



**UNIVERSIDAD DEL PAPALOAPAN**

**CAMPUS LOMA BONITA**

**INGENIERÍA EN MECATRÓNICA**

**ADQUISICIÓN DE DATOS POR  
MEDIO DE WI-FI**

**Tesis profesional para obtener el grado de  
INGENIERO EN MECATRÓNICA**

**Presenta:**

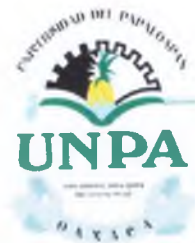
**GLADYS SERRA ALAVEZ**

**Asesor de tesis:**

**M.C. Rafael Fernando González Zarate**

**LOMA BONITA, OAX.**

**AGOSTO 2015**



## UNIVERSIDAD DEL PAPALOAPAN

### INGENIERIA MECATRONICA

LA PRESENTE TESIS TITULADA “ADQUISICION DE DATOS POR MEDIO DE WI-FI”. PRESENTADA POR LA SUSTENTANTE DE LICENCIATURA C. GLADYS SERRA ALAVEZ BAJO LA ASESORIA DEL M. EN C. RAFAEL FERNANDO GONZALEZ ZARATE, HA SIDO REVISADA Y ACEPTADA POR EL COMITE EXAMINADOR PARA SER DEFENDIDA EN EL EXAMEN PROFESIONAL Y OBTENER EL TITULO DE INGENIERO EN MECATRONICA.

M. en C. José Luis Najera Sánchez  
Jefe de la carrera de Ingeniería en  
Mecatrónica.



Dr. Hiram Netzahualcoyotl García  
Lozano.  
Presidente.

M. en C. Rafael Fernando González Zárate.  
Asesor de tesis.

Dr. Eduardo Sánchez Soto.  
Secretario.

M. en C. Rafael Fernando González Zárate.  
Vocal.

# Dedicatoria

Estas tesis se la dedico con todo mi cariño a:

## *A Dios*

Por haberme dado una oportunidad de existir, ayudándome en los momentos más difíciles que surgieron en este comienzo de vida, por haberme dado la inteligencia de resolver los problemas que me venían, por haberme dado la fuerza de seguir adelante sin importar los obstáculos que se ponían en mi camino, por ser un factor importante en mi vida, te dedico esta y todo lo que viene por delante si aún me sigues dando vida para concluir las.

## *A mis padres*

Que con mucho cariño me fueron formando mi crecimiento con amor y buenos consejos, porque sin su apoyo no hubiera logrado esta meta, porque son las personas más importantes de mi vida que creyeron en mí, principalmente a mi madre que me brindo todo el apoyo que pudo darme para salir adelante y estar al cuidado de mí, por tener el tiempo de escucharme en los momentos que más la necesitaba.

## *A mis hermanos*

Quienes fueron y son un gran apoyo para mí, a Marcos quien con su experiencia hizo que fuera más fuerte en esta vida, a Idalia quien me regañaban para que seguir adelante que no me diera por vencida cuando se me presentaba algún problema y quien me daba ánimos para seguir y a Nallely quien me apoyo en todo, que desde pequeña siempre me cuidaba y me sigue cuidando. Son a ellos a quien le dedico esta tesis que hoy concluyo, ya que son las personas que más admiro.

## *A mis profesores*

Porque fueron la parte más importante en mi formación profesional, por haberme brindado el apoyo incondicional en estos cinco años enseñándome nuevos conocimientos.

*Gladys S. A*

## Agradecimientos

Al pueblo de México que pagó gran parte de mis estudios con sus impuestos.

A la **Universidad del Papaloapan** por haberme dado la oportunidad de iniciar una carrera y concluirla.

A mis profesores que en estos cinco años aprendí mucho de ellos, principalmente a mi asesor de tesis **M.C. Rafael Fernando González Zarate** quien se tomó el tiempo de ayudarme en las dudas que surgían y para las cuales buscábamos una solución, al **Dr. Hiram Netzahualcóyotl García Lozano** quien me apoyo en las correcciones de mi redacción, así como a mis sinodales **Dr. Eduardo Sánchez Soto** y al **M.C. Eduardo Ortiz Hernández** quienes tuvieron el tiempo de estudiar y revisar mi tesis y la aprobaron.

A mis amigos quienes ya forman parte de mi vida, por haberme brindado su amistad y confiado en mí, aguantándome estos cinco años. Quiero agradecer principalmente a mi amiga Olga con quien viví momentos tristes y felices, por el apoyo incondicional que nos brindamos cada una, por repetirme muchas veces que no pusiera al revés sus apellidos y aun así no lo hacia, por las noches de desvelos que pasábamos haciendo trabajos y en ocasiones solo platicando de muchas cosas, por eso y por muchas cosas mas te quiero agradecer amiga.

A una persona especial que durante los cinco años de la carrera fue importante y que en el presente aún lo es y espero que en un futuro lo sea, por apoyarme y hacerme reír mucho.

A todos estas personas y a las que me faltó por mencionar pero que las llevo en mis recuerdos.

*¡Muchas Gracias!*

## Resumen

*En esta tesis se realiza el diseño y construcción de un sistema de adquisición de datos (AD) con comunicación inalámbrica (WiFi). Este sistema estará constituido por un módulo con conexión inalámbrica (WiFly) conectado a una red local, el cual será acoplado con un sistema mínimo de desarrollo (Arduino Mega) de manera que al trabajar en conjunto con sensores, adquirirán señales físicas deseadas. Esta información sensorial se presenta en una interfaz computacional con acceso a la red local.*

*Con este fin se realizó un estudio sobre la comunicación inalámbrica, su modo de configuración, que tipos de redes existen y en nuestro caso cual sería útil para trabajar con el módulo. Adicionalmente se investigaron los tipos de programación que convenía para realizar la interfaz.*

*En el primer capítulo habla brevemente de la historia de las comunicaciones y de los sistemas de adquisición de datos: que son, como funcionan y que aplicaciones tienen en la práctica. Se hace hincapié en el crecimiento que a lo largo del tiempo han tenido estos sistemas así como su evolución. En el segundo capítulo se presentan los sistemas de adquisición de datos. Aquí se explica, de manera sencilla, conceptos básicos que conforman los sistemas AD (datos analógicos y digitales así como su importancia). A continuación, en el tercer capítulo, se exponen los distintos tipos de comunicación inalámbrica, las nuevas tecnologías, así como los protocolos estándares que en la actualidad se están utilizando. Para finalizar este capítulo se habla de los sitios web, que son una manera más sencilla de interactuar entre el hombre y la computadora. Por último, en el cuarto capítulo, se da a conocer la descripción del sistema diseñado. Este capítulo se centra en el acoplamiento del módulo y la tarjeta Arduino, así como en la adquisición de datos mediante una tarjeta con tecnología Wi-Fi como medio de transmisión. Dicha información, diseñada con programación HTML y PHP, podrá ser visualizada por medio de cualquier dispositivo de cómputo que tenga acceso a la red en la cual está conectado el sistema diseñado.*

*Se finaliza con breves conclusiones. En ellas se presentan distintas aplicaciones donde se muestra el funcionamiento final de este sistema.*

## Abstract

*The objective of this thesis is to carry out the design and construction of a data acquisition system (DAQ) with (WIFI) wireless communications. This system will be made up of a module with free wireless (WiFi) connected to a local network, which will be coupled with a system of minimal development (Arduino Mega) so that the system works together with sensors, to acquire desired physical signals. This sensory information will be presented in a computational interface with access to local network.*

*For this reason a study was carried out on wireless communications, as it is one of the most important parts of this thesis, its configuration mode was investigated, what types of network exist and in our case which would be the most useful to work with the module. Other types of programming were also investigated to see which would be suited for the interface.*

*The first chapter speaks briefly of the history of communications and (DAQ) systems, what they are, how they function and what useful applications they have. Also, emphasizing on the growth and evolution of these systems over time.*

*In the second chapter the (DAQ) systems are presented. Here the basic concepts that make up the analog and digital data systems, which are of equal importance are also presented. Continuing in the third chapter, the different types of wireless communication, new technologies, as well as standard protocols currently being used are explained. This chapter also speaks of web sites which are the easiest ways for man and computer to interact.*

*Finally, in the fourth chapter the manner in which the systems are designed are described. This chapter focuses on the Arduino card, as well as by using a (DAQ) card with WiFi technology as a means of transmission. This information, which is designed with HTML y PHP programming, may be displayed by means of any computer device that has access to a local network.*

*In conclusion, the distinct applications and functional operations of this system will be demonstrated.*

# Índice

Dedicatoria	I
Agradecimientos	II
Resumen	III
Abstract	IV
Índice de figuras	VII
Índice de tablas	IX
Índice de códigos	X
<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Historia de la comunicación . . . . .	1
1.2. Adquisición de datos . . . . .	4
1.3. Aplicaciones . . . . .	4
<b>2. SISTEMAS DE ADQUISICIÓN DE DATOS</b>	<b>6</b>
2.1. Introducción . . . . .	6
2.2. Datos analógicos y digitales . . . . .	6
2.2.1. Señales analógicas . . . . .	7
2.2.2. Señales digitales . . . . .	7
2.3. Sensores . . . . .	9
2.3.1. Sensores de luz (LDR) . . . . .	9
2.3.2. Sensores de temperatura (LM35) . . . . .	12
2.4. Tarjeta de adquisición de datos . . . . .	13
<b>3. COMUNICACIÓN INALÁMBRICA</b>	<b>15</b>
3.1. Introducción . . . . .	15
3.2. Tipos de redes . . . . .	15
3.3. Protocolos de redes inalámbricas . . . . .	20
3.4. Comunicación mediante sockets . . . . .	21
3.5. Servicio Web (Red informática mundial) . . . . .	24
3.6. Aplicaciones con comunicación inalámbrica. . . . .	27

3.6.1.	Plataforma domótica para la interconexión de objetos físicos a Internet empleando conectividad WiFi . . . . .	27
3.6.2.	Robot auxiliar de limpieza de suelos doméstico . . . . .	29
3.6.3.	Sistema de telemetría para implantarse en prototipos móviles para el monitoreo de sitios remotos . . . . .	29
3.6.4.	Comunicación Inalámbrica Digital con un Controlador Lógico Programable para el Control Industrial en Zonas de Alto Riesgo . . . . .	30
<b>4.</b>	<b>IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS</b>	<b>32</b>
4.1.	Introducción . . . . .	32
4.2.	Descripción del Módulo WiFly . . . . .	32
4.3.	Conexión y configuración del Módulo WiFly . . . . .	35
4.4.	Pruebas del sistema de adquisición . . . . .	40
4.5.	Servicio Web implementado en el módulo WiFly . . . . .	42
4.6.	Funcionamiento . . . . .	50
<b>5.</b>	<b>CONCLUSIONES</b>	<b>56</b>
<b>A.</b>	<b>Códigos <i>Arduino</i><sup>®</sup></b>	<b>57</b>
<b>B.</b>	<b>Código HTML en conjunto con PHP</b>	<b>59</b>
	<b>Bibliografía</b>	<b>62</b>

## Índice de figuras

1.1. Historia de la comunicación. . . . .	1
1.2. Alejandro Graham Bell teléfono . . . . .	2
1.3. El transistor impactó en las comunicaciones. . . . .	2
1.4. Los microprocesadores, comienzo tecnológico de las telecomunicaciones . . . . .	3
1.5. Sistemas de adquisición utilizados en el monitoreo de pruebas de calidad. . . . .	5
2.1. Componentes de un sistema de adquisición de datos . . . . .	6
2.2. Representación de una señal analógica. . . . .	7
2.3. Representación de una señal digital. . . . .	8
2.4. Tipos de sensores . . . . .	9
2.5. En la Figura (a) esquema de una fotoresistencia (b) símbolo. . . . .	10
2.6. Velocidad de respuesta de un fotoconductor típico de una pieza. . . . .	11
2.7. Tarjeta AD para comunicación con PC. . . . .	13
2.8. Medio de comunicación para la transferencia de información de los sistemas AD. . . . .	14
3.1. Clasificación de las redes inalámbricas. . . . .	16
3.2. Esquema de la red WAN (World Area Network). . . . .	17
3.3. Esquema de la red LAN (Local Area Network). . . . .	17
3.4. Ejemplo de una red de área personal PAN (Personal Area Network). . . . .	18
3.5. Configuración de transmisión y recepción del infrarrojo. . . . .	18
3.6. Ejemplo de conexión punto a punto. . . . .	19
3.7. Ejemplo de roaming entre varios access points. . . . .	20
3.8. Modelo cliente-servidor . . . . .	22
3.9. Ciclo de vida de un conector TCP. . . . .	24
3.10. Funcionamiento de un servidor de aplicaciones PHP. . . . .	26
3.11. Interfaz Gráfica de Usuario (GUI) del prototipo de aplicación de plataforma domótica. . . . .	28
3.12. Módulo de Comunicación y Control WiFly . . . . .	28
3.13. Prototipo ensamblado del robot auxiliar de limpieza de suelos doméstico. . . . .	29
3.14. Imagen tomada a Gabinete de Sistema Telemétrico en terreno para pruebas . . . . .	30

