente de energía a la Raspberry. En la guía [28] se entregan las instrucciones para el primer encendido de Raspberry Pi, luego de varias configuraciones se llegara al ambiente clásico de escritorio (ver figura 3-3).



Figura 3-3: Escritorio Raspberry Pi (fuente: www.raspberrypi.org)

Desde el escritorio se puede comenzar a utilizar Raspberry Pi como cualquier computador, instalar programas o escribir códigos en los varios lenguajes soportados por la placa (Python, C, C++, Java, etc.), siendo Python el lenguaje nativo. Para ejecutar programas, se utiliza la *Terminal*, una interfaz gráfica simple (sólo texto), que permite navegar por las carpetas del sistema y ejecutar funciones del sistema operativo y códigos escritos por el usuario.

Raspberry Pi puede ser utilizada de forma remota por medio de SSH (Secure Shell), sólo necesita una conexión a internet, por lo que se pueden desconectar todos los otros periféricos excepto la energía. El ambiente de trabajo es el mismo de una ventana de terminal.

3.3 Arduino

3.3.1 Breve historia de Arduino

Esta placa de desarrollo fue inventada en el 2005, por Massimo Banzi en el Interaction Design Institute Ivrea IDII, en Ivrea Italia [30]. Fue inventado, con la idea de tener una tarjeta para programar y utilizar micro-controladores de bajo costo, de fácil uso, para fomentar la creación de proyectos en electrónica.

Banzi había sido contratado en el IDII, para promocionar nuevas formas de diseño interactivo (computación física). En el 2002 se utilizaba una placa llamada BASIC Stamp, una placa de circuito impreso con las partes esenciales para el uso de un micro-controlador PIC, la placa tenía dos problemas: no tenía suficiente capacidad de procesamiento y el precio era muy alto US\$100.

Un amigo de Banzi en el MIT (Massachusetts Institute of Technology), había creado un lenguaje de programación llamado Processing, destacado por la facilidad de programar usando su ambiente de programación IDE (Integrate Development Environment). Banzi, pensó en implementar un IDE para la programación de micro-controladores. Un IDE fue creado por un estudiante de Banzi, Hernando Barragán, una plataforma llamada Wiring que incluía una placa

lista para programar y usar. Este desarrollo en conjunto con el uso de micro-controladores de bajo costo Atmel, fueron los primeros pasos en la creación de Arduino [30].

Años después se llegaría al desarrollo de la placa que se conoce actualmente. La comercialización de Arduino no es exclusiva de la agrupación creadora, está basado en el modelo Open-Source, lo que permite que se puedan crear Arduino en varias partes del mundo, sólo se necesita la aprobación del producto por parte de la fundación Arduino, para poder ser vendido como Arduino.

3.3.2 Aspectos técnicos

Existen 54 modelos oficiales de Arduino [31], sumado a los varios modelos realizados por empresas externas. Arduino es una plataforma para usar procesadores de marca Atmel, es una tarjeta de circuito impreso, con los componentes para poder programar y utilizar todas las funciones del micro-controlador, de forma rápida, sin tener que perder tiempo cableando un circuito. A continuación, se enumeran las partes de una placa modelo Arduino UNO, índice de las partes en la figura 3-4.

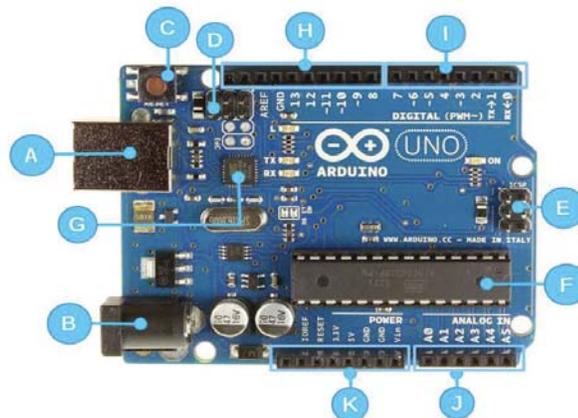


Figura 3-4: Arduino Uno (fuente: www.arduino.cc)

A) Puerto USB B: Conexión USB para comunicación con PC, se utiliza para programar el microcontrolador desde el Arduino IDE, también para el intercambio de información bidireccional entre el PC y Arduino, la conexión por cable USB también da energía a la placa.

B) Entrada de voltaje: Conector Jack 2,1[mm] x 5[mm] con polarización de centro positivo. Es un entrada para voltaje continuo que da energía a la placa, el rango de trabajo está entre los +7[V] a +12[V]. Se permite la conexión simultánea con el puerto USB.

C) Botón Reset: Botón para reiniciar el funcionamiento de Arduino, los programas que ejecuta el microcontrolador se ejecutan de forma lineal. Al pulsar este botón la placa comienza a ejecutar el programa desde el inicio.

D) ICSP Comunicación USB: In Circuit Serial Programming, programación serial en circuito para modificar la funcionalidad del controlador USB de la placa.

E) ICSP Microcontrolador: Puerto para programar el bootloader del micro-controlador de forma directa, sin pasar por el IDE de Arduino.

F) Microcontrolador: El micro-controlador es la parte principal de Arduino, es donde se almacenan y ejecutan los programas cargados desde IDE, todas las capacidades que se asocian a esta placa de desarrollo vienen del funcionamiento del micro-controlador. El modelo del micro-controlador varía según la versión de la placa.

G) Controlador USB: Es un micro-controlador que gestiona la comunicación USB entre el micro-controlador y el puerto USB del PC.

H) y I) Pines Digitales: Pines para el uso de señales digitales, cada pin reconoce dos posibles estados (alto o bajo) como entrada o salida, se utilizan para recibir señales de sensores (entradas) o para accionar o comunicarse con otros dispositivos electrónicos (salidas). Existen funciones alternativas para algunos pines de este conjunto, que serán detalladas más adelante.

J) Pines de entrada análogas: Pines para leer señales análogas, el rango de trabajo de estos pines está entre los 0[V] y 5[V]. A diferencia de los pines digitales que reconocen dos estados, los pines análogos reconocen 1024 valores de voltajes.

K) Pines de voltajes: Estos pines entregan salidas de voltaje y conexiones a tierra, dos pines dan conexión a la tierra de Arduino marcados como GND, otros dos pines entregan 5[V] estos son IOREF y el que está marcado como 5[V], otro pin entrega 3.3[V], finalmente un pin da acceso a la señal de reset.

A continuación, se muestra en la tabla 3-2, un resumen de las características de Arduino UNO Buses de comunicación soportados por Arduino, se incluye el número del pin:

- Inter-Integrated Circuit I2C:
 - A4 SDA Datos
 - A5 SCL Reloj

- Serial Peripheral Interface SPI:
 - 11 MOSI salida maestro, entrada esclavo
 - 12 MISO entrada maestro, salida esclavo
 - 13 SCLK reloj serial
 - 10 SS selector de esclavo

- Universal Asynchronous Receiver/Transmitter UART:
 - 1 TX transmisor
 - 0 RX receptor

Tabla 3-2: Características Atmega328P Arduino UNO (fuente:www.atmel.com)

Característica	Valor
Microcontrolador	ATmega328P
Voltaje de operación	5[V]
Voltajes de alimentación recomendados	7[V] a 12[V]
Pines de entradas y salidas digitales	23
Pines para PWM (3, 5, 6, 9, 10, 11)	6
Pines de entradas análogas	8
Resolución ADC	10[bit]
Corriente por pin	20[mA]
Memoria Flash	32[KB]
RAM	2[KB]
EEPROM	1[KB]
Reloj(XTAL)	16[MHz]

3.3.3 Utilización

Para programar la placa Arduino es necesario utilizar el IDE de Arduino (ver figura 3-5), el IDE es un ambiente de programación, que permite escribir programas desde el PC donde se instala el IDE y cargar los programas a la placa, mediante la conexión USB. IDE es soportado por Mac, Windows y Linux en conjunto con los respectivos controladores para las placas Arduino, también existe una versión online del IDE.

```

Blink | Arduino 1.0.1
File Edit Sketch Tools Help
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeats.
 * This example code is in the public domain.
 */
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage)
  delay(1000); // wait for a second
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Done uploading.
Binary sketch size: 1,072 bytes (of a 30,720 byte maximum)
done with autoreset
13 Sleepy Pi on /dev/ttyS0

```

Figura 3-5: Arduino IDE (fuente: www.arduino.cc)

El lenguaje de programación usado se llama Arduino, un lenguaje de alto nivel, esto quiere decir que se asemeja más al lenguaje humano que al que entiende una máquina. Cuando un código escrito en Arduino es cargado a la placa, pasa por el compilador AVR-GCC que traduce el código a lenguaje C, un lenguaje de bajo nivel entendido por el micro-controlador Atmel [32].

La estructura de un código de Arduino es muy parecida a la de uno escrito en C, se parte por la declaración de variables y luego se puede generar funciones escribiéndolas en bloques, delimitados por los paréntesis {}.

El ejemplo código en el listado 3-1, suma de forma repetitiva un 1 a una variable llamada a, el primer bloque del ejemplo es el setup, esta parte del código se ejecuta una sola vez al iniciar el funcionamiento de la placa, en este bloque se declaran las variables que usará el programa. El bloque siguiente es el loop, este bloque se ejecutará después del setup, el código se irá ejecutando línea por línea hasta el final del bloque, para volver a la primera línea y repetir el proceso hasta que la placa se apague o se presione el botón de reset.

Listado 3-1: Código ejemplo Arduino

```
1 void setup()
2 {
3     int a=0;
4 }
5 void loop()
6 {
7     a=a+1;
8 }
```

Todo lenguaje de programación tiene un grupo de funciones básicas, como funciones aritméticas, manejo de texto, manejo de número binarios, etc. Para el uso de funciones más complicadas, se utilizan librerías, que agrupan conjuntos de códigos con funcionalidades complejas. Arduino IDE viene con un grupo de librerías de uso común, también permite agregar o crear librerías según las necesidades de cada proyecto.

Todos los materiales necesarios para comenzar a utilizar Arduino son la placa, una conexión USB y el Arduino IDE. Cargando el programa en la placa, éste va a ejecutarse de forma repetitiva mientras la placa está energizada.

3.4 Shields Arduino

Como se había comentado anteriormente, Las placas de Arduino tienen funcionalidades limitadas a los tópicos básicos de la electrónica. Los Shields son dispositivos compatibles con Arduino, la disposición de pines del Shield es la misma que tienen los Arduinos, se conectan sobre el Arduino y usando los pines del mismo para comunicarse y energizarse (ver figura 3-6). Por medio de la comunicación con el microcontrolador Atmel, la placa Arduino puede usar todas las funcionalidades del Shield. La comunicación varía según la complejidad del Shield, puede ser desde los pines I2C, SPI o por medio de señales digitales en los pines de entrada y salida digital.



Figura 3-6: Shield GPS (fuente: www.sparkfun.com)

Hay una gran variedad de Shields para Arduino, en [33] se registran aproximadamente 317 diseños de Shields. Las funcionalidades de los Shields pueden ser desde conexión Ethernet hasta controladores de motores. Como los Shields de Arduino son alimentados por medio de los pines de energía, todos funcionan con 5[V].

3.5 Problemática de la compatibilidad entre Raspberry Pi y Arduino

Habiendo comentado las características principales de Arduino y Raspberry Pi, se puede asociar cada placa a ciertos tipos de proyectos electrónicos, esta asociación viene definida por las fortalezas y debilidades de cada placa, que serán listadas en la tabla 3-3.

Tabla 3-3: Comparativa entre Raspberry Pi y Arduino

	Arduino	Raspberry Pi
Voltaje de trabajo	5[V]	3.3[V]
Corriente de salida por pin	20[mA]	16[mA]
Velocidad de procesamiento	16[MHz]	1.2[GHz]
Memoria programable	>1[GB]	32[KB]
Memoria para variables volátiles	1[GB]	2[KB]
Precio promedio (CLP)	\$7990 (Arduino NANO)	\$42.990 (V 2B+)
Convertidor Análogo/Digital	5 canales	No

Arduino tiene como fortalezas su precio y la compatibilidad con electrónica externa, debido al voltaje de trabajo y la cantidad de corriente que puede dar por pin, también la capacidad de leer señales analógicas. Esto se traduce en la gran cantidad de Shields, sensores y actuadores compatibles. Otra característica importante es la facilidad de implementación, para programar Arduino sólo se necesita cargar el programa y la placa ya está lista para ser utilizada.

Las debilidades de Arduino son su baja velocidad de procesamiento, poca memoria para almacenar programas y variables, es porque se asocia normalmente a proyectos más simples, en donde la necesidad principal es medir alguna variable y ejecutar una acción mediante un actuador.

Raspberry Pi tiene dos importantes fortalezas, la velocidad procesamiento y la capacidad de almacenar información. Esto implica que puede ser utilizado para proyectos con una gran carga de procesamiento de datos, proyectos con páginas web locales en Raspberry Pi, proyectos con bases de datos en Raspberry Pi.

Las debilidades de Raspberry Pi es la poca compatibilidad con electrónica externa, excluyendo los periféricos comunes de un computador, los dispositivos electrónicos compatibles con Raspberry Pi son escasos, la mayoría de la electrónica disponible para sensores usa voltaje TTL de 0[V] a 5[V]. Por eso Raspberry Pi está normalmente asociada a proyectos en donde cumple el rol de visualizar los datos, almacenarlos o procesarlos, pero no extraerlos ni medirlos.

Las debilidades y fortalezas de ambas placas de desarrollo son complementarias. Si se juntan las fortalezas de ambas, por una parte tendría todo el poder de procesamiento de Raspberry Pi, con un conjunto de sensores y actuadores capaces de interactuar con el medio, gracias a los Shields y sensores compatibles con Arduino.

Hay problemas al intentar la compatibilidad de estas dos tecnologías, el principal problema son los rangos de voltaje de trabajo, Raspberry Pi utiliza voltajes CMOS (0[V]-3.3[V]), Arduino y sus Shields utilizan voltaje TTL(0[V]-5[V]). Si se conectara un sensor que entregue una señal de 5[V] a un pin de Raspberry Pi, ésta se quemaría. Por otro lado, si se pretendiese energizar un Shield de Arduino con 3.3[V], puede dañarse o simplemente no funcione, no logre captar bien las señales digitales enviadas por Raspberry Pi. Otro problema es la falta de un convertidor análogo digital en Raspberry Pi, varios Shields y sensores de Arduino funcionan con señales análogas.

En este proyecto se quiere dar solución a la problemática descrita previamente, la compatibilidad de Raspberry Pi con Arduino, para crear una herramienta que pueda aprovechar las mejores características de ambas placas de desarrollo.

3.6 Medio de comunicación de la red

El medio de comunicación va a definir en gran parte la estructura de la red, la propuesta para esta red de sensores es una comunicación híbrida, mezclando la comunicación cableada e inalámbrica. Para la comunicación cableada se usará la comunicación Ethernet, para la comunicación inalámbrica se utilizará la comunicación Wi-Fi, los estándares de comunicación elegidos, fueron seleccionados por la facilidad de implementación, también porque son de las tecnologías más usadas en las redes de área local (Local Area Network LAN), las placas de desarrollo son totalmente compatibles con estos medios de comunicación.

3.6.1 Comunicación Cableada

Ethernet es un estándar de comunicación, se puede aplicar a varios medios físicos de comunicación, cable coaxial, cable UTP o fibra óptica, funcionando a distintas velocidades desde 10[Mbps] hasta 10[Gbps], el grupo de estándar que define las de aplicaciones de Ethernet son las IEEE802.3 [34].