3.15. Prueba de operación con obstáculos en espacio cerrado . . . . .	31
4.1. Diagrama del sistema de adquisición de datos . . . . .	32
4.2. Magnitudes del módulo inalámbrico. . . . .	33
4.3. Módulo de comunicación inalámbrico WiFly GSX RN. . . . .	34
4.4. Placa Arduino Mega 2560 . . . . .	35
4.5. Diseño del circuito de la WiFly acoplado al circuito del Arduino Mega. . . . .	35
4.6. Esquema de conexión del WiFly con <i>ArduinoMega</i> ®. . . . .	36
4.7. Ensamble final del módulo WiFly al Arduino mega. . . . .	36
4.8. Respuesta del módulo al conectarse. . . . .	37
4.9. Ejecución de comandos. . . . .	38
4.10. Detección de la red creada por la WiFly . . . . .	39
4.11. Características configuradas por medio de sus comandos. . . . .	39
4.12. Transmitiendo una cadena. . . . .	41
4.13. Visualización del voltaje adquirido por el potenciómetro. . . . .	42
4.14. Interfaz para mostrar los datos. . . . .	44
4.15. Software XAMPP para inicializar la conexión localhost. . . . .	50
4.16. Divisor de voltaje usando una fotorresistencia. . . . .	51
4.17. Circuito de la fotorresistencia donde la salida disminuye al aumentar la intensidad luminosa. . . . .	51
4.18. Interfaz donde se muestra el cambio de la salida cuando se aumenta la intensidad luminosa de la fotorresistencia. . . . .	52
4.19. Interacción del módulo WiFly con la fotorresistencia. . . . .	53
4.20. En la Figura (a) se muestra la configuración del sensor de tempe- ratura y en la Figura (b) circuito del sensor LM35 con el Arduino. . . . .	54
4.21. Temperatura adquirida por el sensor en un salón y mostrada en la interfaz. . . . .	55
4.22. Circuito real del sensor junto con el Arduino y el módulo WiFly. . . . .	55

## Índice de Tablas

4.1. Descripción del funcionamiento de los leds del módulo WiFly, . . .	34
4.2. Tabla de modo de autenticación . . . . .	38
4.3. Familia de protocolos . . . . .	45
4.4. Tipos de sockets disponibles . . . . .	46
4.5. Protocolos comunes . . . . .	47

## Listados de código

1.	Comunicación del puerto serial . . . . .	40
2.	Envío de la cadena Hola Mundo . . . . .	41
3.	Adquisición del valor del potenciómetro. . . . .	42
4.	Interfaz del módulo WiFly en HTML. . . . .	43
5.	Estructura de inicio de la programación en PHP. . . . .	44
6.	Estructura del socket create en PHP. . . . .	45
7.	Socket last error para obtener una explicación del error. . . . .	47
8.	Estructura del socket de conexión. . . . .	48
9.	Conexión del socket a partir de una IP y un puerto. . . . .	49
10.	Parametros de socket recvmsg. . . . .	49
11.	Obtención del valor analógico de la fotorresistencia. . . . .	52
12.	Adquiriendo la señal del sensor LM35. . . . .	54

# 1. INTRODUCCIÓN

## 1.1. Historia de la comunicación

Año tras año las circunstancias particulares de la innovación tecnológica han requerido del intercambio de información, lo que llevó al rápido desarrollo de los sistemas de comunicación. Los inicios de la comunicación entre los humanos fue la por medio de pinturas rupestre (Figura 1.1a). En la antigua Grecia el arte de hablar y persuadir llamada la retórica, era un área de estudio importante (Figura 1.1b) .



(a) Forma antigua de comunicación de los humanos (b) Estudio de la retórica en la antigua Grecia

Figura 1.1: Historia de la comunicación.

Los sistemas digitales de transmisión, representados inicialmente por el telégrafo, fueron desarrollados a partir de 1850. En 1876, Alejandro Graham Bell, desarrolla y obtiene una patente por el teléfono. Este aparato convirtió en una realidad práctica la transmisión del habla mediante la codificación eléctrica y la replicación del sonido. La primera versión del teléfono fue burda y limitada (Figura 1.2) y únicamente permitía la comunicación a distancias cortas. A pocos años de su inicio, se generó el interés por automatizarlo. En 1897 A.B. Strowger, ideó el conmutador automático que lleva su nombre, el cual fue el más utilizado.



Figura 1.2: Alejandro Graham Bell teléfono

Con el transcurso de los años la transmisión del habla fue mejorando hasta llegar al desarrollo de los sistemas de imagen y voz. De esta forma Philo T. Farnsworth probó el primer sistema de televisión totalmente electrónico en 1928, y posteriormente Vladimir K. Zworykin hizo lo mismo en 1929.

A la par de este descubrimiento Edwin H. Armstrong propone un nuevo sistema que revolucionó la transmisión de la voz a distancias más largas que las obtenidas hasta antes de 1933, este nuevo sistema se nombró como Frecuencia Modulada o FM.

Los sistemas de comunicación tuvieron importantes descubrimientos durante este periodo de tiempo, entre éstos el desarrollo de la primera computadora digital electrónica, y la invención del transistor (Figura 1.3) en los Bell laboratories en 1947. Esto estimuló la aplicación de la electrónica en las comunicaciones, particularmente en conmutadores (utilizando relevadores electromecánicos) y en sistemas digitales, con el fin de mejorar la confiabilidad, ampliar la capacidad y reducir los costos.



Figura 1.3: El transistor impactó en las comunicaciones.

Fue hasta 1950 cuando se logró la comunicación, a larga distancia, entre computadoras y las terminales. La invención del primer circuito integrado (CI) de silicio en 1958, el desarrollo de estos a una escala muy grande de integración, así como su evolución a microprocesadores de un solo chip (Figura 1.4), trajo como consecuencia una transformación en la naturaleza del procesamiento de señales y con ello la industria de las telecomunicaciones cambió para siempre.



Figura 1.4: Los microprocesadores, comienzo tecnológico de las telecomunicaciones

Entre 1950 y 1970 se realizaron varios estudios entorno a las redes de computadoras. No obstante, el más importante en términos del impacto en las comunicaciones fue el Advanced Research Project Agency Network (ARPANET). Este trabajo fue pionero en el manejo de paquetes de datos. Este último se renombró en 1985 como Internet. Este modo de comunicación ha crecido de forma exponencial desde entonces. Inicialmente se necesitó de cables para su conexión, sin embargo, debido a las nuevas tecnologías (wifi, Bluetooth, infrarrojo) presente en las computadoras portátiles, agendas electrónicas y teléfonos móviles de acceso a este medio, reclamó una red sin ataduras ni cables.

Desde el 2000 hasta la actualidad se han dado grandes avances en distintas áreas de la tecnología, por ejemplo: en la microelectrónica, computadoras digitales, sistemas de ondas luminosas, procesamiento de señales digitales, receptores sintonizados digitalmente, sistemas de espectro ensanchado, sistemas digitales satelitales y sistemas personales de comunicación, entre otras. Estos avances han propiciado el asombroso crecimiento en el mundo de las telecomunicaciones [6].

## **1.2. Adquisición de datos**

Las comunicaciones juegan un papel muy importante en nuestra vida diaria. Los seres humanos nos comunicamos de distintas maneras, ya sea con el habla, imágenes, símbolos, entre otras. Desde el punto de vista tecnológico, todas estas formas de expresión pueden ser consideradas datos. Para que estos puedan ser procesados en los sistemas de comunicación, es decir, utilizando sistemas de adquisición.

Estos se pueden describir como la manera de medir, con un sistema digital, un fenómeno físico tal como temperatura, presión, sonido, video, entre otras. Muchas veces este sistema está conformado por una computadora en conjunto con una tarjeta AD las cuales realiza en forma ágil el procesamiento, análisis, almacenamiento, distribución y despliegue de los datos. En el mercado existen muchas tarjetas de AD las cuales cuentan con entradas y salidas analógicas, digitales entre otras especializadas. Teniendo en cuenta los beneficios que nos da la programación, los sistemas AD pueden realizar la recolección de datos sin necesidad de que el usuario esté presente, La AD es utilizada en la industria, la investigación científica, el control de máquinas y de producción, en la detección de fallas y el control de calidad entre otras aplicaciones.

En algunas ocasiones la información recabada por un sistema AD es requerida no solo en el lugar donde fue obtenida, sino también en otra localidad. Por ejemplo la información recabada por satélites referente al clima se requiere en estaciones meteorológicas en la tierra por lo que esta información debe ser transmitida y recibida. Existen muchos dispositivos para transmitir la señal y para enviar y recibir datos de un lugar a otro. Esta puede hacerse a través de diferentes medios como puede ser un cable, fibra óptica o el "espacio" como en el caso de las transmisiones de radiofrecuencia y comunicación inalámbrica.

## **1.3. Aplicaciones**

Podemos encontrar aplicaciones de los sistemas AD en todas las áreas de la tecnología. Desde un sistema de tipo encendido y apagado para el control de humedad en un sistema de riego, hasta sistemas complejos como control de lazo cerrado con múltiples entradas y salidas de equipo de generación de energía

eléctrica, tales como turbinas y compresores. Estos pueden prevenir interrupciones inesperadas, optimizar el rendimiento de la máquina y reducir los costos de reparación y mantenimiento.

Los ingenieros y operadores de pruebas de manufactura y calidad comúnmente usan hardware y software de adquisición de datos para verificar la funcionalidad y asegurar la calidad de los productos terminados (Figura 1.5). Esto puede incluir un amplio rango de pruebas eléctricas, mecánicas o ambientales ya sean manuales o automatizadas.



Figura 1.5: Sistemas de adquisición utilizados en el monitoreo de pruebas de calidad.

## 2. SISTEMAS DE ADQUISICIÓN DE DATOS

### 2.1. Introducción

Un sistema de adquisición de datos Figura 2.1 consiste en un conjunto de sensores, hardware de AD y una computadora personal con software programable.

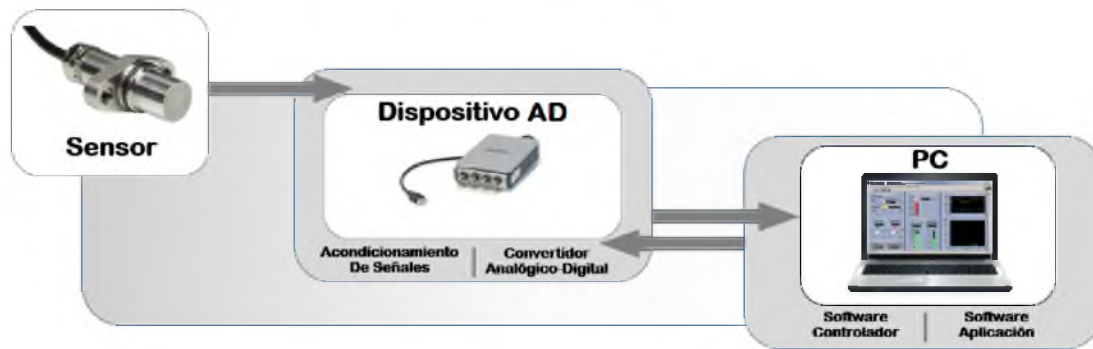


Figura 2.1: Componentes de un sistema de adquisición de datos

En distintos sectores, donde se requiere registrar y almacenar información de procesos en forma remota, pueden ser utilizados AD basados en Wi-Fi. Estos deben de contar con una interfaz que permita obtener información que se esté monitoreando desde cualquier parte del mundo, si se tiene acceso a Internet, o en su caso, trabajando en un espacio cerrado a una conexión de red local o simplemente una red punto a punto.

En las siguientes secciones abordaremos cada una de las partes que componen un sistema de AD así como el tipo de señales que maneja.

### 2.2. Datos analógicos y digitales

En esta sección se hablará de los datos analógicos y digitales a fin de entender el comportamiento y los tipos de señales que maneja un sistema AD. Las señales de entrada del sistema AD tienen una forma continua (analógica) y las salidas son del tipo digital.

Los términos analógico y digital corresponden a sistemas continuos y discretos, respectivamente. Estos términos son usados frecuentemente en la comunicación de

datos en al menos en tres contextos: datos, señalización y transmisión. Definimos datos como entidad que se transmite información. Las señales son representaciones eléctricas o electromagnéticas de datos. La transmisión es la propagación física de la señal a lo largo de un medio [15].

### 2.2.1. Señales analógicas

Las señales son variables eléctricas que evolucionan en el tiempo en forma similar a alguna variable física. Estas señales toman valores continuos. En la Figura 2.2 se observa una señal analógica típica. Las variaciones en amplitud y en frecuencia transportan la frecuencia fundamental de la voz o de la música. Se usan señales similares para transmitir imágenes en televisión, pero con frecuencias mucho más altas.



Figura 2.2: Representación de una señal analógica.

La transmisión analógica es una forma de enviar las señales, y en cualquier caso, está se irá debilitando (atenuándose) con la distancia. Para conseguir distancias más largas, el sistema de transmisión analógico incluye amplificadores que inyectan energía en la señal. Desgraciadamente el amplificador también inyecta energía en las componentes de ruido. Para conseguir distancias mayores, al utilizar amplificadores en cascada, la señal se distorsiona cada vez más. Para datos analógicos como la voz, se puede tolerar una pequeña distorsión, ya que en ese caso los datos siguen siendo inteligibles.

### 2.2.2. Señales digitales

Las señales digitales son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo, transmitiendo información según un código previa-

mente acordado. Cada nivel eléctrico representa uno de dos símbolos lógicos: 0 ó 1, V o F. Los niveles específicos dependen del tipo de dispositivos utilizado. Por ejemplo si se emplean componentes de la familia lógica TTL (transistor-transistor-logic) los niveles son 0 V y 5 V, aunque cualquier valor por debajo de 0,8 V es correctamente interpretado como un 0 y cualquier valor por encima de 2 V es interpretado como un 1. En el caso de la familia CMOS (complementary metal-oxide-semiconductor) los valores dependen de la alimentación. Para una alimentación de +5 V, los valores ideales son también 0 V y 5 V, pero se reconoce un 0 hasta 2,25 V y un 1 a partir de 2,75 V.

Un ejemplo de una señal digital son las teclas pulsadas por un usuario desde una computadora que convierten en una cadena de dígitos binarios (1 y 0) como se muestra en la Figura 2.3.

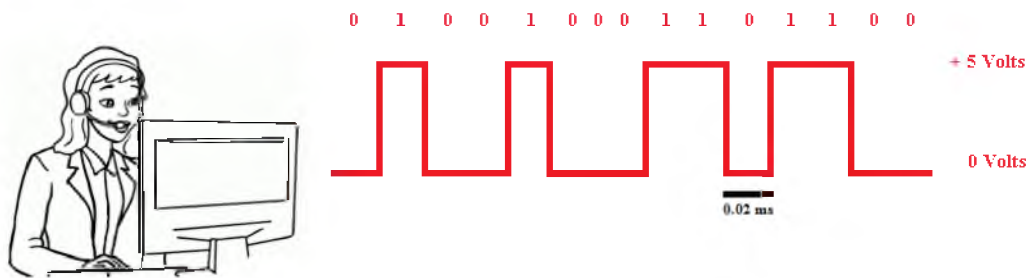


Figura 2.3: Representación de una señal digital.

Las señales digitales tienen un número de ventajas con respecto a las analógicas, es decir, pueden utilizarse desde circuitos digitales relativamente económicos para sus manejo, al igual que pueden emplearse para mantener la privacidad mediante el uso de encriptación de datos. Con las transmisiones digitales es posible obtener un rango dinámico mayor que con las analógicas (la diferencia entre el valor más grande y el más pequeño). Los datos de fuentes de voz y video pueden unirse y ser transmitidos sobre un sistema digital común de transmisión, el ruido no se acumula, como en el caso de los analógicos y pueden usarse repetidores en sistemas de larga distancia. Los errores detectados en los datos en la señal recibida pueden ser muchos y estos pueden corregirse mediante el uso de codificación.

## 2.3. Sensores

Uno de los principales elementos que se debe tener en cuenta en un sistema de AD son los sensores. Estos son los componentes que conectan al sistema AD con su entorno físico. La función de los sensores es obtener señales eléctricas en respuesta a magnitudes de entrada no eléctricas, es decir, se encargan de realizar la medida de un fenómeno físico. Por ejemplo la temperatura de un invernadero, la intensidad de luz en un salón, entre otros.

Los sensores también pueden ser llamados transductores, ya que convierte la señal de un fenómeno físico a una señal eléctrica para ser medida. En la actualidad existen distintos tipos de sensores, como se muestra en la Figura 2.4.

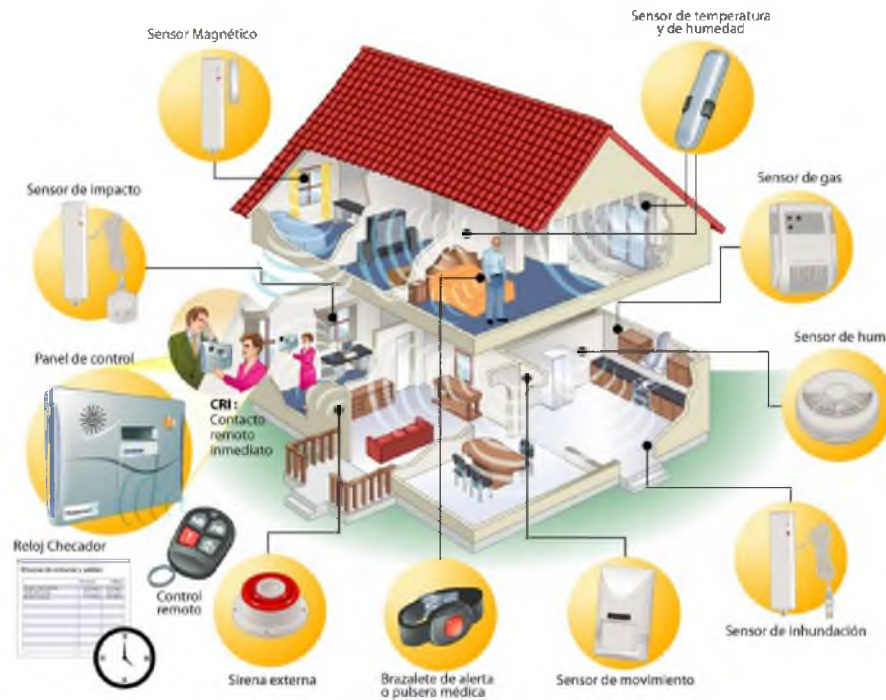


Figura 2.4: Tipos de sensores

Entre la gran diversidad de sensores, en este trabajo nuestro interés son los sensores de luz y el de temperatura.

### 2.3.1. Sensores de luz (LDR)

Existen diferentes tipos sensores de luz, entre los más usados se encuentran las llamadas comúnmente fotorresistencia. Estas son utilizadas para el control

de iluminación en sistemas de alumbrado, público o doméstico, como sensores auxiliares en sistemas fotográficos y pantallas, así como de sensores de presencia.

El funcionamiento de estas fotorresistencias es el siguiente. Cuando se añade suficiente energía por cualquier medio a un material semiconductor los electrones de valencia escapan de sus átomos y se convierten en electrones libres. La idea básica de toda fotorresistencia, es convertir la luz en una señal eléctrica. Para conseguir esto, la fotorresistencia debe fabricarse de un material prácticamente transparente a las longitudes de onda que interesen, con un intervalo de energía menor que la energía del fotón.

En la figura 2.5a se muestra la forma de cómo está constituido un fotoresistor, el cual está fabricado con material fotoconductor depositando por evaporación sobre un sustrato de cerámica. Para complementar el dispositivo se añaden electrodos metálicos y se encierra en una cápsula con una ventana transparente, en la Figura 2.5b se muestra el símbolo eléctrico con el que se representa una fotoresistencia.

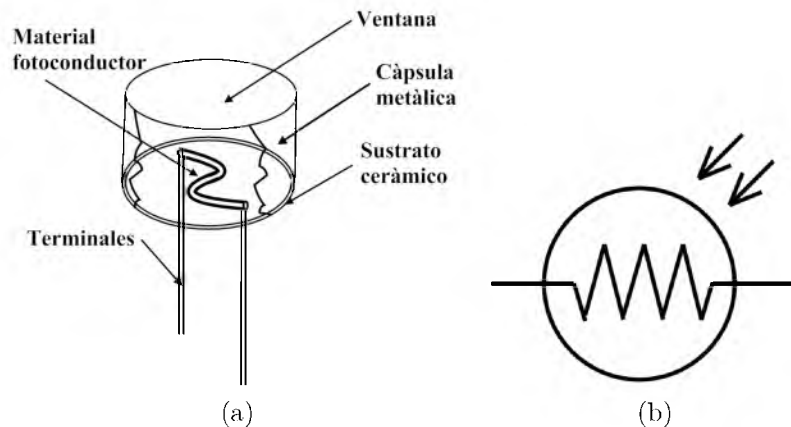


Figura 2.5: En la Figura (a) esquema de una fotoresistencia (b) símbolo.

El comportamiento de una fotorresistencia se caracteriza por medio del tiempo de subida (en ambiente iluminado) y el tiempo de caída (ambiente no iluminado) el tiempo de caída es considerablemente más largo, esto es porque que lleva más tiempo a los electrones en volver a la banda de valencia debido a las imperfecciones cristalinas (Figura 2.6).

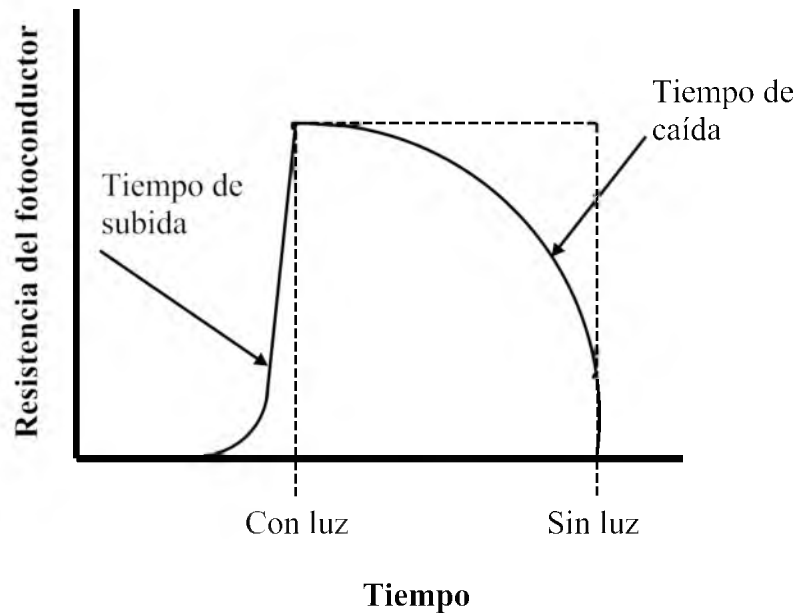


Figura 2.6: Velocidad de respuesta de un fotoconductor típico de una pieza.

Esto es debido a que los dispositivos tienen un efecto de memoria, es decir, su resistencia específica depende de la intensidad y duración de una exposición previa y al tiempo transcurrido desde una exposición anterior.

Los fotoconductores resistivos cuentan con las siguientes ventajas: Alta sensibilidad (debido a la gran superficie), fácil empleo, bajo costo, alta relación resistencia luz-oscuridad. Sin embargo puede presentar efectos de histéresis, respuesta lenta en materiales estables, falta de linealidad entre resistencia e iluminación.

Sus principales características eléctricas son las siguientes:

- Resistencia de oscuridad, valor de la resistencia después de 20 seg. En la oscuridad ( $10^4 \leq R_D \leq 10^9 \Omega$ ).
- La disipación máxima, ( $50mW-1W$ ).
- Resistencia de iluminación (100 lux), ( $10 \leq R_1 \leq 5 \times 10^3 \Omega$ ).
- Voltaje máximo, (600V).
- Tiempo de respuesta.

### 2.3.2. Sensores de temperatura (LM35)

El otro sensor que se abordará en este trabajo es el LM35 el cual proporciona la temperatura en grados centígrados. El LM35 es un circuito integrado cuyo voltaje de salida es linealmente proporcional a la temperatura en grados Celsius, y es equivalente a  $+ 10.0 \text{ mV} / \text{C}^\circ$ . Por ejemplo en las siguientes lecturas se presentan la equivalencia en grados centígrados.

- $+1500\text{mV} = 150\text{C}^\circ$
- $+250\text{mV} = 25\text{C}^\circ$
- $-550\text{mV} = -55\text{C}^\circ$

Este sensor tienen una salida de baja impedancia, salida lineal, y calibración precisa inherente para la lectura. Otras de sus características eléctricas son las siguientes:

- Calibrado directamente en grados Celsius, centígrado [ $\text{C}^\circ$ ]).
- Escala de factor lineal.
- $0.5 \text{ C}^\circ$  de precisión (en  $+25 \text{ C}^\circ$ ).
- Con rango de  $-55^\circ$  a  $+150 \text{ C}^\circ$ .
- Conveniente para aplicaciones remotas.
- Bajo costo debido al ajuste material con el que está fabricado.
- Opera entre 4 y 30 Volts de alimentación.
- Corriente de drenado menos que  $60 \mu\text{A}$ .
- Bajo auto-calentamiento,  $0.8 \text{ C}^\circ$ .
- Baja impedancia de salida,  $0.1 \Omega$  a  $1 \text{ mA}$  carga.
- No linealidad sólo  $\pm 1/4 \text{ C}^\circ$  típico.