Un protocolo es un conjunto de reglas aplicadas a un proceso, como la comunicación entre dispositivos electrónicos, por cada evolución en la ejecución del proceso, el conjunto de reglas cambia. Para explicar las bases de Ethernet, se definirán los elementos básicos de una comunicación Ethernet.

El protocolo Ethernet define la forma en la que se transmite la información entre dispositivos en la red (estaciones), el formato más básico de información es un bit, por lo que Ethernet define la forma de transmitir y recibir esa información, con tal de obtener una comunicación eficiente.

La agrupación de bits transmitidos se denomina trama, las tramas se conforman de varios grupos de bits, destinados a representar información relevante para el protocolo Ethernet, en la figura 3-7 se puede observar la estructura básica de una trama de Ethernet.

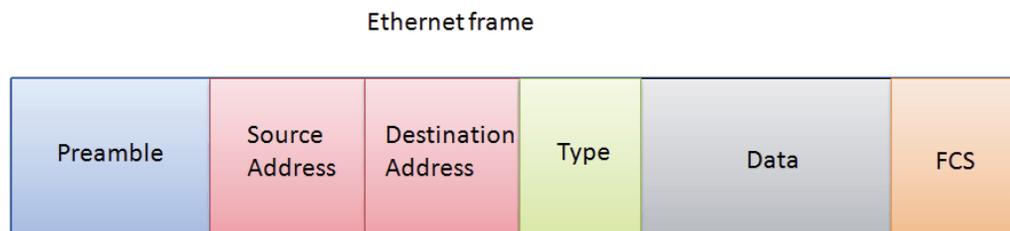


Figura 3-7: Trama Ethernet (fuente: www.lostintransist.se)

Con la trama definida se puede tener un formato de los datos transmitidos, este formato entrega información para que las estaciones cumplan las reglas del protocolo, las características y funciones de cada segmento de la trama, están definidas en la tabla 3-4.

Tabla 3-4: Contenido de la trama Ethernet

Segmento de la trama	Explicación
Preamble	Es un grupo de 68[bits], que informa a la estación de llegada que se prepare para recibir una trama [35].
Source Address y Destination Address	Dirección MAC (Media Access Control), un número identificador de 48[bits]. Son las direcciones de los dispositivos de origen y destino de la trama respectivamente [35].
Type	Es un grupo de 16[bits], que entregan información acerca del grupo de bits siguiente, el largo y el tipo de protocolo de alto nivel, se aplica sobre los bits del siguiente bloque [35].
Data	Es una agrupación de 46[bits] a 1500[bits], En Data esta la información “útil” de la trama. Esta agrupación de bits está sometida a un protocolo de mayor nivel que Ethernet, como por ejemplo TCP/IP o UDP [35].
FCS (Frame Check Sequence)	Información acerca de la integridad de la trama, por un análisis de CRC (Cyclic Redundancy Checksum) [35].

Las reglas definidas para la coordinación de dispositivos Ethernet, están definidas en el control de acceso al medio MAC (Media Access Control), un subconjunto de reglas definidas dentro del protocolo Ethernet. Las reglas de MAC varían según la aplicación de Ethernet Half o Full Duplex o las velocidades de comunicación.

Para las aplicaciones más antiguas, la red era un canal compartido por todas las estaciones, por lo que todos recibían los mensajes enviados. En este tipo de aplicación las reglas del MAC aseguran la igualdad de oportunidades del uso de la red para todas las estaciones.

En un canal compartido es necesario identificar el destino de cada trama, para que las estaciones no procesen todos los mensajes recibidos, haciendo una comunicación más eficiente. Ya se había comentado la existencia de las direcciones MAC en las tramas Ethernet, cuando una estación recibe una trama se lee primero el preámbulo y luego la dirección de destino, si la dirección leída concuerda con la de la estación, entonces el mensaje es procesado completamente, en caso contrario es descartado.

Al ser el canal de comunicación compartido, las estaciones deben esperar a detectar disponibilidad del canal para enviar una trama (Carrier Sense), es normal que varias estaciones detecten esa oportunidad simultáneamente, lo que provoca que se envíen tramas al mismo tiempo, distorsionando las tramas recibidas, a esto se le llama colisión (Collision).

Para trabajar con los defectos del canal compartido, se implementó el protocolo CSMA/CD (Carrier Sense Multiple Access Collision Detect). Las primeras dos siglas (Carrier Sense) indican la detección de la disponibilidad del canal para enviar una trama, las siguientes dos siglas indican el hecho de que varias estaciones utilizan el canal (Multiple Access). Finalmente se implementó una forma de detectar una colisión (Collision Detect), cuando se detecta una colisión los dispositivos esperan un tiempo aleatorio (rango de microsegundos), para volver a enviar las tramas, asegurando así una posibilidad reducida de una nueva colisión (algoritmo Backoff) [35].

Las aplicaciones Ethernet más modernas se rigen bajo nuevos protocolos, como en el caso de Full-Duplex Ethernet (sin CSMA/CD) o en velocidades de transmisión más alta, pero los principios comentados sirven para entender las bases de la comunicación Ethernet.

3.6.2 Comunicación Inalámbrica

Wi-Fi es un estándar de comunicación inalámbrica, el grupo de normas que rigen la aplicación de Wi-Fi es el IEEE802.11, la banda de frecuencia de trabajo es 2,4[Ghz] (5[GHz] para IEEE802.11a), una banda no licenciada de uso libre [36], Wi-Fi es el modo de comunicación inalámbrica más usado para WLAN (LAN inalámbrico).

La comunicación Wi-Fi puede ser bajo distintas normas de las IEEE802.11, con distintas velocidades y formas de transmisión, pero, utilizando el principio del Spread Spectrum (espectro ensanchado), es una familia de métodos que utilizan varias frecuencias para la transmisión de la señal, usando las disponibles en la banda de frecuencia.

Direct- Sequence Spread Spectrum (DSSS), método de transmisión usado en la IEEE802.11b, en el DSSS se utilizan varios canales fijos de la banda de frecuencias para enviar una señal. Cada bit de la señal original es reemplazado por una secuencia (secuencia de Baker), esto provoca que la señal original en frecuencia sea esparcida en una banda de frecuencia (22[MHz] de ancho). Cada link DSSS usa un solo canal de frecuencia, una señal DSSS usa menos potencia que una señal normal [37]. El receptor de la señal DSSS debe tener la misma secuencia Baker, totalmente coordinado con el receptor para decodificar la señal recibida y transformar la secuencia obtenida en el bit correspondiente. Teóricamente se pueden alcanzar velocidades de hasta 11[Mbps], si la calidad de señal (QoS) no es buena, el receptor y transmisor usa un proceso llamado Dynamic Rate Shifting para ajustar la velocidad de transmisión hasta mejorar la QoS.

Orthogonal Frequency Division Multiplexing (OFDM), método usado en las IEEE802.11a/g/n [37], es un proceso más complejo que DSSS, en OFDM el flujo de bits es separado y transmitido en forma paralela por varios canales de frecuencia dentro de la banda de 2,4[GHz], aumentando la

tasa de transmisión del enlace [38]. Se usan 52 canales de los cuales 4 entregan información piloto de los otros 48, para reducir pérdidas.

La norma IEEE802.11 a especifica 8 canales de un ancho de 20[MHz], cada canal está dividido en 52 subcanales de 300[KHz] [37]. Cuando se detecta una señal IEEE802.11 a en la recepción, se ensamblan los bits paralelos en un flujo de bits de alta velocidad y se utiliza la información piloto para comprobar la integridad de los datos. Igual que DSSS el OFDM detecta la QoS y puede reajustar la velocidad de transmisión en caso de necesitarlo. Teóricamente se pueden alcanzar velocidades de hasta 54[Mbps].

Wi-Fi o IEEE802.11 también emplea la utilización de tramas, existen tres tipos de tramas, trama de datos, encargada del transporte de datos de los protocolos de alto nivel. La trama de control encargada del control de las conexiones. La trama de administración encargada de dar la información para el protocolo y seguridad, necesaria para el cumplimiento del MAC [39]. En la figura 3-8, se puede observar la estructura de una trama IEEE802.11, obtenida desde [39].

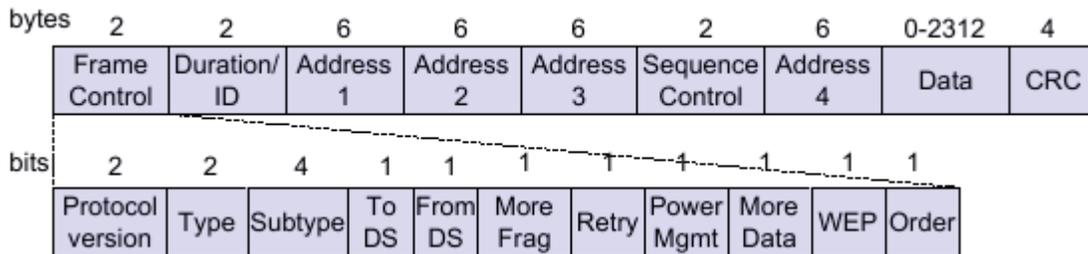


Figura 3-8: Trama IEEE802.11(fuente: www.ecovi.uagro.mx)

En la tabla 3-5, se describe cada uno de los segmentos de la trama Wi-Fi. Cada segmento de la trama, entrega información relevante para el cumplimiento de las normas, que rigen al protocolo de comunicación, con las cuales el estándar pretende asegurar una comunicación eficiente y confiable.

Tabla 3-5: Tabla explicativa de la trama Wi-Fi

Segmento	Descripción
Frame Control	Control de trama, 16[bits] que se dividen en varios campos de información.
Duration/ID	Indica cuánto tiempo durará la transmisión de la trama y su confirmación de recepción.
Address 1-4	Direcciones MAC, cada una de 48 [bits], desde Address1 a Address4, las direcciones son de destino, origen, destinatario inmediato y dispositivo que envió la trama.
Sequence Control	Control de secuencia, indica el número de la secuencia asignado a la trama. Las tramas retransmitidas se identifican con números de secuencia duplicados.
Data	Contiene la carga de datos útiles, encapsulados por protocolos de niveles superiores, el largo máximo es de hasta 2034[bytes].
CRC	Es un grupo de bits con información de la integridad de la trama, mediante un código de redundancia cíclica.

Como se puede observar en la figura 3-8, el segmento Frame Control contiene una subtrama, con varios segmentos, los cuales entregan información referente a las características de la trama Wi-Fi. Los datos de esta subtrama, hacen referencia al tipo de trama, características de su transmisión, características del destino y relevancia. Ésta información es usada por los dispositivos Wi-Fi, para saber cómo interpretar la información contenida en la trama Wi-Fi. La explicación de cada segmento, de la subtrama Frame Control, es explicada en la tabla 3-6.

Tabla 3-6: Subtrama de Frame Control

Segmento	Descripción
Protocol version	Es la versión del protocolo, indica con un numero de 2[bits] la versión de la trama IEEE802.11, actualmente sólo existe una versión y el bit es siempre 0.
Type/subtype	Identifica el tipo de trama con dos bits, puede ser 00 trama de administración, 01 trama de control, 10 trama de datos, 11 reservado.
To DS	Un bit con valor 1 para identificar si la trama va a un elemento de distribución de la red, valor 0 en caso contrario.
From DS:	Un bit valor 1 para identificar si proviene desde un elemento de distribución de la red, 0 en caso contrario.
More Frag:	Más fragmentos, un bit que se establece en uno si las tramas tienen otros fragmentos.
Retry	Reintentar, un bit de valor para identificar si la trama es la retransmisión de una trama anterior.
Power Mgmt	Administración de energía, un bit de valor 1 si el nodo de origen está en modo de ahorro de energía, 0 en caso contrario.
More data	Más datos, un bit con valor 1 para indicar a un nodo en ahorro de energía que hay mensajes pendientes para enviarle.
WEP	Un bit de valor 1 si la trama está encriptada mediante WEP, para propósitos de seguridad.
Order	Orden, un bit que se establece en 1 cuando él envió ordenado de paquetes sea muy importante.

En el protocolo Wi-Fi también existe el MAC (control de acceso al medio), funciona de forma parecida al MAC del protocolo Ethernet. MAC implementa un protocolo llamado CSMA/CA (Carrier Sense Multiple Access Collision Avoidance), para evitar colisiones y conflictos en la red [37].

CSMA/CA implica accesos múltiples al medio de comunicación, cuando se envían mensajes simultáneos ocurren colisiones, cuando sucede, todos menos un nodo paran de enviar información y vuelven a intentarlo después de un tiempo [37].

Para enviar tramas, el nodo transmisor espera un tiempo para detectar tramas enviadas desde otros nodos, en caso de no escuchar, vuelve a detectar tramas por un tiempo corto aleatorio, si no hay tramas enviadas se transmite la trama [37].

El nodo que reciba un mensaje debe evaluar la integridad de la trama, entregando una señal de reconocimiento al nodo transmisor en caso de ser aprobada, con el CRC, si el nodo transmisor no recibe la señal espera un tiempo y vuelve a enviar la señal [37].

Wi-Fi tiene medios de seguridad para la conexión, como el reconocimiento de señal por SSID (nombre de la señal) y dirección MAC identificando dispositivos, para conceder o negar acceso a la red.

Estos son los conceptos básicos para entender cómo funciona Wi-Fi, desde la transmisión de los datos hasta las reglas de convivencia para los dispositivos en la red. Las redes Wi-Fi aseguran una compatibilidad con muchos dispositivos de uso común modernos, como celulares inteligente, computadores, tablets, etc. Esto sumado a la movilidad y dinamismo que propicia el uso de Wi-Fi, fueron las principales razones para implementarla en la red de sensores.

3.7 Esquema de la red

La red de sensores propuesta en este proyecto, utiliza las placas de desarrollo Raspberry Pi 2B+ y Arduino UNO, para formar los nodos que componen la red. Comunicando los nodos a través de los medios de comunicación comentados previamente. En la figura 3-9 se puede observar la estructura de la red de sensores.

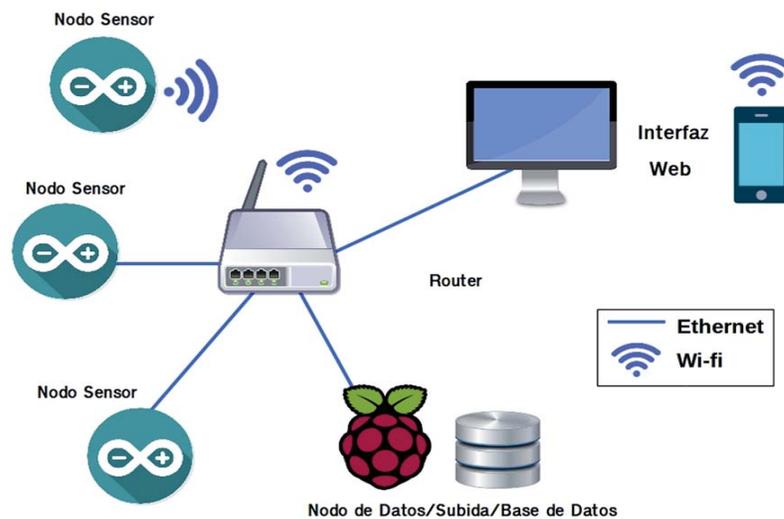


Figura 3-9: Esquema Red de sensores mixta (fuente: elaboración propia)

Los nodos sensores serán desarrollados con placas Arduino, como se vio en el capítulo anterior los nodos sensores cuentan con un sensor o actuador, un controlador y un comunicador. El controlador será la placa Arduino, el sensor o actuador puede ser un Shield Arduino o un sensor o actuador compatible. En cuanto al comunicador, éste va a variar según el tipo de

comunicación del nodo sensor. Hay dos nodos sensores con conexión Ethernet, las placas de esos nodos usarán un módulo de conexión Ethernet ENC28J60. Existe un solo nodo sensor inalámbrico, éste se comunicará por medio de Wi-Fi, el nodo usará un módulo ESP8266 para soportar la comunicación.

Raspberry Pi va a concentrar las funciones de varios elementos de la red, el trabajo del nodo de subida, de datos y base de datos estará siendo realizado por la placa de desarrollo. El nodo de subida se encarga de gestionar la información enviada por los nodos sensores, también de almacenar estos datos en una base de datos y enviarlos a una interfaz de visualización. Raspberry Pi cuenta con la capacidad de procesamiento y la compatibilidad con lenguajes que permiten la programación del control de los datos en la red, también se pueden programar interfaces de visualización. La capacidad de almacenamiento de Raspberry Pi, permite que se pueda trabajar una base de datos local. La conexión de Raspberry Pi es por Ethernet, la placa es totalmente compatible con esta comunicación, ya que tiene todos los controladores y el puerto Ethernet incluidos.

La red tiene una comunicación mixta, para poder unir los dos estándares usados se utilizará un router inalámbrico. Una red Ethernet con varios elementos conectados, en conjunto con una red Wi-Fi con varios elementos conectados, pueden conformar una red mayor comunicándose entre ellas, ese es el objetivo del router, unificar los distintos medios de comunicación.

Finalmente, está la interfaz de comunicación, esta interfaz se construirá en Raspberry Pi, pero será accedida por medio de dispositivos externos a la red de sensores, siempre y cuando puedan acceder a la red por alguno de los dos medios de comunicación, Ethernet o Wi-Fi, cualquier computador, smartphone o tablet, cumple esas especificaciones.

A continuación, se comentará la composición de todos los nodos, tanto en Hardware como en Software. Todos los programas, que serán comentados en los siguientes segmentos del informe, están disponibles en un CD adjunto al informe.

3.8 Arduino como nodo sensor

Como se comentó en la estructura final de la red, se implementarán 3 nodos sensores, de los cuales un nodo se comunica por medio de Wi-Fi y los otros dos por Ethernet. El funcionamiento de los nodos sensores es el mismo para los tres, consiste en la recopilación de medidas con el uso de sensores, adecuar la información a una medida comprensible y finalmente enviar la información al nodo de subida, paralelamente el nodo sensor también tendrá salidas lógicas controladas por el usuario desde la interfaz gráfica, por lo que debe ser capaz de recibir y procesar mensajes provenientes de la red.

3.8.1 Nodo sensor inalámbrico

La placa Arduino UNO no es compatible con la comunicación Wi-Fi, por lo que se debe agregar un comunicador como puente entre la comunicación Wi-Fi y Arduino. El módulo de comunicación ESP8266, ver figura 3-10, es un módulo de bajo costo para agregar

funcionalidades Wi-Fi a cualquier micro-controlador. ESP8266 soporta los estándares IEEE802.11 b/g/n y el protocolo TCP/IP, el módulo puede actuar como punto de acceso Wi-Fi (P2P, Soft-AP) o ser cliente de una red, también puede ser servidor o cliente TCP/IP [40].



Figura 3-10: Modulo ESP8266 (fuente: www.maxelectronica.cl)

ESP8266 se comunica físicamente con micro-controladores externos por comunicación UART (serial), totalmente compatible con Arduino, mediante un cable de transmisión (Tx) y un cable de recepción (Rx) [41]. Para trabajar con el módulo se deben utilizar los comandos AT [40], son un subconjunto de los comandos Hayes, un estándar abierto de comandos para configurar dispositivos de red (modem, router, etc.).

En [42] se indica una advertencia en el uso del ESP8266, necesita una alimentación de 3,3[V] y todas sus señales lógicas se trabajan entre los 0[V]-3,3[V], esto quiere decir que no se puede alimentar desde Arduino y que deben ajustarse las señales del puerto serial (0[V]-5[V]) al rango admitido por el módulo.

Arduino tiene un pin para energizar con 3,3[V], pero no entrega corriente suficiente, el consumo de corriente del ESP8266 durante una transmisión Wi-Fi puede llegar a ser de 215[mA] [43], Para solucionar el problema se agregó un regulador de voltaje variable con capacidad de hasta 3[A], ver figura 3-11, que energizará al ESP8266 con los 3,3[V].

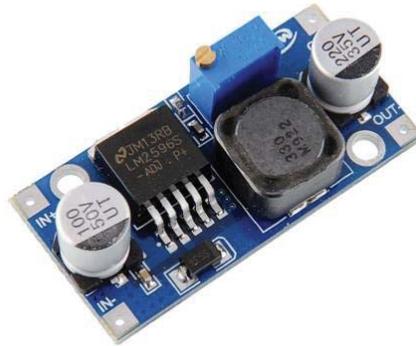


Figura 3-11: Regulador de Voltaje Step-Down (fuente: www.maxelectronica.cl)

El problema de las señales lógicas, se solucionó con un convertidor bidireccional (ver figura 3-12), este convertidor puede pasar una señal lógica desde un rango de valores a otro según las referencias de voltaje que se le entregue, en este caso de 3,3[V] y 5[V]. En la figura 3-13, se puede observar un esquema de conexión para usar el ESP8266 con Arduino.

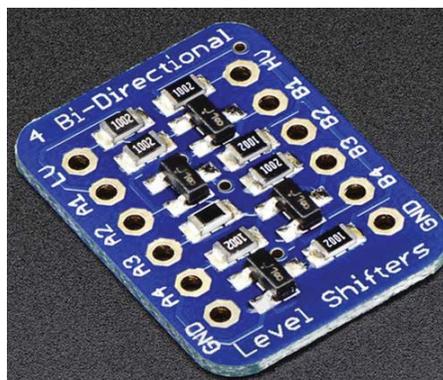


Figura 3-12: Convertidor bidireccional de niveles lógicos (fuente: www.rambal.cl)

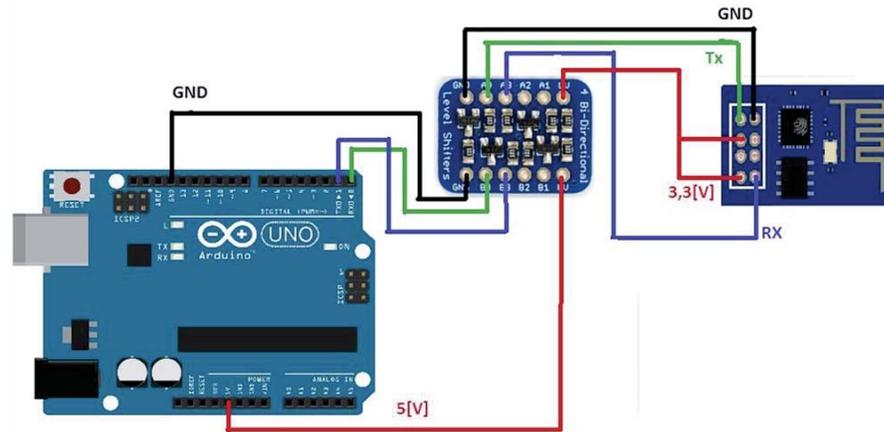


Figura 3-13: Esquema ESP8266 con Arduino (fuente: elaboración propia)

En cuanto a la fuente de energía para el nodo inalámbrico, se utilizaron baterías, la razón principal de esta decisión fue para tener un nodo totalmente sin cables. El arreglo de baterías dispone 9[V] al nodo sensor, Arduino UNO tiene un regulador de voltaje integrado, por lo que no tiene problemas con el voltaje entregado por las baterías.

3.8.2 Software del nodo sensor inalámbrico:

Para lograr la comunicación Wi-Fi del nodo, toda la interacción con el ESP8266 es en términos de comandos AT [41], son palabras que indican instrucciones para el módulo, existen tantos comandos como acciones realizables por un módulo Wi-Fi.

Los comandos AT parten siempre con AT+, después viene un conjunto de letras que representa el comando a ejecutar, por ejemplo AT+RST (reset), los símbolos siguientes definen si el comando AT realiza acciones de prueba, configuración, ejecución o consulta, un ejemplo de la estructura de un comando AT en la tabla 3-7.

Tabla 3-7: Ejemplos comandos AT

Prueba	AT+CWMODE=?	Pregunta los posibles modos AP del módulo. (1= cliente red, 2=Punto de acceso, 3= mixto)
Consulta	AT+CWMODE?	Pregunta el modo AP actual del módulo
Configuración	AT+CWMODE=1	Establece uno de los modos AP para el módulo.
Ejecución	AT+RST	Reinicia el módulo.

La forma de trabajo de los comandos AT, incluye una respuesta del ESP8266 al Arduino, frente a cualquier acción realizada por un comando AT enviado desde Arduino, esta respuesta varía

según el tipo de comando ejecutado (ver tabla 3-8) y el resultado del comando en la configuración del módulo.

Tabla 3-8: Tabla de respuestas comandos AT

Prueba	AT+CMD=?	+CMD:(1-3) OK	Muestra los posibles parámetros para usar el comando. Muestra los parámetros actuales del módulo, referentes al comando.
Consulta	AT+CMD?	+CMD:(parámetros) OK	Confirmación de que se ejecutó la configuración que indica el comando
Configuración	AT+CMD=(parámetro)	OK	Confirma que se ejecutó la instrucción del comando.
Ejecución	AT+CMD	OK	

En caso de que el comando falle por distintas razones, la respuesta del módulo es ERROR. Estas son las consideraciones que se tuvieron presente a la hora de escribir el código que ejecuta Arduino, para cumplir las funciones de nodo sensor inalámbrico.

El programa Arduino del nodo sensor inalámbrico, está incluido en el CD como NODOWI-FI2, a continuación se procederá a explicar los aspectos más importantes del código. La figura 3-14, muestra un diagrama con procesos, que se programaron en el código Arduino del nodo sensor inalámbrico, cada bloque será comentado y relacionado con un bloque del programa.

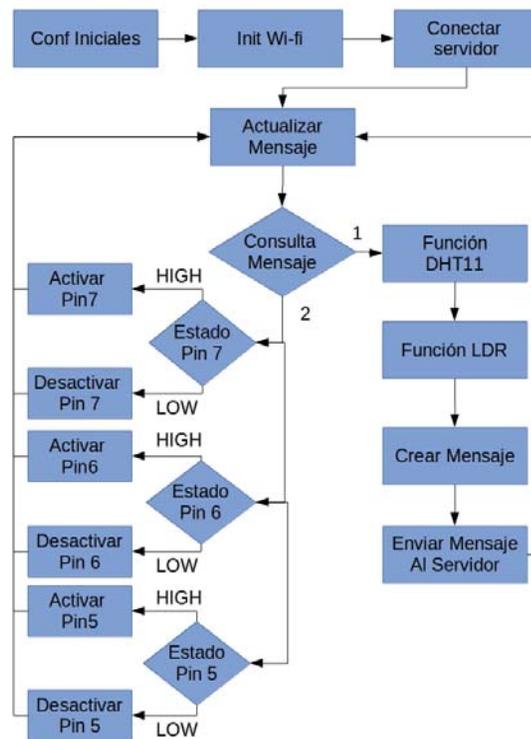


Figura 3-14: Diagrama programa nodo Wi-Fi (fuente: elaboración propia)

El cuadro Conf Iniciales, hace referencia a las primeras líneas del código (1-36), en donde se incluyen las librerías que contienen las funciones del código, estas funciones se usan para control de comunicación con sensores y el módulo inalámbrico, conversiones de tipos variables y manejo de líneas de texto (strings). En el cuadro Conf Iniciales también se configuran las variables, los pines de Arduino y los comandos AT usados en el programa.

Init Wi-Fi representa todo el proceso de configuración del ESP8266, enviando los comandos AT configurados previamente, por el puerto serial designado. Cada mensaje enviado configura el módulo ESP8266, para que se conecte a la señal Wi-Fi del router de la red de sensores, el listado 3-2 muestra los comandos enviados en el arreglo órdenes.

Listado 3-2: Código Arduino comandos AT

```

22 String ordenes[]=
23 {
24     "ATE0\r\n", // ATE0 deshabilitar ECO de comandos AT
25     "AT+CWMODE=1\r\n", // habilitar modo estación (CLIENTE WI-FI)
26     "AT+CIPMUX=1\r\n", // permitir multiples conexiones
27     "AT+CIPSTAMAC=\"1a:fe:34:00:c6:d5\"\r\n", // direccion mac
28     "AT+CWJAP=\"Riquelme\", \"1235813213\"\r\n", //conectarse a la red Wi-Fi
29     // "AT+CIPSTART=1,\"TCP\", \"192.168.0.37\", 9988\r\n",
30     "FINAL"
31 };
  
```

Por cada mensaje enviado se debe esperar por la confirmación del módulo (OK\r\n), se programó una función, expuesta en el listado 3-3, que toma un argumento y espera la respuesta del ESP8266, comparando lo recibido con el argumento, retornando el mensaje del módulo si calza con el argumento, el tiempo máximo de espera es de 11[seg], que fue el tiempo de demora para la respuesta más tardía observada.

Listado 3-3: Función Esperar respuesta

```
1 String Esp_Respuesta(String indicador)
2 {
3   msj="";
4   starttime=millis();
5   while(millis()-starttime <= timeoff)
6   {
7     if(ESP.available() > 0)
8     {
9       e = ESP.read();
10      msj += e;
11  if(msj.indexOf(indicador)>=0){delay(1000);return msj;}
12  }
13  }
14  return msj;
15 }
```

Conectar servidor hace referencia a la función en donde se envían comandos AT al módulo, para conectarse con el servidor de mensajes en Raspberry Pi, detallando la dirección IP y el puerto a cual conectarse (ver listado 3-4). En este caso se utiliza una función muy parecida a la del listado 3-3, pero la variable de retorno es un booleano indicando si el argumento coincide con el mensaje recibido desde el ESP8266.

Listado 3-4: Función de conexión con el servidor

```
1 void Rpi_server()
2 {
3   delay(1000);
4   ESP.print("AT+CIPSTART=1,\"TCP\", \"192.168.1.143\", 8899\r\n");
5   boolean conn=Esperar_por_respuesta("OK\r\n");
6   if(conn){Serial.print("Conectado al servidor en 192.168.0.18 puerto 8899");};
7   return;
8 }
```

Actualizar mensaje es una función que recibe los datos enviados por el ESP8266 (ver listado 3-5), estos datos provienen de la comunicación con la Raspberry Pi, para los datos de comunicación de la red se creó un formato de mensaje, para evitar que Arduino malinterprete un mensaje o reciba información errónea. Cada mensaje de la red debe partir con un símbolo # y finalizar con un símbolo !, existen dos tipos de mensaje, uno para solicitar las lecturas de los sensores a los nodos sensores, el segundo tipo es para ejecutar control sobre los pines de Arduino, ambos tipos de mensaje se mostraran en la tabla 3-9.