## 2.4. Tarjeta de adquisición de datos

En esta sección se hablará de dispositivos o tarjetas AD, los cuales actúan como el medio de comunicación entre una computadora y las señales obtenidas por medio de los sensores. Las tarjetas AD tiene como trabajo principal digitalizar las señales analógicas obtenidas para que la computadora pueda interpretarlas (Figura 2.7). Los tres componentes claves de un dispositivo AD son el circuito de acondicionamiento de señal, convertidor analógico-digital (CAD) y un bus de datos hacia la computadora. Varios dispositivos AD incluyen otras funciones para automatizar sistemas. Por ejemplo, existen convertidores digitales-analógicos (CDA) con salidas analógicas, líneas de E/S digital, contadores/temporizadores entre otras.



Figura 2.7: Tarjeta AD para comunicación con PC.

### **Sistema de acondicionamiento de señales**

Las señales de los sensores pueden contener ruido eléctrico, el cual distorsiona en gran medida la información recabada. El circuito de acondicionamiento manipula una señal de forma tal que la hace apropiada para la entrada de los CAD. Éste circuito puede incluir amplificación, atenuación, filtrado y/o aislamiento, entre otros. Algunos dispositivos AD incluyen acondicionamiento de señales, diseñado para ser usados con tipos específicos de sensores.

### **Convertidor Analógico - Digital**

Otro de los componentes necesarios en los sistemas AD son los CAD's. Las señales analógicas de los sensores varían continuamente con el tiempo. Estas deben convertirse en señales digitales antes de ser manipuladas por un equipo digital, como una computadora. Un CAD es un circuito integrado que proporciona una representación digital de una señal analógica por medio del proceso del muestro.

Este proceso consiste en realizar "muestras" periódicas de la señal a una razón predefinida. Estas muestras son transferidas a una computadora a través de un bus, donde la señal original se reconstruye por medio de un software [2].

### **Bus de comunicación**

Finalmente, el último elemento de este sistema es el bus de la computadora. Los dispositivos AD se conectan a una computadora a través de una ranura o puerto. El bus de la computadora sirve como la interfaz de comunicación entre el dispositivo AD y la computadora para transmitir instrucciones y procesar los datos medidos. Existen distintos tipos de buses y cada uno de ellos ofrece diferentes ventajas para diferentes tipos de aplicaciones, por ejemplo USB, PCI, PCI Express, entre otros. Recientemente, los dispositivos AD están disponibles para comunicación inalámbrica Wi-Fi bajo el estándar 802.11, como se muestra en la Figura 2.8.



Figura 2.8: Medio de comunicación para la transferencia de información de los sistemas AD.

## **3. COMUNICACIÓN INALÁMBRICA**

### **3.1. Introducción**

En esta sección se hablará un poco la historia de las comunicaciones inalámbricas, así como su teoría, seguido del concepto de redes inalámbricas y sus distintos tipos. Se hará mención de los protocolos existentes para enviar y recibir información desde cualquier sistema con comunicación inalámbrica.

En los últimos años se ha intensificado el uso de redes inalámbricas, esto se debe a varias razones, por ejemplo el estilo de vida actual, la necesidad de mantener conectividad a redes locales o Internet de forma constante, el soporte a la movilidad, mayor flexibilidad.

Las comunicaciones digitales han revolucionado la industria de la computación desde sus orígenes. En 1971 un grupo de investigadores en la Universidad de Hawaii, crearon el primer sistema de conmutación de paquetes mediante una red de comunicación por radio, dicha red se llamó ALOHA. Ésta fue la primera red de área local inalámbrica (WLAN). Estaba formada por 7 computadoras situadas en distintas islas que se podían comunicar con un ordenador central para realizar cálculos. Un año después ALOHA se conectó mediante ARPANET. Esta red de computadoras fue creada por el Departamento de Defensa de los EE.UU. como medio de comunicación para los diferentes organismos del país [7].

La industria de la comunicación de datos ha crecido a una velocidad astronómica. En consecuencia, la necesidad de proporcionar comunicación entre distintos sistemas de comunicación, también ha aumentado.

Uno de los primeros problemas que tuvieron y que tiene todo nuevo tipo de red inventada fue el control de acceso al medio, es decir, el protocolo a seguir para evitar que las distintas estaciones de trabajo traslapen sus mensajes entre sí. Por lo tanto, para asegurar un intercambio de información ordenada, entre dos o más sistemas de comunicación, un consorcio de organizaciones, fabricantes y usuarios se reúnen regularmente para establecer las guías y estándares [17].

### **3.2. Tipos de redes**

De forma general, las redes computacionales se pueden clasificar según su extensión geográfica (Figura 3.1).

- WAN (World Area Network), red de área local.
- MAN (Metropolitan Area Network), red de área metropolitana.
- LAN (Local Area Network), red de área local.
- PAN (Personal Area Network), red de área personal.

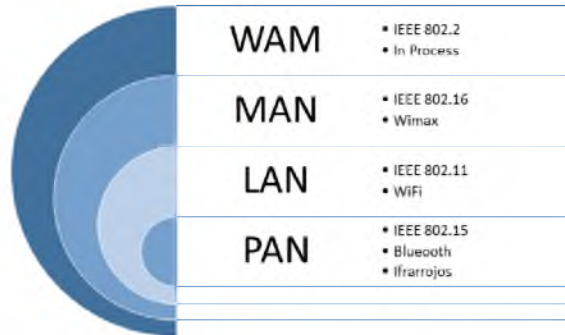


Figura 3.1: Clasificación de las redes inalámbricas.

Se sabe que el término inalámbrico hace referencia a la tecnología sin cables, es decir, que permite conectar varias computadoras o dispositivos entre sí. Las conexiones inalámbricas confieren a los grupos de usuarios flexibilidad y prestaciones muy avanzadas en sus comunicaciones.

En la primera y segunda categoría WAN/MAN, se encuentran ubicadas las redes que cubren desde decenas hasta miles de kilómetros. Las redes de este tipo tienen las computadoras situadas en lugares distantes, como diferentes ciudades, provincias, regiones, países, continentes o, simplemente, edificios muy lejanos dentro de una misma zona. Por lo general, utilizan la línea telefónica para conectarse entre sí, aprovechando la infraestructura lograda por Internet. No obstante, las empresas de mayor magnitud unen las computadoras que forman parte de la red mediante una conexión satelital, por ejemplo para conectar varias sucursales situadas en diferentes países (Figura 3.2).

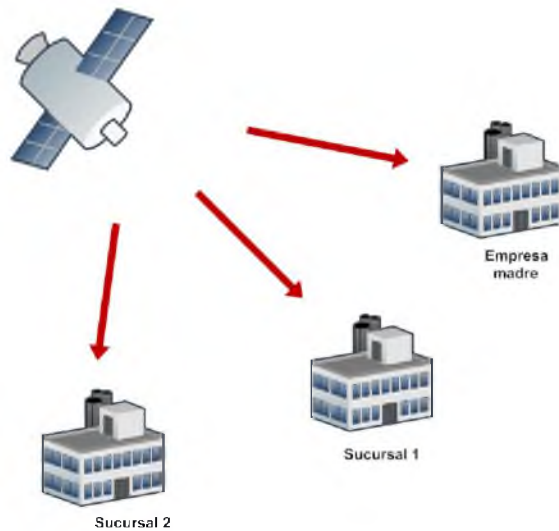


Figura 3.2: Esquema de la red WAN (World Area Network).

En la tercera categoría se encuentran las Redes de área local (LAN). Estas comprenden de varios metros hasta decenas de metros. Son aquéllas que tienen cerca las computadoras: en la misma habitación, en diferentes pisos de un edificio o en edificios muy cercanos (Figura 3.3). Las redes de área local proveen una excelente velocidad de transferencia, que va desde los 10 hasta los 1000 Mbps. Esto se debe a la corta distancia existente entre las computadoras, lo cual evita interferencias.



Figura 3.3: Esquema de la red LAN (Local Area Network).

En la última categoría la Red de área personal (PAN). Básicamente es una red integrada por todos los dispositivos en el entorno local y cercano de su usuario, es decir que la componen todos los aparatos que están cerca del mismo. La principal característica de este tipo de red es que permite al usuario establecer una comunicación con sus dispositivos de forma sencilla, práctica y veloz (Figura 3.4). Hoy en día son varias las tecnologías que permiten la creación de una red de área personal, entre las cuales se destacan el Bluetooth y los sistemas que utilizan la transmisión de infrarrojos [14].

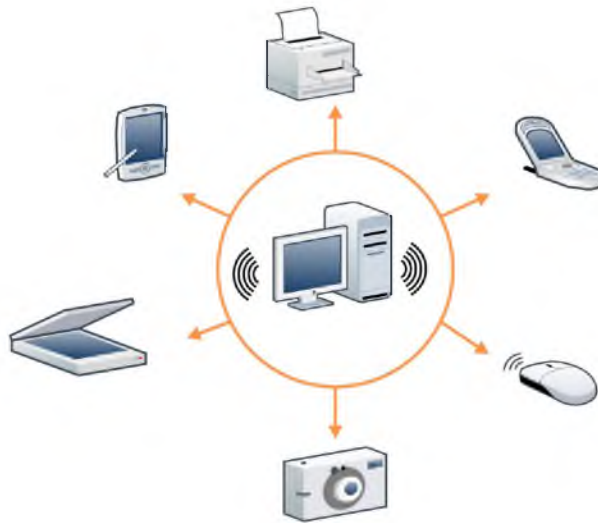


Figura 3.4: Ejemplo de una red de área personal PAN (Personal Area Network).

Esta forma especial de transmisión, mediante un haz de luz, modula la información y se envía de un transmisor a un receptor a una distancia relativamente corta (Figura 3.5). La radiación infrarroja (IR) es la misma tecnología usada para controlar una televisión con un mando a distancia.

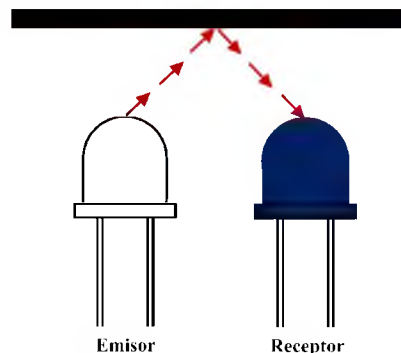


Figura 3.5: Configuración de transmisión y recepción del infrarrojo.

Estas redes son muy limitadas, dado su corto alcance, la necesidad de “visión sin obstáculos” entre los dispositivos que se comunican y su baja velocidad (hasta 115 kbps, 4Mbps). Se encuentran principalmente en ordenadores portátiles, PDAs (Agendas electrónicas personales), teléfonos móviles y algunas impresoras.

Por su parte el Bluetooth, está orientado a la conectividad inalámbrica entre dispositivos: computadoras de escritorios, PDAs, teléfonos móviles, auriculares, inclusive impresoras y otras más.

Este es un estándar libre, lo que simplifica su uso para diseñar y producir nuevos productos, que beneficien este tipo de conectividad. Los dispositivos de radio que soportan esta tecnología no requieren de licencia y deben tener un espectro de 2.4 Ghz para asegurar la compatibilidad en todo el mundo. Las conexiones son uno a uno, con un rango máximo de 10 metros, aunque el uso de repeticiones se puede lograr un alcance de hasta 100 metros con algo de distorsión.

Las configuraciones de las redes pueden ser de muy diversos tipos y tan simples o complejas como sea necesario. Por ejemplo la red llamada red peer to peer (punto a punto), que consiste en dos computadoras equipadas con tarjetas adaptadoras wireless, de manera tal que pueden hacer funcionar una red independiente (siempre que se encuentren dentro del área de cobertura de las tarjetas adaptadoras). Cada computadora poseerá acceso únicamente a los recursos de la otra Figura 3.6.

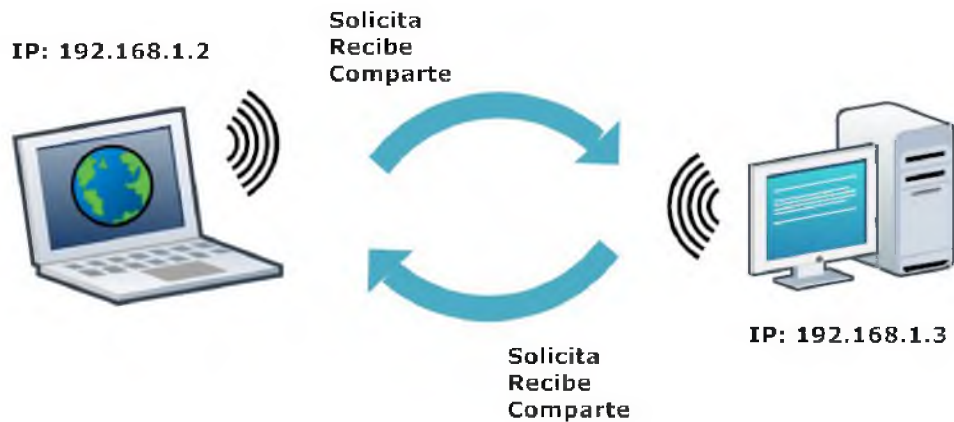


Figura 3.6: Ejemplo de conexión punto a punto.