Listado 3-5: Código Recepción de mensaje

```

1 boolean actualizar_mensaje(){
2   if (ESP.available())
3     { c = ESP.read() ;
4       if(c=='#'){mesread=true;}
5       if (mesread){message[strlen(message)] = c;} //Concatenamiento mensaje.
6     }
7   if(c=='!'){
8     mesread=false;
9     return true;
10  }
11
12  return false;
13 }

```

Tabla 3-9: Formato de Mensajes

#OE!	Solicitud de lecturas sensores Instrucción para controlar pines desde la interfaz.
#P0/1/0!	#P(valor pin7)/(valor pin6)/(valor pin5)! 0 salida 0[V] 1 salida 5[V]

En el cuadro consulta mensaje se revisa la primera letra del mensaje recibido, para saber si se debe enviar los datos o sólo cambiar el estado de los pines controlados, en caso de ser una solicitud de los datos, se procede a usar los sensores y extraer la información del ambiente (cuadros Función DHT11 y Función LDR). Finalmente, los datos obtenidos vienen en variables de tipo float e int, estos son transformados a string y anexados al tipo de mensaje de salida del nodo sensor, en el cuadro Crear mensaje, el formato del mensaje de salida es mostrado en la tabla 3-10.

Tabla 3-10: Formato mensaje de control de pines

#valor1/valor2/valor3!	Valor 1, valor2 y valor3 son los resultados de la medición de los sensores, convertidos en string, el símbolo / sirve para separar la información.
------------------------	--

El cuadro final es enviar mensaje, hace referencia a los procesos para enviar mensajes al ESP8266 por comandos AT, una función del programa gestiona la construcción de los comandos AT necesarios (ver listado 3-6), la función toma como argumento el mensaje de salida del nodo sensor.

Listado 3-6: Función envío de mensajes

```

1 void send_msj(String outmsj)
2 {
3   String sendcomm="AT+CIPSEND=1,";
4   String largo= String(outmsj.length());
5   largo += "\r\n";
6   sendcomm += largo;
7   ESP.print(sendcomm);
8   boolean arg=Esperar_por_respuesta(">");
9
10  if (arg){ESP.print(outmsj);}
11
12  if(Esperar_por_respuesta("SEND OK\r\n"))
13  {Serial.println("Mensaje Enviado!");};
14
15  return;
16 }

```

La lógica para enviar mensaje es construir y enviar primero el comando “AT+CIPSEND=n,m”, en donde n es el número del canal de comunicación, en el caso del programa de Arduino es 1, m es el número de caracteres del mensaje, ese valor es agregado en la línea 4 del listado 3-6. Al enviar el comando completo se debe esperar que el ESP8266 mande el carácter “>”, en ese momento se envía el mensaje de salida, cuando el mensaje sea enviado exitosamente, el módulo enviará una nueva respuesta “SEND OK”, el código esperará esa respuesta y luego retornará al loop principal para actualizar el mensaje de entrada y repetir el proceso desde ese punto.

En caso de que el mensaje recibido sea una instrucción para controlar los pines, entonces se procede a revisar el mensaje, según el valor de los espacios indicados en la tabla 3-9, se escriben las salidas digitales de los pines controlados, luego de ejecutar estas acciones se vuelve al loop principal para actualizar el mensaje de entrada y repetir el proceso desde ese punto.

Para concluir la explicación del programa del nodo sensor, se reitera que el programa se ejecuta de forma repetitiva, desde el punto de recepción de los mensajes de la red, esto permite al nodo sensor comunicarse constantemente, cambiando los procesos a ejecutar según la instrucción del mensaje de llegada, lo que puede tomarse una forma de comunicación sincrónica, ya que se espera la solicitud para enviar los datos a la red, este tipo de coordinación genera una comunicación con baja tasa de errores o pérdida de información.

3.8.3 Nodo sensor cableado

La comunicación Ethernet, puede ser usada por Arduino mediante la utilización de un módulo Ethernet ENC28J60 (ver figura 3-15), permite conectar Arduino a internet por medio de cable de red, es una alternativa menos costosa que el Shield Ethernet de Arduino. El módulo se basa en el controlador Ethernet ENC28J60 [44], cumple todas las especificaciones de la norma IEEE 802.3, se comunica con micro-controladores externos mediante un bus SPI (Serial Peripheral Interface). El ENC28J60 se alimenta con 3,3[V] y soporta 5[V] en los pines de comunicación, lo que significa que parcialmente compatible con Arduino, sólo necesita un regulador de voltaje como el del nodo sensor inalámbrico.

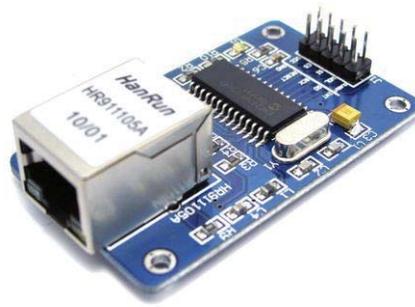


Figura 3-15: Modulo Ethernet ENC28J60 (fuente: www.internetdelascosas.cl)

Existen varias librerías que permiten a Arduino utilizar este módulo, se extrajeron y probaron tres librerías que presentaron diferencias en la implementación, tanto por la capacidad de la librería para soportar códigos más complejos, como en la implementación física. A continuación, se presenta la indicación de conexión pin a pin entre el módulo Ethernet y Arduino Uno (ver tabla 3-11), según la librería utilizada.

Tabla 3-11: Conexión del ENC28J60 según librería

Pines Modulo ENC28J60	Pines Arduino		
	ETHER_28J60	Ethercard	UIPEthernet
CS	10	8	10
MOSI(SI)	11	11	11
MISO(SO)	12	12	12
SCK	13	13	13

En los primeros intentos de implementación, se llegó a la conclusión de que la mejor librería a utilizar es la UIPEthernet.h, ya que presenta funciones más útiles para el propósito del proyecto. La librería presentó una falla, en la función que detecta el estado de la conexión `client.connect()`, lo que impide reconocer cuando se rompe la comunicación. Se encontró una solución al problema, esta solución será expuesta en el segmento de explicación del proyecto.

Con respecto a la energía del nodo sensor se utilizó un transformador AC/DC, el transformador se conecta a un enchufe de pared y entrega una salida de 5[V] y 1[A], Esta fuente de energía se eligió porque al ser un nodo cableado, su posición será fija, lo que implica que su fuente de energía no necesita ser móvil. Además los transformadores AC/DC son una fuente de energía económica, fácilmente reemplazable y estable en la señal de salida.

3.8.4 Software nodo sensor cableado

Los programas de los nodos sensores Ethernet se encuentran en el CD adjunto como NODO1 y NODO2, tienen las mismas facetas que el programa del nodo sensor Wi-Fi, la figura 3-16 muestra un diagrama que simboliza los procesos que ejecuta el código del nodo sensor Ethernet, cada bloque será explicado a continuación.

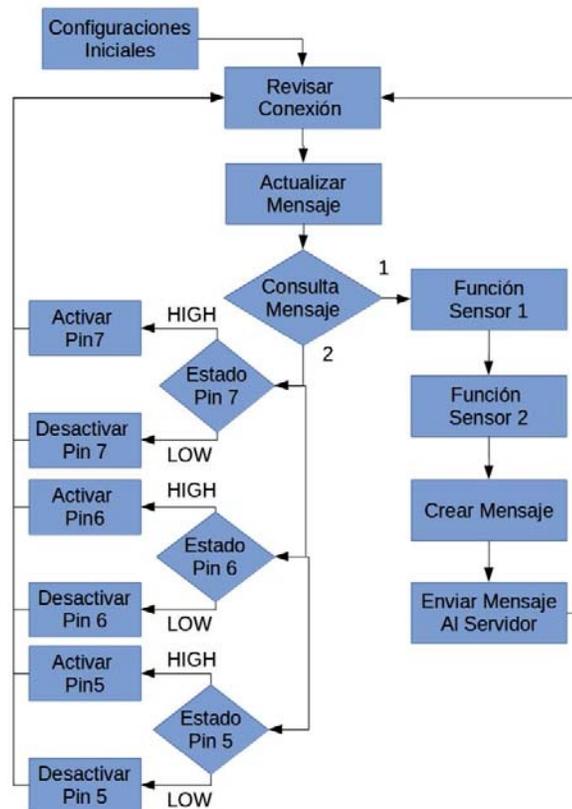


Figura 3-16: Esquema Código nodo sensor cableado (fuente: elaboración propia)

El bloque de configuraciones iniciales hace referencia al primer proceso del código, es la importación de librerías, configuración de las variables, estructuras, pines y funciones que utilizará el programa. Todo este proceso se entiende como el grupo de líneas de código previos a los bloques de funciones. El inicio de la comunicación con el ENC28J60 ocurre en el bloque `setup()` del programa en el listado 3-7, este proceso se incluye en el cuadro de configuraciones iniciales.

Listado 3-7: Configuraciones iniciales

```

117 void setup() {
118     pinMode(7, OUTPUT);
119     pinMode(6, OUTPUT);
120     pinMode(5, OUTPUT);
121     Serial.begin(9600);
122     Serial.println("serial-OK");
123     Ethernet.begin(mac);
124     delay(500);
125     Serial.print(Ethernet.localIP());
126     if(client.connect(servidor,puerto)){Serial.print("conectado!");}
127     else{client.connect(servidor,puerto);}
128 }

```

Las funciones que inician la comunicación con el ENC28J60 son provistas por la librería UIPEthernet.h, la primera se encuentra en la línea 123 del listado 3-7, puede observarse que se entrega una dirección MAC para asignarse al módulo. La conexión con el servidor en Raspberry Pi se realiza en la línea 126, especificando la dirección IP del servidor y el puerto que utiliza el programa servidor, en la línea 127 se vuelve a conectar en caso de que la primera conexión falle.

El bloque revisar conexión es un proceso para detectar si el módulo se desconectó de la red, esto es posible a través de una variable booleana (conOK), actualizada en cada iteración, que es verdadera cuando llega un mensaje al nodo y falsa en el caso contrario, función mostrada en el listado 3-8.

Listado 3-8: Función recepción de mensajes

```

29 void Recibir(){
30     if((size = client.available()) > 0)
31     {
32         conOK=true;
33         uint8_t* msg = (uint8_t*)malloc(size);
34         size = client.read(msg, size);
35         Serial.write(msg, size);Serial.println(msg[1]);
36         reconocer_mensaje(msg);
37         free(msg);
38         String outdato=funcion_medicion_dht();
39         String outdato_suelo=funcion_medicion_humedad_suelo();
40         String mensaje_final= '#' + outdato
41         + '/' + outdato_suelo + '!';
42         client.print(mensaje_final);
43     }
44     else{
45         if(conOK){
46             conOK=false;
47             offtime=millis();}
48     }

```

Otra función, en el listado 3-9, revisa el tiempo en el que la variable lleva siendo falsa, si supera los 3 segundos el programa asume una desconexión, comenzando un proceso de reconexión.

Listado 3-9: Loop principal función de conexión

```
130 void loop() {
131   if(!conOK&&(millis()-offtime >= 3000L)){
132     client.stop();
133     delay(500);
134     if(client.connect(servidor,puerto)){
135       Serial.print("reconectado!");
136       conOK=true;}
137   }
138   Recibir();
139   delay(500);
140 }
```

El bloque de actualizar mensaje, hace referencia a la recepción de los mensajes desde la red, en el listado 3-8, líneas 32 y 33, se ve la función que obtiene los mensajes desde el ENC28J60. Desde la línea 35 del listado 3-8, se llama la función que reconoce el tipo de mensaje, haciendo el trabajo representado por el cuadro de consulta mensaje.

Los procesos ejecutados después de recibir un mensaje, son esencialmente iguales a los procesos del mismo punto en el programa mostrado anteriormente, si el mensaje es una solicitud de datos de sensores, entonces se realizan las mediciones, se transforman a string y se acomodan al formato de mensaje de salida, la función para enviar el mensaje por la comunicación Ethernet se puede ver en la línea 41 del listado 3-8. Se indica en la figura 3-16, que después de enviar el mensaje se vuelve a la función de revisar función y todos los procesos se repiten desde ese punto.

En caso de que el mensaje recibido sea una instrucción para manejo de pines, se ejecuta el mismo proceso comentado en el código de nodo sensor Wi-Fi, se analizan los valores del mensaje, para asignar las salidas correspondientes a los pines controlados, luego de esto se vuelve al punto de revisar conexión para ejecutar los procesos siguientes nuevamente.

Los procesos explicados en esta sección hacen referencia a los códigos de ambos nodos sensores Ethernet, por lo que no se especificó el tipo de sensor, los códigos de ambos nodos presentan diferencias sólo en los aspectos del uso de sensores.

3.9 Raspberry Pi como nodo de subida

Las características más útiles de Raspberry Pi para el desarrollo de una red de sensores, son su capacidad de procesamiento y almacenamiento, también considerar que tiene un puerto Ethernet incluido, lo que asegura completa compatibilidad con el diseño propuesto para la red de sensores.

Una guía completa para tener operativa una Raspberry Pi se encuentra en [28], aquí se explica que debe instalarse en una tarjeta micro SD un sistema operativo de los disponibles en [29], esto mediante un lector de tarjetas SD y un programa para cargarle la imagen del sistema operativo

(Win32DiskImager para Windows). Cuando esta lista la tarjeta SD, se debe insertar en la Raspberry Pi, conectando una fuente de energía y los periféricos de mouse (USB) teclado (USB) y pantalla (HDMI) podemos proceder con la instalación descrita en [28].

La función del nodo de subida es almacenar la información capturada por la red de sensores, en una base de datos local. Según una lista detallada de tarjetas SD compatibles con Raspberry Pi en [45], se puede utilizar tarjetas SD de hasta 64[GB], quitando el espacio de memoria utilizado por el sistema operativo, la memoria utilizable son 62[GB] aproximadamente, lo que da mucho espacio para una gran base de datos local.

Raspberry Pi necesita usar pantalla, teclado y mouse para usarse por primera vez, todos estos dispositivos pueden ser descartados si se utiliza un acceso remoto por Ethernet a Raspberry Pi, utilizando una conexión por cable de red y otro computador en la misma LAN. Se instaló un programa llamado Samba [46], es un conjunto de aplicaciones para Linux, que implementan el protocolo de comunicación SMB utilizado por los sistemas operativos de Windows, para compartir carpetas e impresoras [47], con esto se puede programar fuera de Raspberry Pi para luego guardar el programa en la placa.

3.9.1 Comunicación con la red de sensores

La comunicación con la red de sensores implica la comunicación de las placas de desarrollo por medio del cable Ethernet y Wi-Fi, en donde Raspberry Pi pueda recibir los datos y almacenarlos, para subir la información a una base de datos y entregar esa información a un usuario.

Para lograr la comunicación de Raspberry Pi con los tres nodos sensores, se escribieron tres códigos Python, la elección del lenguaje es debido a que es nativo en sistemas operativos basado en Linux, los códigos utilizan una librería que permite usar una interfaz de comunicación llamada Socket [48], por la cual se puede recibir y enviar mensajes bajo el protocolo TCP/IP, Socket y el protocolo TCP/IP son compatibles con las librerías utilizadas por Arduino en los nodos sensores.

El protocolo TCP/IP (Transmission Control Protocol/Internet Protocol), consta de dos protocolos, uno es el protocolo IP, permite el enrutamiento de los datos en una red, direcciones IP son asignadas a cada elemento de la red, tomando en cuenta la estructura de la red y las rutas para la información.

TCP es un protocolo orientado a la conexión [49], pretende asegurar la integridad de la comunicación, se intercambia información para configurar una buena comunicación. El protocolo TCP también busca una seguridad en la transmisión de los datos, ya que implementa un sistema de reconocimiento de recepción de mensajes, confirmando la recepción íntegra del mensaje, en caso contrario se genera un reenvío de datos en caso de pérdidas de información.

Hubo intención de escribir un programa que actué como servidor de la red, pero resultó ser complejo el manejo simultáneo de la comunicación de los nodos, los procesos de detección de desconexión no siempre eran oportunos, lo que causaba que el programa dejase de funcionar si se utilizaba un canal caído. Con motivos de simplificar el funcionamiento del programa, se creó

un programa por cada nodo sensor, pudiendo controlar de forma más segura la comunicación, los programas sólo se diferencian en la dirección IP del nodo sensor respectivo, el puerto usado para el canal y la tabla de datos utilizada para guardar la información.

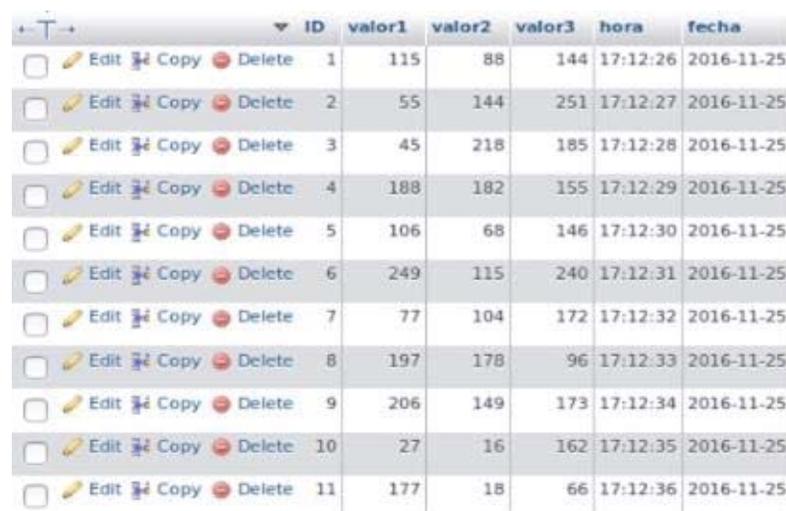
3.9.2 Construcción de la base de datos

Para construir la base de datos de la red, se utilizó un sistema de gestión de base de datos relacional, llamado MySQL, el sistema trabaja con SQL (Structured Query Language) un lenguaje de consultas [50]. Existen diversas interfaces de usuario para el sistema, línea de comando, páginas web, GUI, interfaces de programación, disponibles en varios lenguajes, como PHP, Python, C y Java [51].

Se utilizó una interfaz web llamada Phpmyadmin, para administrar la base de datos MySQL, de gran utilidad para trabajar con las bases de datos por medio de un navegador web de forma más cómoda, MySQL y Phpmyadmin son instalados en Raspberry Pi mediante terminal de comandos [52].

La base de datos se construyó con la interfaz Phpmyadmin, una base de datos consta de varias tablas, las cuales almacenan los datos en una estructura de filas y columnas [53], las columnas agrupan los distintos tipos de datos. Las filas se conforman con los datos alineados por tipo, la base de datos de la red de sensores se compone de 6 tablas, 3 tablas para los datos de los nodos sensores y 3 tablas para almacenar información del control de pines de los nodos sensores.

La estructura de las tablas de datos de nodos sensores es exhibida en la figura 3-17, la primera columna es un numero identificador del registro, el registro son los datos en una fila de la tabla, las columnas valor 1 a 3 son registros de los nodos sensores, la columna hora muestra la hora de la red en que se realizó la medición, la columna fecha muestra el día mes y año en que se realizó la medición. Los datos de las columnas que no son mediciones, permiten identificar cada dato y entregar un registro más preciso, hay una tabla de este tipo por cada nodo sensor.



	ID	valor1	valor2	valor3	hora	fecha
<input type="checkbox"/> Edit Copy Delete	1	115	88	144	17:12:26	2016-11-25
<input type="checkbox"/> Edit Copy Delete	2	55	144	251	17:12:27	2016-11-25
<input type="checkbox"/> Edit Copy Delete	3	45	218	185	17:12:28	2016-11-25
<input type="checkbox"/> Edit Copy Delete	4	188	182	155	17:12:29	2016-11-25
<input type="checkbox"/> Edit Copy Delete	5	106	68	146	17:12:30	2016-11-25
<input type="checkbox"/> Edit Copy Delete	6	249	115	240	17:12:31	2016-11-25
<input type="checkbox"/> Edit Copy Delete	7	77	104	172	17:12:32	2016-11-25
<input type="checkbox"/> Edit Copy Delete	8	197	178	96	17:12:33	2016-11-25
<input type="checkbox"/> Edit Copy Delete	9	206	149	173	17:12:34	2016-11-25
<input type="checkbox"/> Edit Copy Delete	10	27	16	162	17:12:35	2016-11-25
<input type="checkbox"/> Edit Copy Delete	11	177	18	66	17:12:36	2016-11-25

Figura 3-17: Tabla de datos sensores (fuente: interfaz phpmyadmin)

La estructura de las tablas de control de pines es exhibida en la figura 3-18, la tabla sólo tiene una fila, el registro nuevo reemplaza en la fila al anterior, las columnas indican el valor asignado a cada pin controlado por el usuario en la interfaz, la columna ID permite identificar el dato, es útil al acceder a la base de datos desde una interfaz de programación, existe una tabla de este tipo por cada nodo sensor.



ID	estadopin7	estadopin6	estadopin5
1	1	1	1

Figura 3-18: Tabla valores pines (fuente: interfaz phpmyadmin)

Se utilizó la interfaz de programación en lenguaje Python, es una librería llamada MySQLdb, incluyendo esta librería en un código Python, se pueden utilizar funciones para acceder, leer y escribir en las tablas, en general administrar la base de datos desde el código. En los programas escritos para la comunicación de la red, se incluyó una sección de subida a la base de datos, permitiendo almacenar la información recibida y transmitida por Raspberry Pi de forma inmediata.

Otra interfaz de programación usada fue para PHP, este lenguaje se utilizó durante la programación de la interfaz web, en varios de los códigos del apéndice C, PHP permite acceder a las tablas de la base de datos, para escribir, leer o administrar la base de datos desde la interfaz web, por interacción con un usuario.

3.9.3 Programa Nodo de subida

Existen tres códigos que coordinan la comunicación entre cada nodo sensor con el nodo de datos, el mismo código sube la información recibida a una base de datos local, también obtiene los valores de las tablas de control de pines. Cada uno de estos programas se diferencia en el puerto utilizado para comunicarse, la tabla de datos y la tabla de pines del nodo sensor respectivo.

Los programas desarrollados se encuentran en el CD adjunto, con los nombre servidornodo1.py, para explicar el funcionamiento de los códigos se comentarán los procesos que sigue uno de los programas, representado en la figura 3-19. En la explicación, se asociará cada bloque con una función o proceso de los programas del CD.

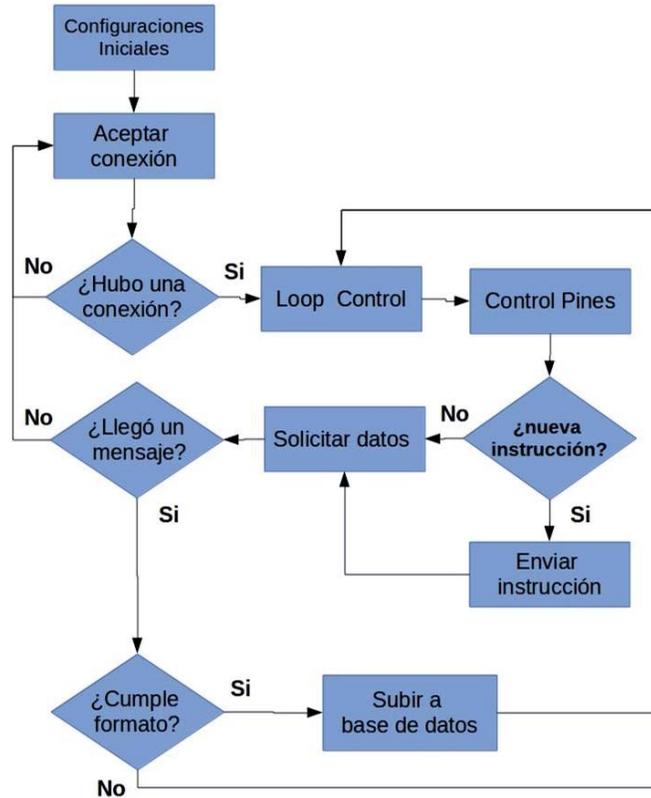


Figura 3-19: Diagrama funcionamiento del programa servidor (fuente: elaboración propia)

El cuadro de configuraciones iniciales encapsula los procesos de importar las librerías del programa, para el uso de las interfaces Socket y Mysql, funciones de sistema para detección de errores, funciones de fecha y hora, después se crea y configura el objeto servidor Socket, finalmente se configuran los valores iniciales de los pines controlados por la interfaz web.

El cuadro aceptar conexión y el cuadro de decisión siguiente, hace referencia a un loop en donde se espera una conexión entrante (línea 140, listado 3-10), esta es la conexión del nodo sensor a la red, cuando el nodo sensor se conecta al servidor del programa, se llama una función llamada loopcontrol(línea 131, listado 3-10), esta función toma como argumento la conexión con el nodo sensor (conn), es un loop en donde se ejecutan las funciones de envío del mensaje de instrucción de control de pines, envío de la solicitud de datos y recepción de los datos, de forma repetitiva hasta que no se detecte respuesta del nodo sensor.

Listado 3-10: Loop central código servidor

```

127 def enviar(conn,mensaje):
128     nsend= conn.send(mensaje)
129     return
130
131 def loopcontrol(conn):
132     while 1:
133         control_pines(conn)
134         enviar(conn,"OE")
135         if not recibir(conn):
136             break
137
138 #now keep talking with the client
139 try:
140     while 1:
141         conn, addr = s.accept()
142         conn.settimeout(6)
143         loopcontrol(conn)
144         s.close()
145
146 except KeyboardInterrupt:
147     s.close()
148
149 except:
150     s.close()
151     restart_program

```

Una vez dentro del loop control se pasa al bloque de control de pines, en conjunto con el bloque de decisión que le sigue y el cuadro de envío de instrucción, representan la función control_pines, presentada en el listado 3-11. En esta función se accede a la base de datos extrayendo la información de la tabla de pines, se analiza si es que los valores de la tabla han cambiado, sólo si se detecta un cambio se manda la instrucción al nodo sensor con el formato descrito en el segmento anterior y los datos actualizados.

Listado 3-11: Código función control pines

```

45 def control_pines(conn):
46     db = MySQLdb.connect("localhost","root","dagol313","sensores" )
47     cursor = db.cursor()
48     sql = "SELECT `estadopin7`, `estadopin6`, `estadopin5` FROM `NEWPINES`"
49     try:
50         global pines_anterior
51         cursor.execute(sql)
52         db.commit()
53         print "acceso a tabla NEWPINES OK"
54         pines=cursor.fetchall()
55
56
57         if pines[0]!=pines_anterior[0] or pines[1]!=pines_anterior[1] or
58 pines[2]!=pines_anterior[2]:
59             pines_anterior=pines
60
61         mensaje_pines='#'+str(pines[0])+'/'+str(pines[1])+'/'+str(pines[2])+'!'
62         enviar(conn,mensaje_pines)
63     else:
64         pass
65     except:
66         db.rollback()
67         return
68     db.close()
69     return

```

El bloque solicitar datos hace referencia al envío del mensaje de solicitud de datos al nodo sensor, la función para enviar el mensaje se llama en la línea 134, se define en la línea 127 a 129 del listado 3-10.

El cuadro de decisión que consulta por la recepción de un mensaje hace referencia a la línea 135 del listado 3-10, en donde se obtiene el retorno de la función recibir (), presentada próximamente en el listado 3-12, la variable de retorno de esta función es un booleano que es verdadero si se detecta un mensaje entrante y falso en caso contrario. Si no se detecta un mensaje durante 6 segundos se asume un error en la conexión por lo que se vuelve al punto de aceptar conexiones.

Listado 3-12: Recepción de mensajes

```
109 def recibir(conn):
110     conOK=False
111     try:
112         data = conn.recv(512)
113         if data:
114             print data
115             conOK=True
116             msgOK, mensaje= check_mensaje(data)
117             if msgOK:
118                 subir_a_Database(mensaje)
119
120     except socket.error as msg:
121         print 'Desconexion: '+ str(msg[0])
122         conOK=False
123         conn.close()
124
125     return conOK
```

El cuadro de decisión, que consulta por el formato del mensaje recibido, representa la línea 116 del listado 3-12, se llama una función llamada check_mensaje(), mostrada en el listado 3-13, la función analiza si el mensaje entrante cumple con el formato de mensaje de datos, el retorno de esta función son una variable booleana que confirma el cumplimiento del formato , también los datos de los sensores sin el formato de entrada.

Listado 3-13: Función para revisar mensajes entrantes

```
73 def check_mensaje(paquete_recibido):
74     mensajevalido=False
75     mensaje_formato=" "
76     if '#' in paquete_recibido:
77         if '!' in paquete_recibido:
78             a=paquete_recibido.index('#')
79             b=paquete_recibido.index('!')
80             mensaje_formato=paquete_recibido[a+1:b]
81             mensajevalido=True
82             print mensaje_formato
83     return mensajevalido, mensaje_formato
```

Como se indica en la figura 3-19, en caso de una confirmación del formato del mensaje, se procede a subir los datos a la base de datos, en la línea 118 del listado 3-12, se llama la función subir_a_Database, expuesta en el listado 3-14, toma como argumento el mensaje con los datos

de los sensores, la función obtiene los tres valores de los sensores, también genera una variable para la fecha y otra para la hora, estos cinco valores son subidos a la base de datos. Una vez subida la información el programa retorna al loopcontrol(), el proceso se repite desde este punto, hasta que se detecte una desconexión.

Listado 3-14: Función de acceso a la base de datos

```
86 def subir_a_Database(mensaje):
87     a=mensaje.find('/')
88     val1=mensaje[:a]
89     b=mensaje.find('/',a+1)
90     val2=mensaje[a+1:b]
91     val3=mensaje[b+1:]
92     fecha= time.strftime('%Y/%m/%d')
93     hora= time.strftime('%H:%M:%S')
94
95     db = MySQLdb.connect("localhost","root","dago1313","sensores" )
96     cursor = db.cursor()
97     sql = "INSERT INTO NEWVALORES (valor1,valor2,valor3,fecha,hora) VALUES
98 (\"" +val1+',' +val2+',' +val3+',' +\"" +fecha+',' +\"" +hora+\"" )"
99     try:
100         cursor.execute(sql)
101         db.commit()
102     except:
103         db.rollback()
104         return
105     db.close()
106     return
```

Con esto concluye la explicación del programa del nodo de subida y base de datos, lo comentado cubre los aspectos básicos del funcionamiento del programa, como resumen se puede decir que el programa está centrado en poder esperar una conexión, detectar el estado de la conexión y coordinar el envío de datos, desde o hacia los nodos sensores para subir la información a la base de datos.