Por medio de la instalación de un punto de acceso (Access Point), es posible duplicar la distancia a la cual los dispositivos pueden comunicarse, ya que éstos actúan como repetidores de la señal. Desde que el Access Point se conecta a la red, cualquier cliente tiene acceso a los recursos del servidor. Además, este dispositivo

gestiona el tráfico de la red entre las terminales más próximas. Cada Access Point puede servir a varias máquinas, según el tipo y el número de transmisiones que se realicen. En zonas grandes, como por ejemplo un campus universitario o un edificio, es probable que se necesite la instalación de más de uno de éstos dispositivos (repetiendo la señal). El objetivo es cubrir un área definida con células que traslapen sus áreas, de forma tal que los clientes puedan moverse sin cortes entre los grupos de Access Point de alcance. Esto es llamado roaming Figura 3.7.

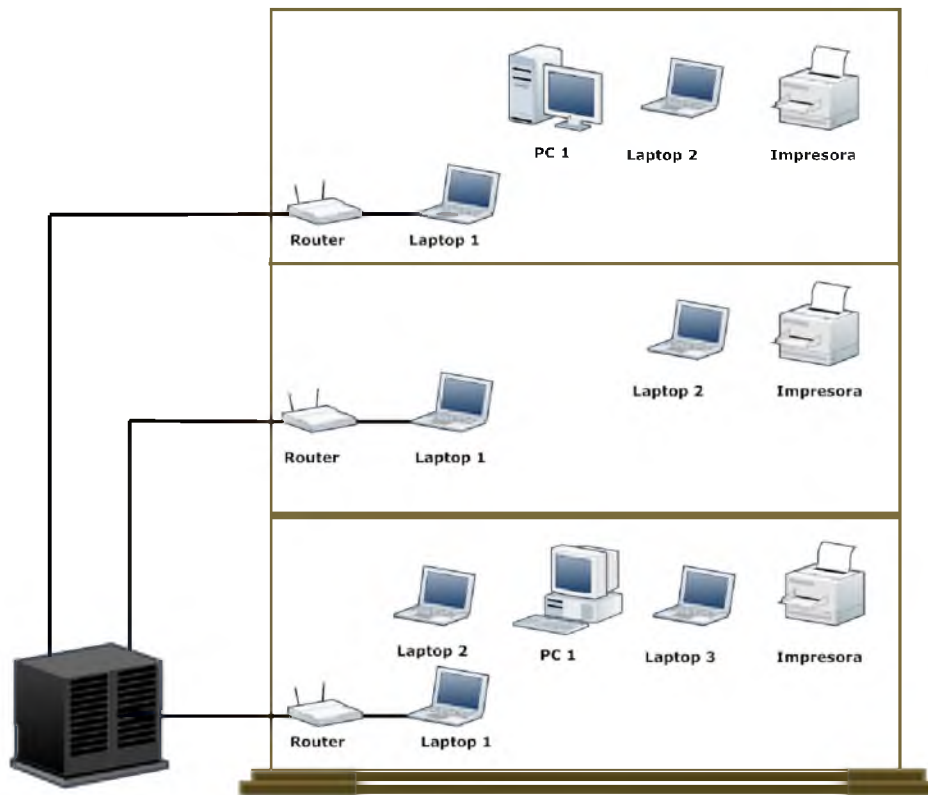


Figura 3.7: Ejemplo de roaming entre varios access points.

### 3.3. Protocolos de redes inalámbricas

Uno de los problemas que tuvo la primera red de comunicación por radio fue el control de acceso al medio. Esto es que otras redes no interfirieran en su comunicación. Para esto se desarrollaron protocolos de comunicación.

Los protocolos de comunicación son un conjunto de reglas y procedimientos que proporcionan una técnica uniforme para gestionar un intercambio de información. Estas reglas y procedimientos proveen la administración, asignación y control de

los recursos involucrados en el proceso, así mismo, establecen métodos para evitar y/o resolver problemas ocurridos en cualquiera de los elementos que intervienen en el proceso.

El Protocolo TCP (“Transmission Control Protocol”) y el Protocolo IP (“Internet Protocol”) conforman un conjunto de protocolos desarrollados para permitir, que una computadora o usuario comparta recursos a través de una red o red de redes (Internet). Este conjunto de protocolos se desarrollaron teniendo como base la red ARPANET.

Existen protocolos de tipo “Internet”, pero TCP e IP son los protocolos más conocidos y comúnmente se utiliza el término compuesto TCP/IP para denominarlos. En general, estos protocolos ofrecen servicios de transferencia de archivos (FTP), conexión remota con otra computadora en la red (TELNET), correo electrónico, entre otras.

En ocasiones los usuarios desean utilizar algunos servicios especiales proporcionados por computadoras, tales como sistemas de archivos de red, impresión y graficación remotas, , esto ha llevado a la definición del modelo “cliente/servidor”.

Un **servidor** es un sistema que provee un servicio específico para el resto de la red; mientras que un **cliente** es otro sistema que utiliza ese servicio. Cabe señalar que el servidor y el cliente no necesariamente están en diferentes computadoras; ellos pueden ser diferentes programas en una misma computadora.

### 3.4. Comunicación mediante sockets

Hoy en día no se puede plantear la existencia de alguna aplicación que no utilice un mecanismo de comunicación. Esto se debe a que la aplicación ha sido desarrollada como un conjunto de procesos que se comunican entre ellos, o bien porque la aplicación está en red y distribuida en diversos equipos donde sus componentes interactúan de una forma ordenada, por ejemplo con un modelo cliente-servidor Figura 3.8 .

Las aplicaciones web que consultamos a diario y las aplicaciones que se ejecutan en los dispositivos móviles, hasta las que se ejecutan en los microcontroladores de los electrodomésticos, usan primitivas de comunicación para sincronizar e intercambiar información o reproducir contenidos multimedia.

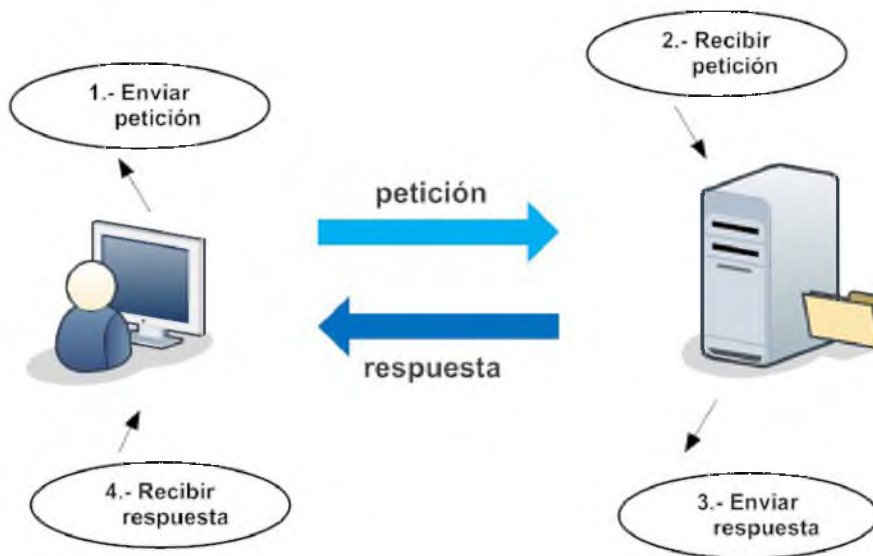


Figura 3.8: Modelo cliente-servidor

Todo mecanismo de comunicación tiene que considerar la fiabilidad en el envío de datos, es decir, que asegure que cualquier información enviada llegará a su destino, o en caso contrario, notificará que no ha podido enviarla. Un mecanismo de comunicación también puede garantizar la re-emisión de la información y definir la topología del proceso de comunicación, es decir, si los mensajes se envían uno a uno (unicast o de igual a igual), uno a muchos (multicasting o broadcasting) o muchos a uno (cliente-servidor).

Para la realización de estas comunicaciones se utilizan puntos de acceso o socket's. Un socket es un punto de acceso a los servicios de comunicación en el ámbito de transporte. Cada socket tiene asociada una dirección que lo identifica. Conociéndola, se puede establecer una comunicación con éste para que actúe como extremo de un canal bidireccional.

Los sockets representan la implementación del paradigma de paso de mensajes. La mayoría de mecanismos de comunicación usan los sockets con fin de la transmisión de datos. Por ejemplo, la implementación de los servicios web con un protocolo de acceso simple a objetos (por sus siglas en inglés, SOAP) o de un método de invocación remota (RMI) usan un socket para transmitir la información.

La biblioteca de sockets se caracteriza por ser simple y ofrecer las funcionalidades básicas para la creación del socket, así como la lectura y escritura a través de él. Existen varias implementaciones de esta biblioteca, algunos ejemplos son, los Berkeley Sockets, que ofrece el núcleo de Linux, WinSock es otra implementación

usada por los sistemas operativos de Microsoft basada en los Berkeley Sockets.

A continuación se presentan ejemplos de los métodos disponibles en la biblioteca de Berkeley Sockets:

- **socket():** crea un conector de un cierto tipo.
- **bind():** usado en el lado del servidor para asociar el conector a una dirección y un puerto.
- **listen():** utilizado en la banda servidor para poner el conector en estado de LISTEN.
- **connect():** usado en el cliente para conectarse a una dirección y puerto del servidor. Al finalizar la llamada, se establece la conexión.
- **accept():** utilizado en el servidor para aceptar una conexión.
- **send() y recv(), o write() y read(), o recvfrom() y sendto():** se usan para recibir y leer datos de un conector.
- **close():** se cierra la conexión (en caso de ser TCP). Se liberan los recursos.
- **gethostbyname() y gethostbyaddr():** se usan para resolver nombres y direcciones.
- **select():** se usa para seleccionar los conectores que están preparados para lectura/escritura o con errores.
- **poll():** se usa para verificar el estado del conector.

La Figura 3.9 ilustra el ciclo de vida de un conector TCP. Una vez que el conector se ha creado en el servidor se vincula a una dirección y un puerto. Este se mantiene a la espera de peticiones por parte del cliente. Cuando el cliente invoca el **connect**, el servidor acepta la petición y se establece la comunicación.

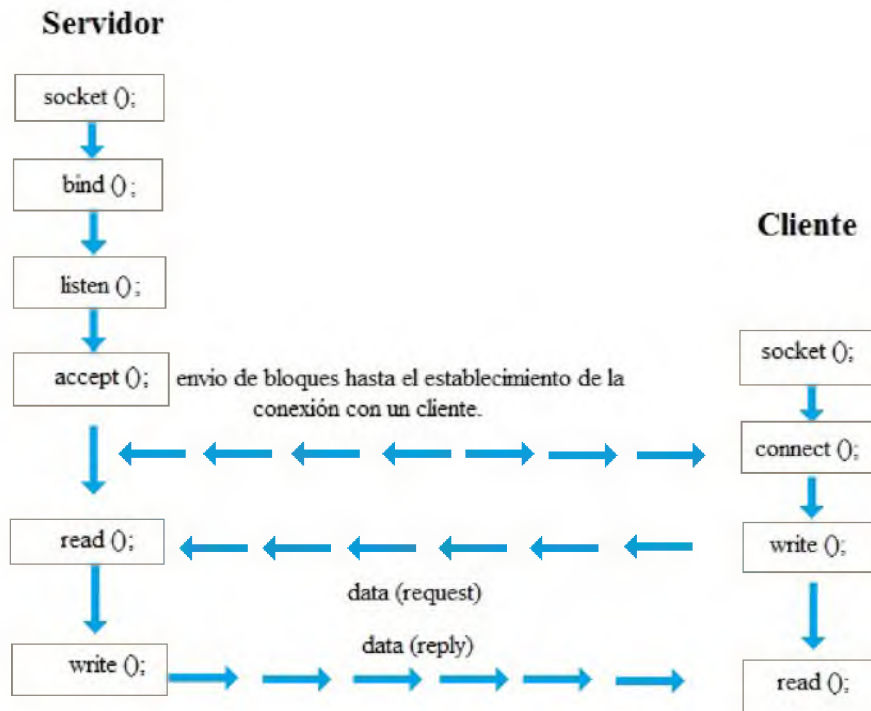


Figura 3.9: Ciclo de vida de un conector TCP.