Los programas comentados se comienzan a ejecutar desde el momento que se enciende Raspberry Pi, por medio de un tutorial [54], se encontró la forma de usar estos programas como un servicio invisible que funciona desde el encendido de Raspberry Pi, por lo que no se necesita trabajar sobre la Raspberry Pi una vez terminada la red de sensores.

3.9.4 Visualización de los datos

Parte importante del proyecto implica visualizar y trabajar con los datos obtenidos, es ideal contar con un método que cumpla este objetivo, con el mínimo de requerimientos al usuario final. Bajo los lineamientos descritos previamente, se implementó un sistema de visualización por navegador web, porque es una herramienta de uso común. El trabajo detrás de esta idea implica la implementación de un servidor web, que maneje una base de datos llenada con la información de los nodos sensores.

En Raspberry Pi se puede implementar un servidor LAMP (Linux, Apache, MySQL, PHP), como se describe en [55]. Los componentes del servidor son un conjunto de herramientas que se deben instalar en Raspberry Pi para poder crear un servidor web.

Apache: es un servidor web HTTP (Hypertext Transfer Protocol) Open-Source, un programa especialmente diseñado para transferir datos de hipertexto entre computadores, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Cuando un cliente web (navegador web) accede a un servidor, envía una solicitud HTTP, la cual el servidor responde en formato HTML (HyperText Markup Language), el lenguaje con el que se componen las páginas web.

MySQL: plataforma para la generación de base de datos, consiste en un sistema de cliente y servidor, se compone de un servidor SQL, varios clientes, bibliotecas y una gran cantidad de interfaces de programación.

PHP (Personal Home Pages): lenguaje de programación que forma parte del servidor web para crear webs dinámicas, es un lenguaje diseñado para la programación de páginas web.

PHPmyAdmin: herramienta para administrar la base de datos a través de interfaz web.

Con la implementación del servidor LAMP se busca visualizar la información recopilada por la red de sensores, a través de un navegador web con un computador conectado a la misma LAN de la red de sensores. En la figura 3-20, se puede observar un esquema de funcionamiento de un servidor LAMP.

Desde un computador conectado a la red, se envía una petición HTTP, por ejemplo, ingresar a una página web, el servidor web recibe la solicitud la procesa y ejecuta los programas (HTML, PHP, etc.) de la página web solicitada, estos programas realizan consultas a la base de datos, que guarda información para visualizar o modificar desde la página web, finalmente el servidor web envía una respuesta HTML, que es la página web solicitada con los datos de la base de datos (gráficos, registros, etc.).

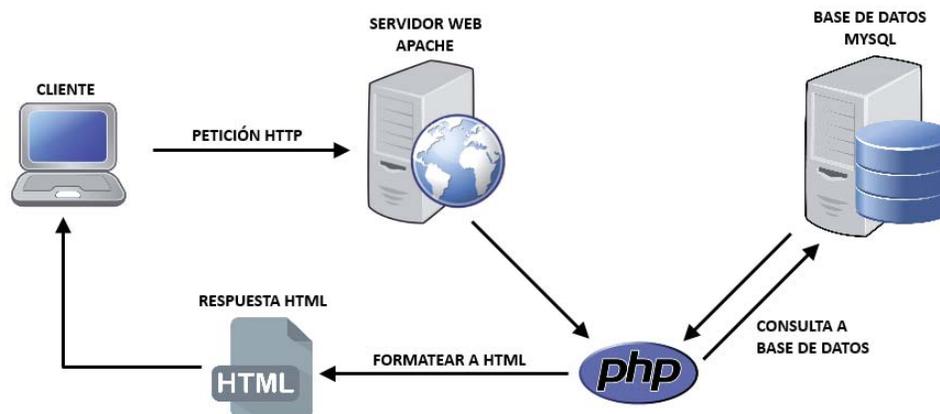


Figura 3-20: Esquema servidor LAMP (fuente: www.diy-makers.es)

3.9.5 Página Web RSM

La Interfaz web de la red de sensores consta de tres páginas web, fue creada con varios lenguajes y meta-lenguajes, cumplen distintos roles, desde el manejo de datos hasta la gráfica y diseño de la página web. A continuación se listan los lenguajes utilizados:

- HTML
- CSS
- PHP
- Javascript

HTML (Hyper Text Markup Language) este lenguaje se usa en todas las paginas, ya que define la estructura de la página web, el navegador web entiende la distribución de los elementos al leer este código. HTML también permite crear elementos de interacción como botones, tablas y cuadros de texto, con gráficas muy básicas.

CSS (Cascading Style Sheets) controla como se verán los distintos elementos de una página en HTML, aspectos como colores, formas, posición y animaciones son definidos en CSS, ahorran mucho trabajo ya que se pueden generar varios “estilos de visualización”, separados por bloques que contienen definiciones de constantes, que hace referencia a configuraciones como color, tamaño, posición, tipo de letra, para aplicarse como el estilo del bloque. Un programa HTML se vincula con CSS en su cabecera (<header>..Cabecera.</header>), en un link con la ubicación del programa CSS, estilizando cada elemento con colores o formas, aplicables indicando en la definición de elemento HTML, el nombre de un bloque del CSS, con esto las definiciones escritas en los bloques, se aplicarán al elemento HTML en la visualización de la página.

PHP (Hypertext Preprocessor): permite trabajar con variables y datos, generados dentro de la página o adquirido de otra página o programa. PHP tiene funciones que permiten trabajar con MySQL, tanto como para leer o escribir en las tablas de la base de datos. Los procesos ejecutados por PHP pueden incluirse en un código HTML, escribiendo un bloque PHP (<?php ...código PHP.... ?>).

Javascript: permite trabajar con elementos para la visualización de datos, también escribir funciones que comparten variables y procesos con otras páginas sin ser visibles, varias herramientas de visualización de gráficos utilizadas usan Javascript. En conjunto con el lenguaje Ajax, se pueden generar cambios en los datos de la página sin la necesidad de recargar la página ni intervención del usuario.

Las tres páginas que conforman la interfaz web, trabajan en conjunto con otros 6 programas y una carpeta de programas CSS, todos estos programas están en la carpeta Interfaz Web del CD adjunto. A continuación se explicará de forma breve los códigos involucrados en la interfaz.

La página principal está programada en el archivo Inicio.html, este código sólo utiliza lenguaje HTML con estilos desde main.css, esta página presenta una breve introducción al uso del

proyecto y se programaron dos botones que dirigen al usuario a las dos páginas de visualización de datos. No se realizará mayor explicación del código, ya que la única funcionalidad práctica de esta página es ser el punto de acceso para el uso de las dos páginas de visualización de datos.

La visualización de registros de datos, se programó para acceder a los datos guardados a lo largo del funcionamiento de la red, el programa de esta página es `Datoshistoricos.php`. El programa consta de tres elementos importantes, un botón de regreso a la página inicial, una tabla para ingresar datos de búsqueda y un gráfico de los datos resultantes de la búsqueda.

Para la tabla de búsqueda, se programó una GUI en JavaScript, que genera un calendario animado para ingresar la fecha de búsqueda, posterior a eso se generan 3 marcadores para elegir el nodo sensor de donde vinieron los datos, también se crearon dos entradas de texto para indicar las horas del día de inicio y termino del registro, finalmente se creó un botón de enviar para cargar la información.

El gráfico de los datos viene desde otra página, que es cargada en una ventana (`iframe`) dentro de la página de registros de datos. Esta otra página se llama `grafica.php`, esta página obtiene los datos de búsqueda desde la página anterior, para realizar la búsqueda en la base de datos MySQL y guardar el resultado en una variable. El gráfico de datos, se logra importando la estructura desde un servidor con material en gráficos y tablas llamado Highchart [56], con esto se puede generar una función que tome los datos desde la variable PHP para graficar la información.

La tercera página es la visualización de datos en tiempo real, esta página se programó en el archivo `Datoslive.php`, la página cuenta con cuatro elementos principales, un botón a la página de inicio, una tabla para indicar el sensor de interés, la ventana para el gráfico y tablas para el control de pines.

La tabla para indicar el nodo sensor de interés son 3 marcadores, los cuales asignan valores a una misma variable para indicar la selección al programa, la selección es también informada a un programa llamado `livedata.php` y a `graph.php` la página del gráfico.

En la ventana para el gráfico se carga la página `graph.php`, el código de esta página se encarga de graficar la información inmediata enviada por el nodo sensor seleccionado, usa una técnica en JavaScript llamada Ajax, para obtener los datos del gráfico desde otro programa, llamado `livedata.php`, en esta página se carga el dato de nodo recibido desde la tabla de datos en tiempo real, para acceder a la tabla correspondiente en la base de datos, luego obtiene la hora y fecha actual para acceder a los datos de la tabla que cumplan con esa fecha y hora, esos datos son enviados nuevamente a la página `graph.php` mediante otra función con el uso de Ajax. Con estos pasos, la página del gráfico va actualizando la información y mostrando un gráfico dinámico con otra estructura sacada desde Highchart.

El último elemento en la página de datos en tiempo real, es la tabla de control de pines, en esta sección se crearon tres ventanas independientes en las cuales se carga una página por ventana, esas páginas son los programas llamados `instruccion.php`, `instruccion2.php` e `instruccion3.php`, Estos programas muestran una tabla para seleccionar los valores de cada pin controlado y un

botón para subir la información. Con esa información cada programa escribe los datos indicados en su tabla de pines correspondiente, que a su vez está asociada a un nodo sensor en el nodo de subida y datos, con esto se puede generar una comunicación entre la interfaz y el nodo sensor, que cambia los valores de sus salidas digitales.

Con esto concluye la breve explicación de la interfaz web, se trabajó bajo un formato de ventanas dentro de la página, para que los procesos de visualización e interacción no significaran un cambio constante de páginas web, en el navegador web del usuario sólo se presenciaban tres páginas, lo que da una sensación de simpleza en el uso. Los códigos CSS no fueron comentados, ya que su implementación se relaciona sólo con aspectos de visualización, los motivos y explicaciones de su configuración sólo son referentes al gusto personal.

4 Resultados

Los resultados del trabajo realizado fueron medidos desde distintas perspectivas, éstas son la estabilidad de la conexión de los elementos de la red, la tasa de transmisión de los datos, la facilidad y estabilidad de la interfaz web, finalmente la velocidad de visualización de los datos en tiempo real.

Con respecto a la estabilidad de la conexión a la red, los nodos sensores que utilizaban la conexión Ethernet presentaban desconexiones espontáneas, en tiempos aleatorios que superaban los 10[*min*], sin embargo, con la función de reconexión los nodos volvían a conectarse a la red para seguir funcionando por otros 10 minutos, con un tiempo entre conexiones de 30[*seg*] en promedio. El nodo sensor Wi-Fi y la Raspberry Pi, presentaron una gran estabilidad en la conexión, sin registros de desconexiones espontáneas como el caso de los nodos Ethernet.

El problema de desconexión, fue investigado en distintos foros, de personas que utilizaron el módulo ENC28J60. La conjetura más común, fue asociar el problema a la fuente de alimentación o a la configuración del router. La solución más simple, fue cambiar la fuente de alimentación de los nodos sensores, por una fuente del mismo voltaje, pero con mayor corriente, el resultado fue una conexión más estable. Otra solución encontrada, fue el uso de un Switch Ethernet, entre el módulo y el router, esta solución fue muy efectiva dando una conexión constante.

La tasa de transmisión de los datos fue evaluada en base a la cantidad de datos guardados en la base de datos por segundo, el conteo se realizó observando la base de datos desde la interfaz web de phpmyadmin, se contaron 2 registros de 3 datos por cada nodo sensor en un segundo, lo que resulta en un total de nueve datos por segundo, se considera una baja tasa de transmisión de información, para una conexión de alta velocidad como Ethernet y Wi-Fi, pero teniendo en cuenta la necesidad de la red de guardar la información, una tasa de transmisión muy alta acabaría con la memoria disponible de Raspberry Pi para la base de datos, también hay que recordar que los programas de los nodos sensores toman un tiempo para procesar las solicitudes y enviar la información a la red.

Una solución, al problema de espacio en la base de datos local, es conectar un disco duro externo, este tipo de disco, es de fácil conexión y tiene mucha memoria disponible, para

contener una mayor cantidad de datos. Otro método posible, es subir la información a un servicio de base de datos online. Existen muchos servicios gratuitos, la empresa Xively [57], entrega este servicio para poder guardar la información en una base de datos online y visualizar los datos guardados, desde cualquier conexión a internet.

Con respecto a la estabilidad de la interfaz, no se registraron casos en que dejase de funcionar, la interfaz demostró una gran estabilidad. La facilidad de uso de la interfaz será descrita, mediante una explicación de los resultados de la programación de las páginas, comentando el uso de las páginas construidas.

Para acceder a la interfaz web de la red de sensores, se debe de estar conectado al router de la red de sensores, por Wi-Fi o Ethernet, ya que el servidor web es local. La interfaz se accede desde un navegador web, la dirección de la interfaz es 192.168.0.18/RedSensores/Inicio.html, los números de la dirección son la dirección IP de la Raspberry Pi en la red.

Es posible, conectarse a la interfaz web, desde una conexión externa a la red. Un método, para realizar lo comentado, es usar un Web Hosting de uso gratuito y crear la misma interfaz web y base de datos, en las dependencias del servicio, el cual permite elegir un nombre para la página, con el cual se puede acceder a la interfaz, desde cualquier conexión a internet.

Otro método, es configurar el router para tener acceso externo a Raspberry Pi y usar un servicio DNS [58]. Este servicio, permite dar un nombre a la página web, que esté vinculado a la dirección IP externa del router, a través de un programa que ellos suministran. Es necesario configurar el router, para direccionar las solicitudes entrantes por el puerto 80 (HTTP) hacia Raspberry Pi, con las instrucciones mostradas en [58]. Siguiendo los pasos comentados, se puede tener acceso a la interfaz web, desde cualquier conexión a internet.

La Interfaz consta de tres páginas, la primera es la página de inicio en donde se muestra información del proyecto, con dos botones que mandan a las otras páginas (ver figura 4-1), el botón de tiempo real redirige a la página de visualización de datos en tiempo real, el botón de registros envía a una página donde se pueden visualizar el registro de los datos por día, hora, minuto y segundo.



Figura 4-1: Página principal del proyecto (fuente: elaboración propia)

La página mostrada en la figura 4-2, es la página de registro, en esta página se pueden ver los datos recopilados por la red de sensores, buscándolos por fecha y hora. En la tabla superior de la página, se ingresa la información de búsqueda, al hacer clic en el cuadro de texto aparece un calendario para buscar la fecha (ver figura 4-3), luego, se selecciona el nodo que envió los datos y finalmente especificar la ventana de tiempo con formato hora:minuto:segundos en ambos cuadro inferiores, luego pulsa el botón enviar. La página va a actualizar el cuadro de gráficos, mostrando los datos recopilados por el nodo sensor seleccionado en la ventana de tiempo especificada.

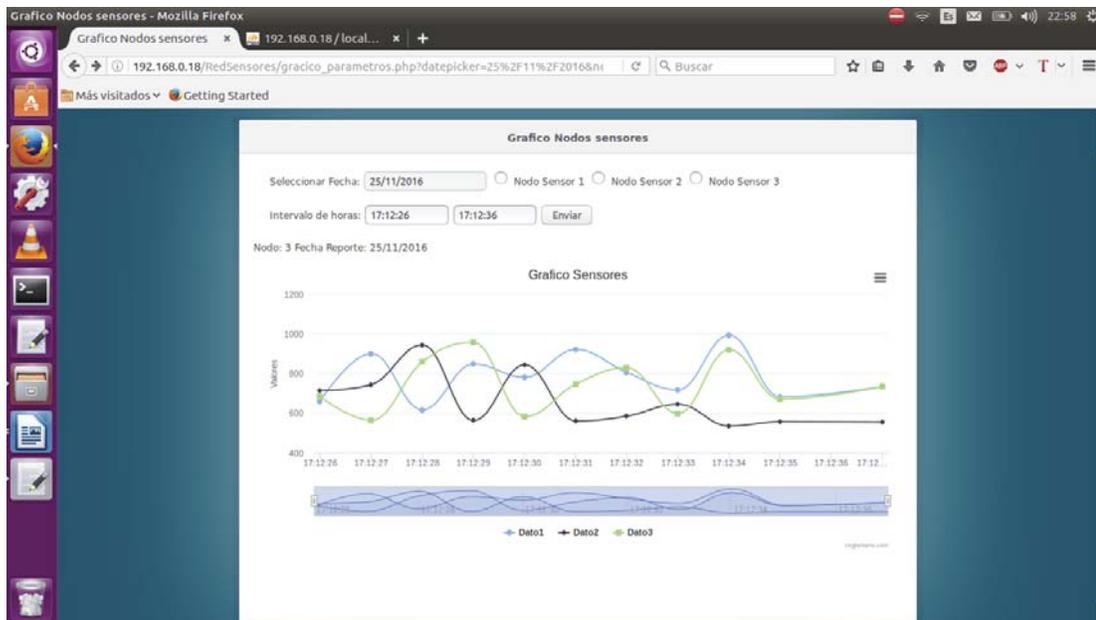


Figura 4-2: Pagina de datos históricos (fuente: elaboración propia)



Figura 4-3: Calendario para selección de fecha (fuente: elaboración propia)

La página de datos en tiempo real tiene dos partes, la primera parte (ver figura 4-4), muestra los datos actualmente registrados por los nodos sensores, en un gráfico animado que va deslizándose a medida que se acumulan datos en la ventana. Para ver la información en tiempo real se debe seleccionar el nodo de interés en la tabla superior de la interfaz y pulsar enviar.

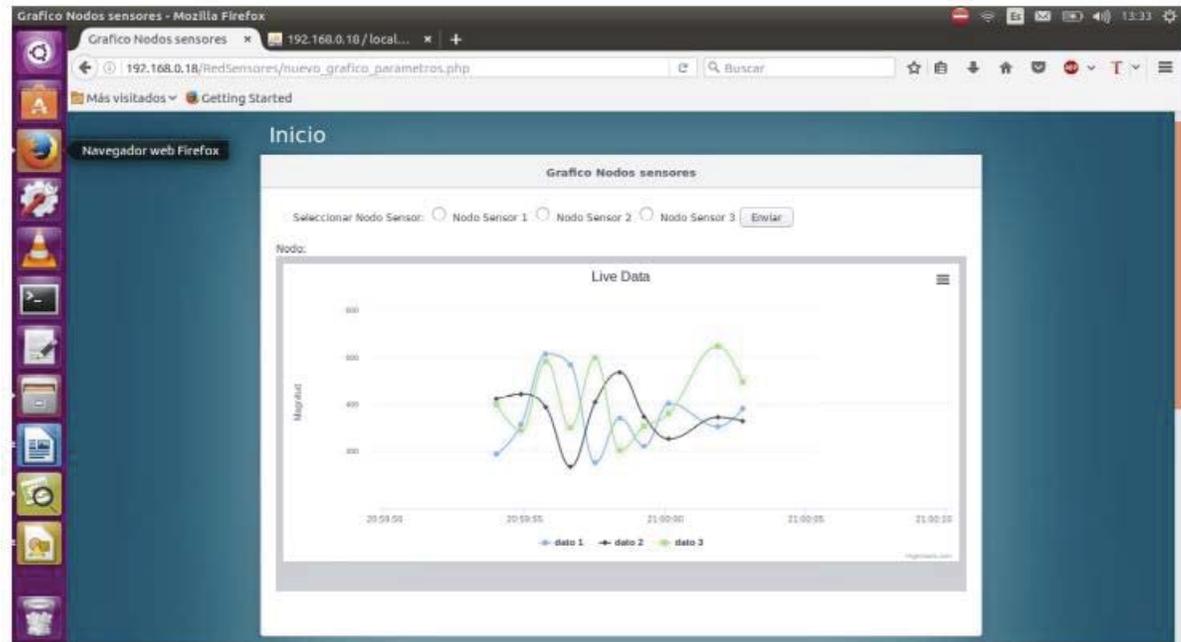


Figura 4-4: Pagina de datos en tiempo real (fuente: elaboración propia)

La segunda parte de la página se muestra en la figura 4-5, son las tablas para el control de pines de los tres nodos sensores, con esta interfaz se pueden activar o desactivar tres salidas digitales desde cada nodo sensor, permitiendo controlar los elementos conectados a esos pines. Las tablas son independientes entre ellas e independiente del gráfico en tiempo real, cada tabla hace referencia a un nodo sensor, se seleccionan los valores para cada pin de la tabla y se ejecuta el cambio al pulsar enviar, se registró un tiempo de retraso desde que se pulsa enviar a cuando se ve el cambio en Arduino de 1 [seg] aproximadamente.

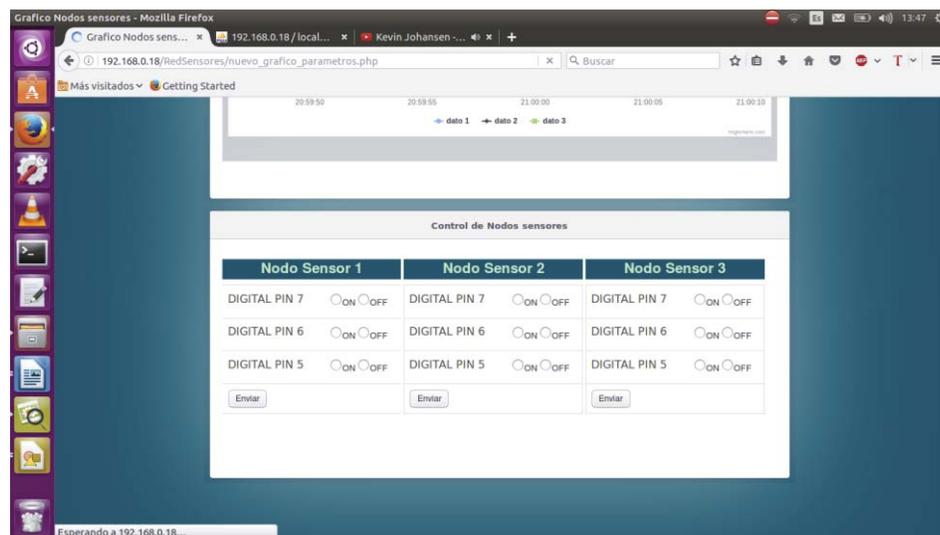


Figura 4-5: Interfaz de control pines Arduino (fuente: elaboración propia)

Se concluyó que el uso de la interfaz web resultó ser fácil, no existe ningún tipo de intervención técnica en la red por parte del usuario, las configuraciones e ingreso de datos en la página son simples y la interfaz funciona de la misma forma para varios navegadores web y desde distintos dispositivos como celulares y computadores.

Con respecto a la velocidad de visualización de datos, la interfaz pudo reproducir de forma fluida los datos entregados desde la red, ya que la velocidad de los datos del gráfico en tiempo real es 3 datos por segundo, no presento una dificultad para las funciones de Highchart.

Discusión y conclusiones

Acerca de las placas de desarrollo:

Desde el estudio de las placas de desarrollo, se encontraron variados libros dedicados al desarrollo de proyectos con las placas Raspberry Pi y Arduino, también cientos de proyectos e ideas disponibles en Internet para implementarse sin costos ni condiciones, esto hace evidente la popularidad a nivel mundial de estas herramientas, reflejada en los proyectos DIY.

DIY (Do It Yourself) o hazlo tú mismo, es una expresión aplicable a una tarea en donde se debe construir, mejorar o reparar un producto, sin intervención de expertos o un servicio profesional [59]. DIY es una tendencia común, ya que siempre las personas han hecho reparaciones o mejoras a objetos cotidianos, lo que es nuevo es el compartir el cómo se hace, con la conectividad global actual se pueden encontrar miles de proyectos en muchas páginas web dedicadas a compartir el DIY. Al popularizarse el DIY se generan distintos fenómenos en la comunidad, como independencia del mercado, reciclaje de materiales, desarrollo de productos o soluciones innovadoras, son sin duda fenómenos beneficiosos para la sociedad.

Los proyectos DIY han ido avanzando en complejidad e incorporado computación física a los procesos, las placas Raspberry Pi y Arduino son protagonistas en el mundo del DIY. En los proyectos que involucran robótica, automatización, telecomunicación o informática, normalmente usan una o ambas placas de desarrollo. Muchos de los proyectos innovadores que se generan desde el DIY, terminan siendo productos comerciales y son construidos en base a Raspberry Pi y Arduino.

Todo proyecto o tutorial realizado es un aprendizaje, la suma y práctica de los aprendizajes se vuelven conocimiento y experiencia, lo que termina en la generación de proyectos e ideas nuevas. El trabajo con las placas de desarrollo abre las puertas al mundo de la innovación, un término muy popular en Chile, ya que según un estudio de la OECD (Organisation for Economic Co-Operation and Development) [60], aumentar el gasto del país en innovación puede generar grandes aumentos en el crecimiento del mismo. Se crean cada vez más fondos concursables con financiamiento estatal, fomentando la investigación y desarrollo (I+D), con el fin de incentivar a la innovación en la pequeña mediana y gran empresa [61], es presumible que varios de los proyectos que ganan estos fondos han estado relacionados con las placas de desarrollo.

En base a lo comentado se concluye que las placas de desarrollo son herramientas para la innovación, aprender a trabajar con ellas es un gran complemento en el mundo profesional actual, por todo el énfasis en la innovación a nivel país, como por la posibilidad de desarrollar soluciones a necesidades que el mercado actual no cubre.

Extrapolando esta conclusión a términos académicos, la educación con respecto a las placas de desarrollo podría ser aplicada en los distintos niveles educacionales, significaría una sociedad con mayor conocimiento en tecnología, aumentando la capacidad como país de innovar, una capacidad estrechamente vinculada con el nivel de desarrollo según el informe [60].

Acerca de la compatibilidad entre las placas

Se hizo un análisis de las fortalezas y debilidades de cada placa, cada una tenía características positivas complementarias desde la visión del diseño de un proyecto de computación física. Raspberry tiene una gran capacidad de procesamiento y almacenamiento, junto con una gran versatilidad de programación, logrando programar servicios de comunicación por puertos de red, manejo de base de datos e implementar un servidor web, estos usos son algunos de las muchas aplicaciones de alta complejidad que se pueden implementar con Raspberry Pi.

Arduino es una herramienta muy versátil, ya que tiene una gran variedad de dispositivos compatibles, que pueden ejecutar procesos complicados como comunicación Wi-Fi o Ethernet. La variedad de módulos genera una variedad de librerías, que hacen de Arduino un lenguaje con muchos recursos de comunicación con electrónica externa. La otra característica importante de Arduino son los puertos análogos, este tipo de pines permite medir las señales de un amplio espectro de sensores.

Para aprovechar las fortalezas de ambas placas, se debe lograr compatibilidad, los problemas en la compatibilidad fueron comentados en el informe, al igual que algunas soluciones comerciales disponibles. Estas soluciones permitían el uso de Raspberry Pi con Shields y sensores compatibles con Arduino, a través del uso de placas de compatibilidad, estas placas utilizaban un micro-controlador Atmel como intermediario entre el Shield o sensor y Raspberry Pi, para programar el microcontrolador con lenguaje Arduino y compartir las mediciones del sensor con Raspberry Pi por USB.

La solución encontrada no parecía aprovechar completamente las capacidades de las placas de desarrollo, ya que Raspberry Pi puede encargarse de muchos más datos que los entregados por el total de sensores o el Shield, por otro lado Arduino puede utilizarse como una interfaz más avanzada entre el sensor y Raspberry.

La red de sensores, demostró ser una estructura que saca mayor provecho de ambas placas, al establecer procesos de mayor complejidad como incluir múltiples agentes en su funcionamiento, la gestión de base de datos y de comunicaciones. Otra ventaja es que con esta

estructura, el proyecto se acerca más a la visión de un producto, ya que con la interfaz web cualquier usuario puede hacer uso del proyecto, sin necesidad de entenderlo complementemente.

Acerca del trabajo

En el desarrollo del proyecto se encontraron recurrente fallas de conexión en los nodos sensores Ethernet, mediante la realización de pruebas se llegó a que el módulo ENC28J60 era la fuente de error, desde la investigación realizada para la implementación de estos nodos, se encontraron algunas posibles causales, algunas hacían referencia a la estabilidad del voltaje de alimentación, otras solo mencionaban un error de fábrica del módulo. Al agregar un regulador de voltaje para alimentar el módulo ENC28J60, se encontraron grande mejoras en el tiempo de conexión, pero el problema no se resolvió completamente ya que habían desconexiones, aunque mucho menos recurrentes. Una conclusión asociada a esto es cambiar el módulo ENC28J60 por un Shields Ethernet W5500, si bien el costo de implementar los nodos sensores subirá, el Shield asegura un buen funcionamiento en espacios libres de ruido de OEM.

El nodo sensor Wi-Fi no presentó problemas de conexión, si bien los requerimientos para implementar el nodo son mayores el esfuerzo es válido, ya que con la comunicación inalámbrica (Wi-Fi) se gana en aspectos como movilidad, resistencia a ruido de OEM, facilidad en la implementación y mantención, finalmente según lo observado la comunicación Wi-Fi resulta ser más estable. Por lo que una conclusión con respecto a este tema es dar prioridad implementación de nodos sensores Wi-Fi, ya que se puede tener una red de sensores móviles, permitiendo cambiar fácilmente la posición de los sensores y la instalación de los mismos.

Se comentó en el informe que Arduino cuenta con una reducida memoria para programación, al igual que la memoria de variables volátiles, durante la programación de los nodos sensores se llegó a usar un 89% de la memoria flash, solo en los procesos de comunicación, con un mínimo procesamiento en el uso de los sensores. Si se pretendiere trabajar con algún sensor o grupo de sensores más complejo, necesitaran más espacio en la programación, esto conduce a la conclusión de que se debe mejorar la forma en la que se programaron los nodos sensores, posibles soluciones son optimizar el tipo y cantidad de variables usadas, el uso de pestañas, trasladar las funciones de comunicación creadas para la red a una librería escrita en C. Esto también permite hacer de Arduino un elemento más inteligente en la red.

Posibles aplicaciones

La red de sensores es un proyecto muy versátil, la aplicación de la red será siempre de monitoreo, pero, el monitoreo puede aplicarse a muchos procesos, el cambio en la red según el proceso son los sensores. Si se suma el control de salidas digitales del nodo sensor, se podría generar el control y monitoreo remoto de un proceso, teniendo disponibles 9 variables para el monitoreo de los procesos y 9 salidas digitales, para controlar los múltiples actuadores que ejercerán el control.