Los sockets se encargan de ofrecer mecanismos de seguridad. Por ello, tienen que ser las capas superiores las que se encarguen de proteger los datos [5].

### 3.5. Servicio Web (Red informática mundial)

El éxito de Internet está absolutamente ligado a la web. Tanto que hoy en día para la inmensa mayoría de las personas es indistinguible que es la web y que es el Internet. Todo se hace en la web, es decir la web ofrece todo tipo de servicios. Esto no ha sido así siempre, inicialmente Internet no tenía web, solo contenía servicios como correo electrónico, transmisión de archivos, grupos de noticias y todos ellos se manejaban de forma tan incómoda que solo los profesionales de la informática podían utilizar Internet.

Todo cambio cuando Tim Bernes Lee ideó las páginas web. El éxito de Internet se disparó gracias a la facilidad de manejo que ofreció la web. Hoy en día se llama servicios web 1.0 los servicios que ofrecían los sitios web clásicos. En realidad la inmensa mayoría de sitios actuales siguen siendo web 1.0 puesto que solo utilizan

tecnologías básicas. La realidad es que web 1.0 se define como lo que no es web 2.0, se trata de la forma de denominar a las páginas web que ofrecen servicios orientados al usuario. Las páginas web tradicionales ofrecían la misma información para todos los usuarios, ahora varían para cada usuario, le permiten un manejo más rico e incluso le hacen participe del contenido [3].

En los inicios la web era simplemente textos e imágenes junto con los hipervínculos que permitan saltar hacia otros contenidos. Ahora el tipo de contenido es mucho más variado: animaciones, video, sonido, juegos, aplicaciones, entre otros. Es difícil definir, pero sin duda hace referencia a un tipo de servicios web muy concretos. Podemos decir que hay tres pilares que conforman las páginas web 2.0:

- Aplicaciones Ricas de Internet (RIA). Son aquellas páginas web que ofrecen servicios que las asemejan con las aplicaciones de escritorio. Estas aplicaciones requieren que el usuario instale plugins, java o flash para poder utilizarlas pero hoy en día con AJAX, basta con el propio navegador.
- Arquitectura Orientada al Servicio (SOA). Es la tecnología que permite diseñar aplicaciones basándose en peticiones a un determinado servicio. De esta forma se puede crear pequeños elementos software muy reutilizables y además independientes del lenguaje con el que fueron creados.
- Web Social. Es la parte más evidente y entendible de la web 2.0 ya que el usuario posee una interacción mayor en la web siendo participe de lo que en ella ocurre.

Los servidores web son los encargados de recibir las peticiones referidas a páginas o elementos de la web a través del protocolo **http** alojado en un ordenador servidor.

El navegador pide al servidor el recurso que desea el usuario, para finalmente recibir dicho recurso (si fue válida la petición) y traducir si es necesario a su forma legible para el usuario (la traducción de HTML la hace el navegador).

Los servidores de aplicaciones tienen la capacidad de almacenar y gestionar aplicaciones web (servicio al que los usuarios acceden).

Este tipo de servidores no solo atienden para atender peticiones **http**, además son capaces de entender instrucciones de lenguajes avanzados de la web y traducirlas o acceder a recursos de otros servidores. Este proceso se hace transparente al usuario, es decir el usuario pide el servicio a través de su navegador y el servidor de aplicaciones es quien atiende la petición e interpreta el código a fin de traducir

y mostrar al usuario el resultado de forma entendible por su navegador (es decir en formato html) [3].

Esta forma de trabajar de un servidor de aplicaciones, es conocida como arquitectura de tres capas (a veces se habla de más capas). La primera capa esta conectada dentro del navegador el cual disponer de todos los componentes necesarios para traducir el código del lado del cliente (HTML, JavaScript, CSS, Flash, entre otras). La segunda capa la conforma el servidor de aplicaciones quien traduce el código en el lado del servidor (JSP, PHP, Ruby on Rails, Cold Fussion) y convertirlo al formato entendible por el navegador. La tercera capa, son los servicios a los que accede el servidor de aplicaciones para ejecutar la aplicación (por ejemplo el acceso a la base de datos) véase la Figura 3.10.

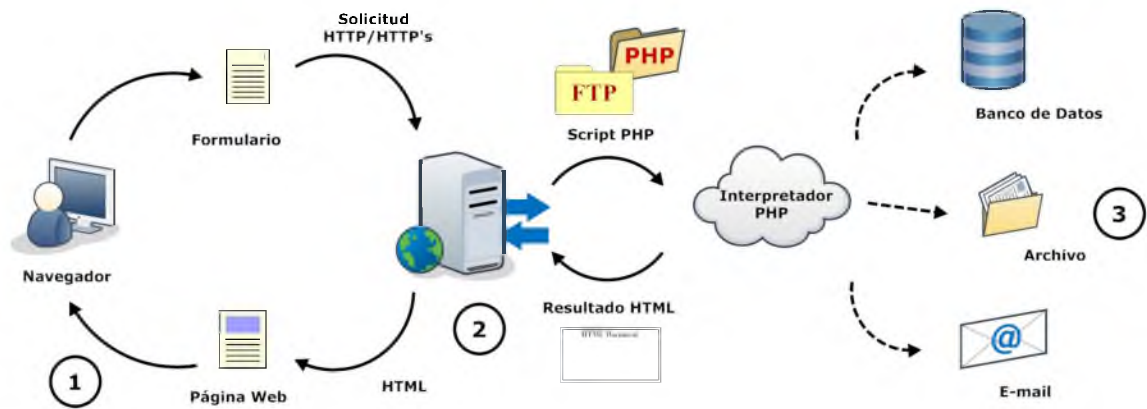


Figura 3.10: Funcionamiento de un servidor de aplicaciones PHP.

Indudablemente Apache es el servidor web más popular de la actualidad. Abarca el 70 % de todos los servidores web instalados. Se trata de un software de código abierto que utiliza una licencia de tipo Apache License que es una variante de la licencia GPL de Linux. Eso significa que se puede distribuir sin problemas e incluso mejorar. Dispone de multitud de módulos que convierten a Apache en un servidor capaz de gestionar todo tipo de aplicaciones, lo que también le convierte en el servidor más popular de la actualidad; por ejemplo dispone de módulos para:

- Implementar SSL. Protocolo de seguridad en la transferencia de información
- Enlace con el servidor Tomcat , para implementar aplicaciones Java de servidor.
- Módulo para Perl

- Módulo para PHP
- Módulo para Python.
- etc.

### **3.6. Aplicaciones con comunicación inalámbrica.**

En la actualidad se utilizan distintos tipos de tarjetas de adquisición de datos. Estas son de utilidad en los casos que se requiere adquirir y transmitir todo tipo de información. A continuación se muestran algunas aplicaciones que ocupan el módulo WiFly, del que se hablará más adelante, o tarjetas con tecnología Wi-Fi similares.

#### **3.6.1. Plataforma domótica para la interconexión de objetos físicos a Internet empleando conectividad WiFi**

La plataforma domótica (Figura 4.14) está desarrollada con un enfoque didáctico. Está se llevó a cabo por alumnos de las carreras de ingeniería en Telemática, software y los posgrados en Ciencias Computacionales y de Tecnologías de la Información de la Facultad de Telemática de la Universidad de Colima [4]. Este prototipo permite la interconexión de los objetos físicos con el Internet para su control. El prototipo está compuesto por los siguientes elementos:

- Interfaz gráfica de usuario GUI.
- Sistema de comunicación y control WiFly.
- Elementos domóticos a controlar como luminarias y contactos.

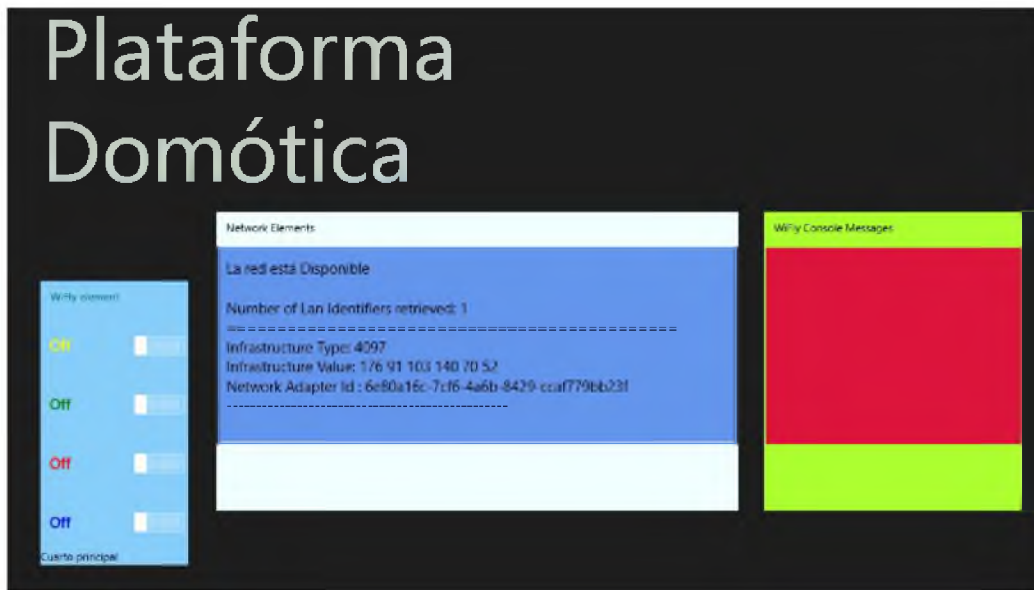


Figura 3.11: Interfaz Gráfica de Usuario (GUI) del prototipo de aplicación de plataforma domótica.

La plataforma domótica utiliza como medio de comunicación el estándar WiFi (802.11.2), controlado por un dispositivo electrónico mediante el módulo RN-XV de Roving Networks. Este módulo permite configurar al dispositivo para que tenga conexión a la red, asignándole una IP, máscara de red, puerta de enlace, y DNS, lo que posibilita a la plataforma la conexión a Internet. Está controla cuatro puertos de entrada-salida, activa un relevador que cierra el circuito para activar la carga o abriéndolo para desactivarla. De esta manera es posible controlar las luminarias y activar/desactivar la red de energía eléctrica en cada habitación Figura 3.12 [4].

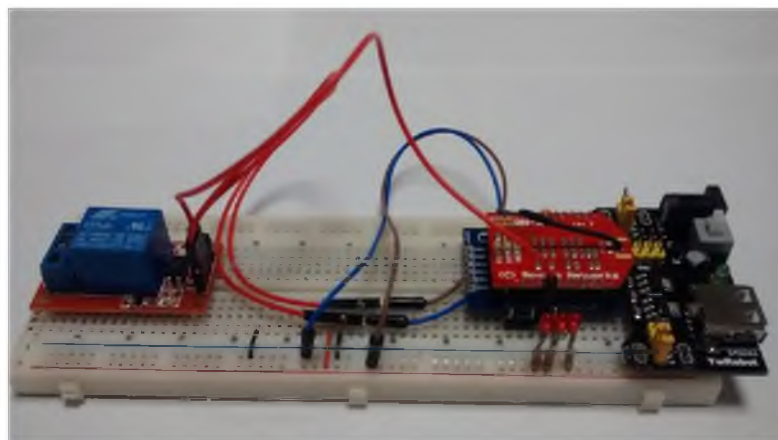


Figura 3.12: Módulo de Comunicación y Control WiFi

### 3.6.2. Robot auxiliar de limpieza de suelos doméstico

El robot de la Figura 3.13 consiste en auxiliar en la limpieza de suelos. Este se encarga de limpiar el polvo y pelusas presentes en la vivienda. El robot dispone de libertad de movimiento para esquivar obstáculos que se encuentren a su paso, gracias a sensores de proximidad y de contacto directo. Durante la ejecución de sus tareas, el robot estará enviando, en tiempo real, sus movimientos a un servidor web, a través del módulo WiFly. Si la conexión con el servidor falla por algún motivo, el sistema dispone de un mecanismo de seguridad que reinicia la conexión [8].

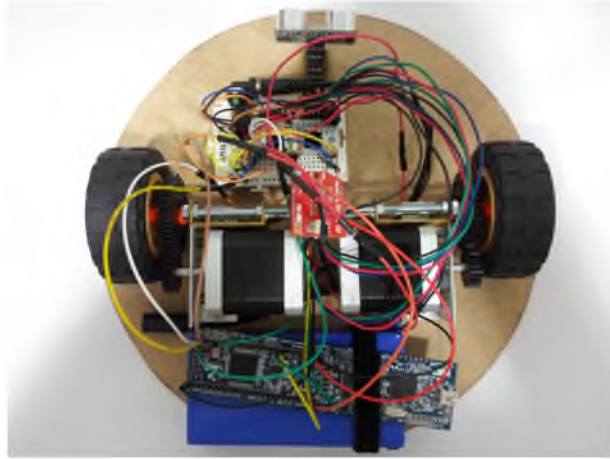


Figura 3.13: Prototipo ensamblado del robot auxiliar de limpieza de suelos doméstico.

La interfaz web esta formada por dos partes, un servidor con un puerto TCP y una GUI simple para visualizar por parte del usuario.

### 3.6.3. Sistema de telemetría para implantarse en prototipos móviles para el monitoreo de sitios remotos

En este trabajo se desarrolló un sistema funcional de telemetría, capaz de monitorear con una cámara de video, medir temperaturas y concentración de gases dañinos para el ser humano. Este sistema fue implementado en un prototipo móvil con la finalidad de realizar monitores en sitios remotos Figura 3.14 [10].



Figura 3.14: Imagen tomada a Gabinete de Sistema Telemétrico en terreno para pruebas

Se utilizó una plataforma universal de WiFi 2.4 Ghz (802.11b/g) para lograr acceder a diferentes muros y cuerpos no acuosos. El sistema de telemetría es usado para medir humedad en el ambiente, censar monóxido de carbono CO, gas metano y temperatura.

#### **3.6.4. Comunicación Inalámbrica Digital con un Controlador Lógico Programable para el Control Industrial en Zonas de Alto Riesgo**

El proyecto utiliza la tecnología inalámbrica Wireless Personal Área Network (WPAN) mediante protocolo ZigBee para la comunicación entre sensores, un Controlador Lógico Programable (PLC) y los actuadores, que en conjunto permiten la automatización de máquinas o procesos. Este sistema inalámbrico permite la comunicación entre 16 señales eléctricas digitales de entrada y/o salida. De forma que pueda alcanzar distancias de comunicación radial de 40 m en ambiente cerrado y de 100 m en espacio abierto. El sistema opera en la banda libre de 2.4 Gigahertz lo que lo hace insensible a señales electromagnéticas producidas en ambientes industriales (Figura 3.15).

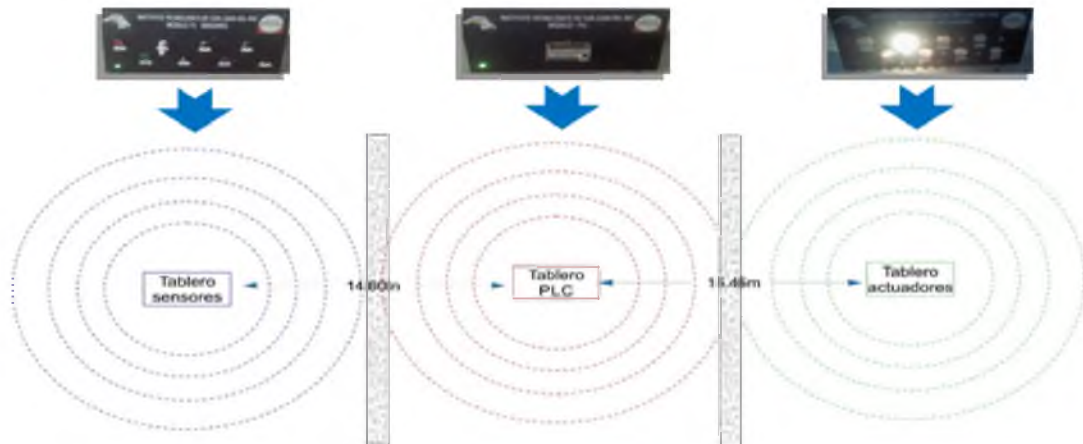


Figura 3.15: Prueba de operación con obstáculos en espacio cerrado

Los dispositivos utilizados para la comunicación inalámbrica son los módulos XBee de la marca DIGI INTERNATIONAL®. Las aplicaciones sobre el uso de las tecnologías inalámbricas abren un abanico de posibilidades para ser implementados en diferentes campos de la automatización industrial, sobre todo, en aquellas zonas que representen un riesgo para el operador [9].

## 4. IMPLEMENTACIÓN DE UN SISTEMA DE ADQUISICIÓN DE DATOS

### 4.1. Introducción

En esta sección se describe, la construcción del sistema de adquisición de datos con tecnología WiFi.

Se diseñó una placa base para la interconexión Arduino WiFly (BIAW), de manera que tuviera las conexiones apropiadas para trabajar con el módulo WiFly y la tarjeta de desarrollo *Arduino Mega*<sup>®</sup>. Esta placa se realizó con el software *EAGLE*<sup>®</sup> Freeware en su versión 6.

Concluida la construcción de la placa BIAW, se realizaron las pruebas necesarias a fin de comprobar el funcionamiento de la BIAW, el *Arduino* y la WiFly en conjunto con los sensores. Los datos obtenidos en estas pruebas se podrán visualizar en la interfaz desarrollada con programación HTML y PHP. En la Figura 4.1 se muestra el diagrama del sistema.

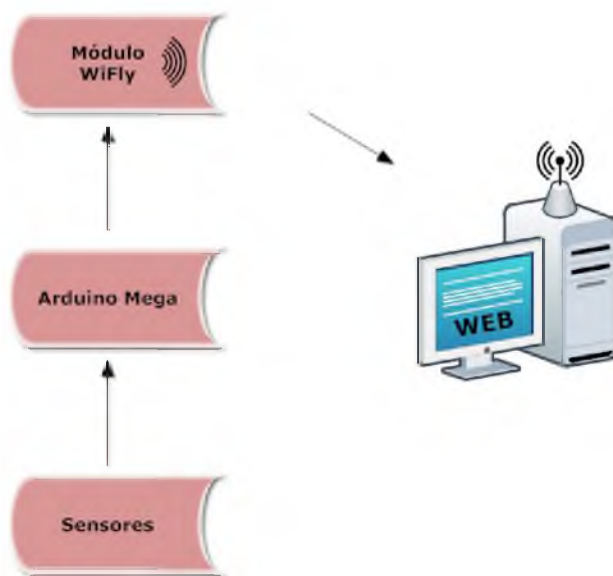


Figura 4.1: Diagrama del sistema de adquisición de datos

### 4.2. Descripción del Módulo WiFly

Como se mencionó anteriormente, el sistema a desarrollar se conectará con una red local por medio de WiFi, por lo que en nuestro caso se utilizó el módulo

WiFly GSX RN-171.

La utilización del módulo reduce notablemente el tamaño y número de componentes utilizados para la realización de cualquier sistema de comunicación de este tipo.

Esta sección se centrará en la descripción de módulo. El dispositivo WiFly es relativamente pequeño (27.7 x 24.38 mm) (Figura 4.2), requiere de una alimentación de 3.3v, los cuales pueden ser provistos por una batería. Cuenta con 20 pines para su conexión y completo funcionamiento, sin embargo para crear una conexión básica a una red inalámbrica solo son necesarios cuatro, alimentación (PWR), transmisión de datos (TX), Recepción de datos (RX) y tierra (GND).

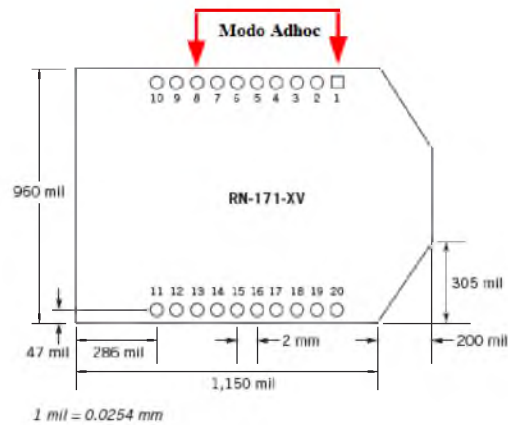


Figura 4.2: Magnitudes del módulo inalámbrico.

El módulo WiFly (Figura 3.12) cuenta con comunicación TCP/IP, 8 MB (Megabyte) de memoria Flash y 128 KB (Kilobyte) de memoria RAM, está certificado como radio transmisor receptor Wi-Fi 2.4 GHz IEEE 802.11b/g. Soporta el modo ADHOC y conexiones de modo infraestructura. Puede ser utilizado en sistemas que manejen WPA2-PSK por Wi-Fi. Con el es posible construir aplicaciones de redes: cliente DHCP, cliente DNS, ARP, ping ICMP, FTP, TELNET, HTTP, UDP, TCP [13].



Figura 4.3: Módulo de comunicación inalámbrico WiFly GSX RN.

El módulo contiene tres leds (verde, amarillo y rojo). En el cuadro 4.1 se da una breve explicación del comportamiento de cada Leds.

Leds	Descripción
Luz verde estática	Conectado a través de la TCP
Parpadeo lento de luz verde	Dirección IP asignada
Parpadeo rápido de luz verde	Dirección IP no asignada
Luz verde apagada	No Asociado
Luz amarilla estática	Ninguna acción
Parpadeo lento de luz amarilla	Ninguna acción
Parpadeo rápido de luz amarilla	Transferencia de dato RX/TX
Luz amarilla apagada	Ninguna actividad
Luz roja estática	Ninguna acción
Parpadeo lento de luz roja	Asociado, Internet no detectado
Parpadeo rápido de luz roja	No asociado
Luz roja apagada	Asociado, internet detectado.

Tabla 4.1: Descripción del funcionamiento de los leds del módulo WiFly,

El módulo WiFly tiene dos modos de entrada, el modo de Datos y el modo Comandos. En el modo Datos, el módulo esta listo para aceptar conexiones entrantes o iniciar conexiones externas. Para configurar los parámetros y/o revisar la configuración actual, el módulo debe estar en el modo Comando (también llamado modo de configuración).

### 4.3. Conexión y configuración del Módulo WiFly

Para la conexión del módulo WiFly se utilizó la tarjeta de desarrollo *Arduino*. (Figura 4.4), este sistema de prototipado rápido contiene un microcontrolador ATmega2560 que dispone de un receptor/transmisor serie de tipo TTL-UART. Esto le permite su comunicación con otros dispositivos. El canal físico de comunicación suele ser USB, sin embargo se pueden utilizar los pines digitales RX y TX de la tarjeta [1].



Figura 4.4: Placa Arduino Mega 2560

El modo de trabajo de la red con la que se va a trabajar es el ADHOC, se diseñó el sistema de adquisición de datos a fin de tener control sobre la configuración inicial del módulo WiFly, para esto en la placa BIAW desarrollada se incluyó un interruptor que facilita la inicialización del módulo en este modo de trabajo (Figura 4.5).

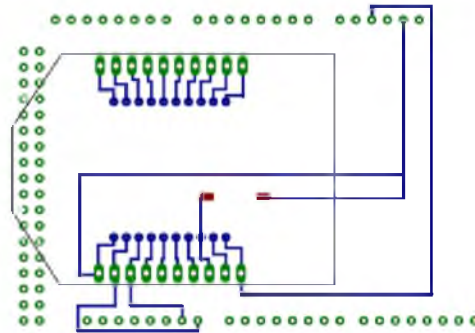


Figura 4.5: Diseño del circuito de la WiFly acoplado al circuito del Arduino Mega.

Para iniciar la comunicación, entre la WiFly y la tarjeta Arduino, se conectaron de forma apropiada los pines de alimentación, tierra y de transmisión (RX y TX). Estas conexiones, físicamente, quedan como se muestran en la Figura 4.6.

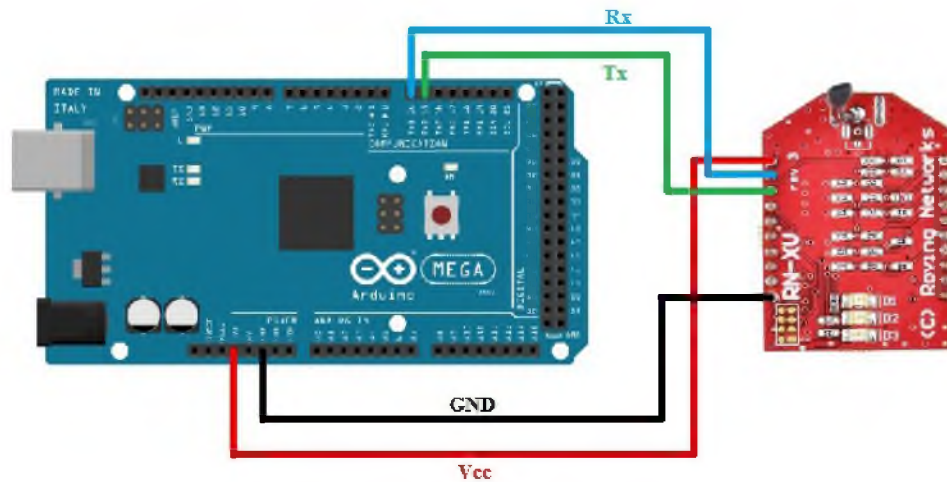


Figura 4.6: Esquema de conexión del WiFly con *ArduinoMega*®.

El montaje físico del módulo con el sistema Arduino se muestra en la Figura 4.7.