Una posible aplicación para la red de sensores es la domótica, desde una investigación dedicada a este tema [59] se entendió la domótica como una disciplina que busca la integración tecnológica de aplicaciones y dispositivos domésticos, gestionando de forma eficiente, segura y cómoda para el usuario, los aparatos e instalaciones tradicionales que conforman un hogar. Teniendo en mente esta definición, la red de sensores puede utilizarse para el control y monitoreo de varios procesos del hogar, por ejemplo, los registros históricos pueden utilizarse para llevar un control del consumo eléctrico, utilizando un sensor de corriente en la entrada principal de energía de la casa. Otro ejemplo de monitoreo puede ser el consumo de agua, con un sensor de flujo en la entrada principal de agua. Un ejemplo del uso de la red para actuar sobre algún elemento del hogar, puede ser dar alimento a una mascota, abrir las cortinas, cerrar o abrir la entrada principal desde la interfaz web.

Otra fortaleza de la estructura de red de sensores lograda, es el acercamiento a las aplicaciones IoT (Internet of Things), estas aplicaciones buscan generar un ambiente virtual de comunicación humano-máquinas, máquina-máquina (M2M) y humano-ambiente, a través del ambiente (estándares, protocolos, etc.) de Internet, con la finalidad de optimizar los flujos de información, mejorando la distribución de recursos y el desempeño de los procesos asociados. Trabajos de investigación acerca de IoT [62], describen que el Things en el nombre IoT, hace referencia a cualquier entidad que se conecte a Internet, esto incluye dispositivos inteligentes (smart TV, smartphones, etc.), personas, sensores, servidores, etc. También se destaca el crecimiento esperado de la inversión mundial en IoT desde 44 billones de dólares en el 2011 a 1423,09 billones de dólares para el 2020, reflejando la importancia que adquirirán estas aplicaciones.

Como los medios de comunicación de la red son dos estándares del ambiente Internet, la red de sensores desarrollada puede ser considerada como una aplicación IoT para Arduino y Raspberry Pi. Esto hace del proyecto y sus posibles aplicaciones una forma de introducción a soluciones innovadoras, pertenecientes a un campo creciente que toma protagonismo en el desarrollo de tecnologías a nivel mundial.

Trabajos futuros:

El flujo de información de la red es reducido y el nivel de interacción humano-máquina es básico, la red de sensores diseñada es escalable y puede administrar un flujo de información mucho mayor, un trabajo futuro es el mejorar la tasa de datos transmitidos, esto podría implicar cambios en el manejo de la base de datos, también significaría una mejora en la visualización de datos en tiempo real, ya que pueden obtenerse más muestras por segundo.

La red de sensores procura solo la obtención de datos, ya que el control de pines lo ejecuta el usuario, un trabajo futuro puede ser el programar procesos de toma de decisiones, que permita controlar un proceso automáticamente, basado en el análisis de los datos obtenidos. Esto es posible con los muchos lenguajes de programación compatibles con Raspberry Pi y su capacidad de procesamiento.

Un avance significativo para acercar el proyecto a una solución más profesional, sería lograr que los nodos sensores tengan la capacidad de enrutar sus datos, comunicando todos los nodos sensores para encontrar las rutas óptimas para la transmisión de la información.

Acerca de los Objetivos

Un objetivo principal del proyecto es implementar una red de sensores mixta, como se puede observar en el desarrollo del informe, se logró generar que las placas de desarrollo cumplan los objetivos de cada elemento de una red, desde la adquisición de datos hasta la creación de una base de datos y visualización, por lo que este objetivo se considera logrado

Otro objetivo importante era generar una interfaz de visualización de datos, con un control de pines del nodo sensor forma remota. La implementación del servidor LAMP permitió cumplir con este objetivo, en conjunto con los programas de PHP, HTML y CSS. Se puede concluir que este objetivo fue completado ya que se logró tener una plataforma para visualizar los datos obtenidos por la red, accediendo a la base de datos de la red de tal forma que se pudieron visualizar datos antiguos y datos en tiempo real. Por otra parte se pudo realizar una interfaz para el control de pines de forma remota, resultando en un control real del funcionamiento de tres pines digitales del nodo sensor, con una latencia no mayor a 1[seg], considerando este resultado como satisfactorio, se da como cumplido este objetivo.

Finalmente con respecto a la compatibilidad entre Raspberry Pi y Arduino, se pudo observar desde el trabajo desarrollado, que existe en la red de sensores una comunicación entre ambas placas de desarrollo, que por medio de esta comunicación ambas placas intercambian información de los procesos ejecutados, permitiendo una coordinación en pos de un objetivo común. Por lo que se puede considerar lograda la compatibilidad entre ambas placas de desarrollo, ya que trabajan de forma coordinada para lograr un objetivo común.

Bibliografía

- [1] opensource initiative, «opensource.org.» [En línea]. Available: <https://opensource.org/osd-annotated>. [Último acceso: 14 febrero 2017].
- [2] D. Nilanjan y M. Amartya, «Embedded Systems and Robotics with Open Source Tools,» Londres, CRC Press, 2016, p. 4.
- [3] A. K. Dennis, Raspberry Pi Home Automation with Arduino, Birgminham: Packt, 2013.
- [4] S. Sitharama Iyengar, N. Parameshwaran, V. V. Phoha, N. Balakrishnan y C. D. Okoye.
- [5] S. Iyengar y R. Brooks, «Distributed Sensor Networks,» de *Distributed Sensor Networks*, London, Chapman & Hall/CRC, 2005, pp. 1-10.
- [6] L. Xinrong y F. Sheikh, «Wireless Sensor Network System Design using Raspberry Pi and,» de *The 9th International Conference on Future Networks and Communications*, Texas, 2014.
- [7] H. Shahnasser y V. Raghavan, «Embedded Wireless Sensor Network for Environment,» *Journal of Advances in Computer Networks*, vol. 3, nº 1, pp. 1-10, 2015.
- [8] dechile, «etimología de proyecto,» 3 Abril 2017. [En línea]. Available: <http://etimologias.dechile.net/?proyecto>. [Último acceso: 30 Marzo 2017].
- [9] M. P. Esteso, «geekytheory,» [En línea]. Available: <https://geekytheory.com/arduino-raspberry-pi-rasduino>. [Último acceso: 12 4 2017].
- [10] Dexter Industries, «dexterindusties.com,» [En línea]. Available: <http://www.dexterindustries.com/arduberry-tutorials-documentation/>. [Último acceso: 2 5 2016].
- [11] Dfrobot, «dfrobot.com,» [En línea]. Available: <http://www.dfrobot.com>. [Último acceso: 2 5

2016].

- [12] Cooking Hacks, «cooking-hacks,» [En línea]. Available: <https://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge/#step3>. [Último acceso: 1 5 2016].
- [13] F. Sheikh Mohammad, «A low-cost Wireless sensor network system using Raspberry pi and arduino for enviromental monitoring applications,» [En línea]. Available: http://digital.library.unt.edu/ark:/67531/metadc500182/m2/1/high_res_d/thesis.pdf. [Último acceso: 9 10 2016].
- [14] MCI Electronics, «¿Qué es XBee?,» [En línea]. Available: <http://xbee.cl/que-es-xbee/>. [Último acceso: 17 4 2017].
- [15] C. Bell, *Beginning sensor networks whit arduino and raspberry pi*, Newyork: Technology in action, 2013.
- [16] usb.org, «FAQ usb.org,» [En línea]. Available: <http://www.usb.org/developers/usbfaq#cab1>. [Último acceso: 17 4 2017].
- [17] C. Chong y K. Srikanta, «Sensor Networks: Evolution, Opportunities, and Challenges,» *PROCEEDINGS OF THE IEEE*, vol. 8, nº 91, pp. 1247-1256, 2003.
- [18] C. Nishimura y D. Conlon M, «IUSS Dual Use: Monitoring whales and earthquakes using SOSUS,» *Marine Technology Society Journal*, vol. 27, nº 4, pp. 13-21, 1994.
- [19] J. W. Garden, V. K. Varadan y O. O. Awadelkarim, «Microsensors, MEMS and smart devices,» Newyork, Wiley, 2001, pp. 1-8.
- [20] Bussines Week, «Ideas for the 21st century,» *Bussines Week*, 1999.
- [21] esri, «esri.in,» 2016. [En línea]. Available: <http://www.esri.in/~media/esri-india/files/pdfs/news/arcindianews/Vol9/sensors-for-smart-cities.pdf?la=en>. [Último acceso: 1 12 2016].
- [22] A. Broring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang y R. Lemmens, «New Generation Sensor Web Enablement,» *Sensors*, nº 11, pp. 2652-2699, 2011.
- [23] M. Richardson y S. Wallace, «Getting Started With Raspberry pi,» de *Getting Started With Raspberry pi*, San Francisco, Maker media, 2016, p. 3.
- [24] Raspberry pi, «www.raspberrypi.org,» [En línea]. Available: <https://www.raspberrypi.org/products/>. [Último acceso: 28 febrero 2017].

- [25] S. Sjogelid, «Raspberry pi for Secret Agents,» de *Raspberry pi for Secret Agents*, livery Place, Packt, 2015, pp. 6-8.
- [26] Broadcom Corporation, «www.raspberrypi.org,» 2012. [En línea]. Available: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/BCM2835-ARM-Peripherals.pdf>. [Último acceso: 13 enero 2017].
- [27] Raspberry shop, «www.Raspberrishop.es,» [En línea]. Available: <http://www.raspberrishop.es/hardware-raspberry-pi.php>. [Último acceso: 1 enero 2017].
- [28] Raspberry pi organization, «Setup,» [En línea]. Available: <https://www.raspberrypi.org/documentation/setup/>. [Último acceso: 5 enero 2017].
- [29] Raspberry organization, «Downloads,» [En línea]. Available: <https://www.raspberrypi.org/downloads/>. [Último acceso: 28 Diciembre 2016].
- [30] D. Kushner, «spectrum.ieee.org,» [En línea]. Available: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>. [Último acceso: 1 noviembre 2017].
- [31] Arduino organization, «arduino.cc,» [En línea]. Available: <http://www.arduino.cc/en/Main/Products>. [Último acceso: 1 11 2017].
- [32] M. Banzi y M. Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, Sebastopol: Maker Media, 2015.
- [33] J. Oxeer, «shieldlist.org,» [En línea]. Available: <http://shieldlist.org/>. [Último acceso: 1 11 2017].
- [34] P. Miller, «TCP/IP: The Ultimate Protocol Guide, Volumen 1,» Florida, Brown Walker, 2009, pp. 26-29.
- [35] C. E. Spurgeon, «Ethernet: The Definitive Guide: The Definitive Guide,» Sebaspool, O'reilly, 2000, pp. 24-31.
- [36] H. Davis, de *Absolute Beginner's Guide to Wi-Fi Wireless Networking*, Indiana, QUE, 2004, pp. 18-21.
- [37] J. Ross, «The Book of Wireless: A Painless Guide to Wi-fi and Broadband Wireless,» de *The Book of Wireless: A Painless Guide to Wi-fi and Broadband Wireless*, San Francisco, No Startch, 2008, pp. 1-43.
- [38] H. Zhang, *Orthogonal Frequency Division Multiplexing for*, Georgia, 2004.

- [39] ecovi, «ecovi.uagro,» [En línea]. Available: <http://ecovi.uagro.mx/ccna1/course/module4/4.4.4.8/4.4.4.8.html>. [Último acceso: 28 11 2016].
- [40] Nurds, «nurdspace,» [En línea]. Available: <https://nurdspace.nl/ESP8266#Characteristics>. [Último acceso: 1 11 2017].
- [41] Naylamp mechatronics, «Tutorial ESP8266 parte 1,» [En línea]. Available: http://www.naylampmechatronics.com/blog/21_Tutorial-ESP8266-Parte-I.html. [Último acceso: 18 1 2017].
- [42] Rancidbacon, «rancidbacon.com,» [En línea]. Available: http://rancidbacon.com/files/kiwicon8/ESP8266_WiFi_Module_Quick_Start_Guide_v_1.0.4.pdf. [Último acceso: 19 12 2016].
- [43] Espressif Systems, «nurdspace.com,» 12 Octubre 2013. [En línea]. Available: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf. [Último acceso: 20 12 2016].
- [44] Microchip, «cloudfront,» [En línea]. Available: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/BreakoutBoards/39662b.pdf>. [Último acceso: 11 enero 2017].
- [45] Elinux, «elinux,» [En línea]. Available: http://elinux.org/RPi_SD_cards. [Último acceso: 11 enero 2017].
- [46] htpc guides, «htpc guides,» [En línea]. Available: <http://www.htpcguides.com/create-samba-share-raspberry-pi/>. [Último acceso: 15 Enero 2017].
- [47] Intef, «ite.educación,» [En línea]. Available: http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m4/instalacin_y_configuracin_de_samba.html. [Último acceso: 20 Enero 2017].
- [48] Python software foundation, «docs.python,» [En línea]. Available: <https://docs.python.org/2/library/socket.html>. [Último acceso: 1 Enero 2017].
- [49] A. G. Blank, TCP / IP JumpStart: Internet Protocol Basics, London: sybex, 2000.
- [50] F. Minera, «Desarrollo PHP y MySQL,» Users, p. 73.
- [51] P. DuBois, «Introduccion,» de *MySQL*, Developer's Library, pp. 1-3.
- [52] M. P. Estes, «geekytheory,» [En línea]. Available: <https://geekytheory.com/tutorial->

- raspberry-pi-15-instalacion-de-apache-mysql-php/. [Último acceso: 20 8 2016].
- [53] aulaclic, «aulaclic.es,» [En línea]. Available: http://www.aulaclic.es/dreamweaver-cs5/t_20_10.htm. [Último acceso: 25 Noviembre 2016].
- [54] S. Breuning, «raspberrywebserver,» 2014. [En línea]. Available: <http://raspberrywebserver.com/serveradmin/run-a-script-on-start-up.html>. [Último acceso: 10 Diciembre 2016].
- [55] diymakers, «diymakers,» 7 agosto 2014. [En línea]. Available: <http://diymakers.es/raspberry-pi-como-servidor-web/>. [Último acceso: 11 Enero 2017].
- [56] Highchart, «Highchart,» [En línea]. Available: <http://www.highcharts.com/>. [Último acceso: 14 Diciembre 2016].
- [57] M. P. Estesos, «Tutorial Raspberry Pi: Conexión a Xively (Internet Of Things),» [En línea]. Available: <https://geekytheory.com/conectando-la-raspberry-pi-a-xively-internet-of-things/>. [Último acceso: 17 abril 2017].
- [58] sajingeo, «Host Your Website on Raspberry Pi,» [En línea]. Available: <http://www.instructables.com/id/Host-your-website-on-Raspberry-pi/>. [Último acceso: 17 abril 2017].
- [59] X. P. V. Stefan Junstrand, Domótica y hogar digital, Madrid: Paraninfo.
- [60] OECD, «Estudios de la OECD sobre políticas de innovación: Chile 2007,» 2007.
- [61] chile innova, «economia.gob,» [En línea]. Available: http://www.economia.gob.cl/1540/articles-187103_recurso_1.pdf. [Último acceso: 20 febrero 2017].
- [62] A. V. D. Rajkumar Buyya, «Internet of Things: Principles and Paradigms,» Boston, Morgan Kaufmann, 2016, pp. 3-7.