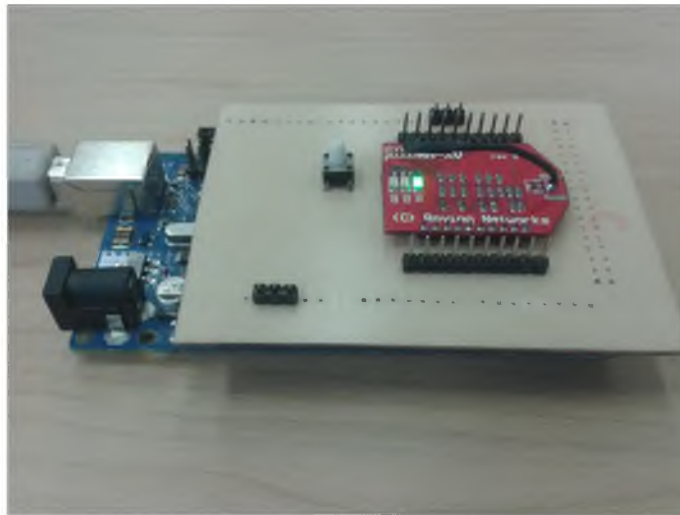


Figura 4.7: Ensamble final del módulo WiFly al Arduino mega.

Al encender el sistema WiFly/Arduino se genera la red ADHOC creando la red llamada **WiFly-GSX-141** (los último dígitos corresponden a los bits finales de la MAC del equipo al cual se encuentra conectado el módulo).

Al conectar la computadora a la red, esta nos arroja la dirección de la WiFly, la cual es 169.254.1.1 con el puerto 2000 (valores por defecto del módulo).

La comunicación debe ser establecida en el sistema operativo. Para modificar parámetros es necesario una terminal que nos permita ejecutar comandos en él. El emulador de terminal utilizado se conoce como *CommOperator* en su versión de evaluación, de esta manera al ingresar la IP y el puerto a la terminal se inicia la configuración de la WiFly. De este modo al ingresar a la red, esta respondería un **\*HELLO\*** indicando el acceso a ella (Figura 4.8).

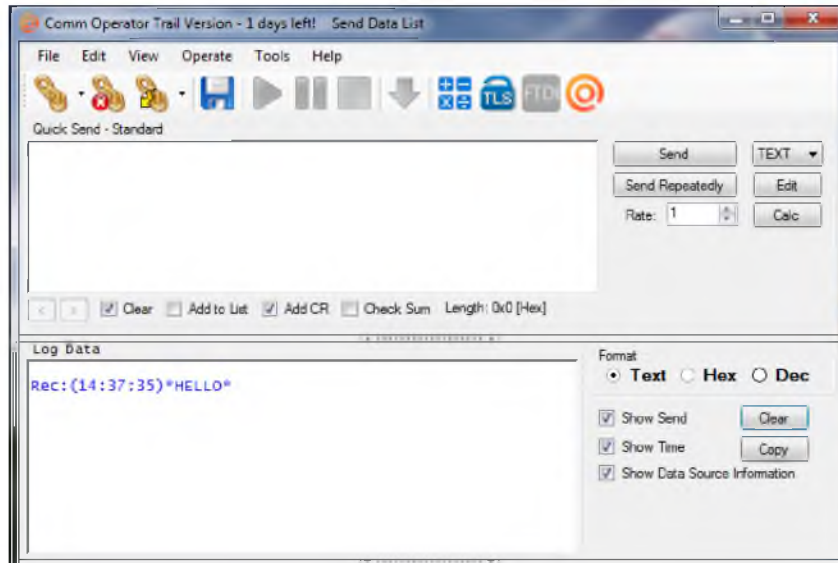


Figura 4.8: Respuesta del módulo al conectarse.

A continuación se configuran los parámetros internos de la WiFly, para esto se debe estar en modo Comandos. Desde la terminal *CommOperator* se introduce los caracteres \$\$\$, como la respuesta del módulo es CMD esto indica que se encuentra en este modo (Figura 4.9).

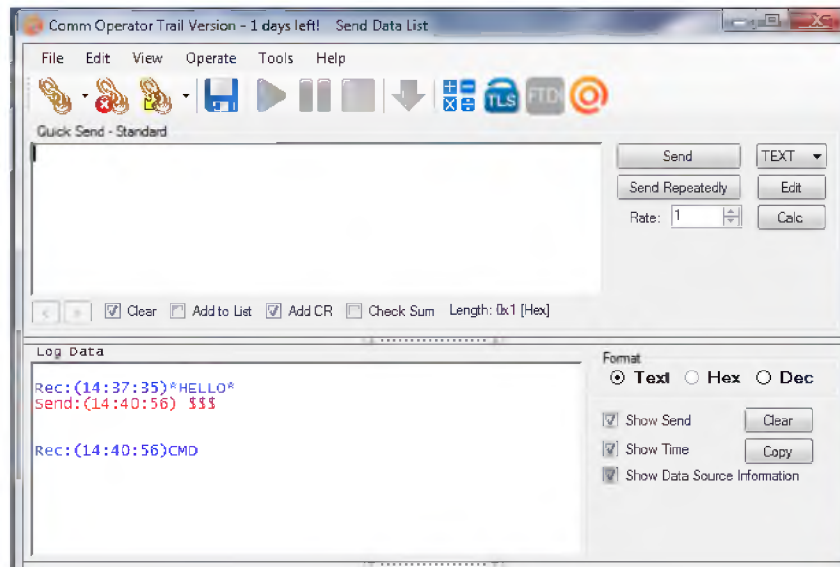


Figura 4.9: Ejecución de comandos.

En secuencia, se ingresa el comando que especifica el tipo de red que será creada (**set wlan join 0**). Aquí el valor de cero indica que se creará una red ADHOC. Para otro tipo de configuración de red ver la tabla 4.2. Cada vez que se ingresa un comando correcto el módulo responderá un **AOK** de lo contrario aparecerá un **EERR** o **-cmd**.

Valor	Modo de Autenticación
0	Open (Default)
1	WEP-128
2	WPA1
3	Mixed WPA1 & WPA2-PSK
4	WPA2-PSK
5	Not Used
6	Adhoc, Join any Adhoc network
8	WPE-64

Tabla 4.2: Tabla de modo de autenticación

Se le asignó un nombre a la red, utilizando el comando **set wlan ssid my\_network**. En este caso el nombre de la red es **my\_network** (Figura 4.10). Una nota importante es que, para el nombre de la red, no se aceptan espacios por lo cual se coloca un guión bajo.

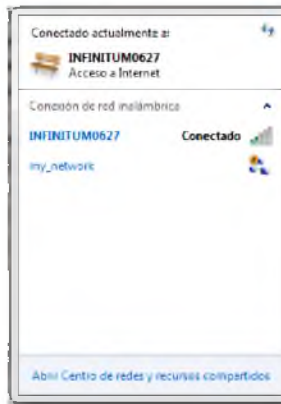


Figura 4.10: Detección de la red creada por la WiFly

A continuación se configuran los canales **wlan** los cuales son del 0 hasta el 13. En este caso se utiliza el canal 1, el cual se asigna de la siguiente manera **set wlan channel 1**. La dirección IP también puede ser cambiada, sin embargo se dejará la misma que trae el módulo. La **netmask** se configura con el comando **set ip netmask 255.255.0.0** y el **DHCP** se deshabilita porque utilizaremos una dirección estática.

Después de haber configurado estos parámetros, se guardan mediante el comando **save**. Para verificar que esto fue realizado correctamente el módulo responderá de manera automática lo siguiente **storing in config**. La configuración concluye ingresando el comando **Reboot**. Este comando reinicia el módulo, y generará la red lista para acceder y trabajar con ella (Figura 4.11).

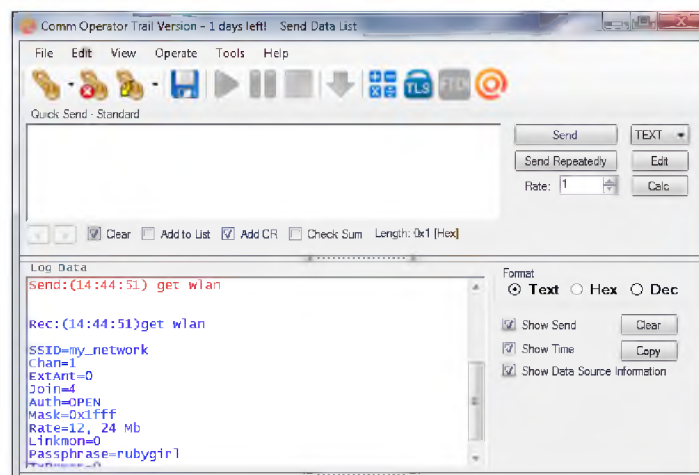


Figura 4.11: Características configuradas por medio de sus comandos.

## 4.4. Pruebas del sistema de adquisición

En la presente sección se describen las pruebas realizadas para verificar el funcionamiento del sistema de adquisición.

Se genera un código para iniciar la comunicación serial, para esto es necesario una variable llamada **wifiSerial** del tipo **SoftwareSerial**. Esta contendrá los pines de transmisión (TX,RX) como parámetros, permitirá la comunicación entre el Arduino y la computadora. Se debe tener en cuenta que si se utilizan los dos pines TX y RX para la comunicación serie, estos no podrán ser usados entonces como entradas/salidas digitales estándar.

Es necesario establecer la velocidad de transmisión para la comunicación serie. Las velocidades de transmisión soportados por Arduino son 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57600 y 115200.

Al ejecutar el código 1 se visualizaron en la terminal *CommOperator* caracteres distorsionados o simplemente la primera letra de la cadena que se enviaba.

Código 1: Comunicación del puerto serial

```
1 #include <SoftwareSerial.h>
2 SoftwareSerial wifiSerial(14,15);
3
4 void setup()
5 {
6   Serial.begin(115200);
7   wifiSerial.begin(9600);
8 }
```

La velocidad adecuada para la comunicación del Arduino y la computadora es de 115200 baudios. Se creó el código 2 teniendo como objetivo enviar la cadena "**Hola Mundo**", de este modo utilizando la velocidad especificada se observó enviaba correctamente la cadena (Figura 4.12).

Código 2: Envió de la cadena Hola Mundo

```
#include <SoftwareSerial.h>
SoftwareSerial wifiSerial(14,15);

void setup()
{
  Serial.begin(115200);
  wifiSerial.begin(9600);
}
void loop()
{
  wifiSerial.println();
  wifiSerial.println("Hola Mundo");
  delay(1000);
}
```

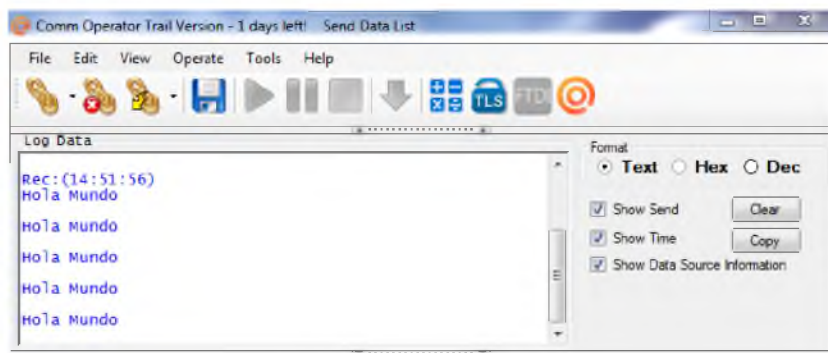


Figura 4.12: Transmitiendo una cadena.

La siguiente prueba consiste en adquirir una señal analógica y transmitirla por el módulo. Para esto se utilizó un potenciómetro de  $5\text{ K}\Omega$  como divisor de voltaje, de manera que el pin analógico (A2) del Arduino se encargará de recibir la señal mandada por el potenciómetro (esta señal será medida en voltaje).

Con este fin se modificando el código 2, se declarará una variable de tipo entero (**int**), la cual será llamada **valorPote** la cual almacenará la señal mandada por el potenciómetro. También se declarará una variable tipo flotante (**float**) llamada **volt**. Esta guardará el valor del voltaje después de realizar la conversión, ya que

el Arduino toma del 0 al 1023 donde 0 es igual a 0 v y 1023 equivale a 5v. En el código 3 se detalla de manera clara la descripción expuesta anteriormente.

Código 3: Adquisición del valor del potenciómetro.

```
1 void loop()  
2 {  
3     valorPot=analogRead(1);  
4     volt=(valorPot*5)/1023;  
5     Serial.println(volt);  
6  
7     wifiSerial.println("Valor del potenciómetro es:");  
8     wifiSerial.println(volt);  
9     delay(1000);  
10 }
```

Al ser ejecutado el código ??, se observa en la terminal *CommOperator* la variación de los voltajes mandados por el potenciómetro (Figura 4.13).

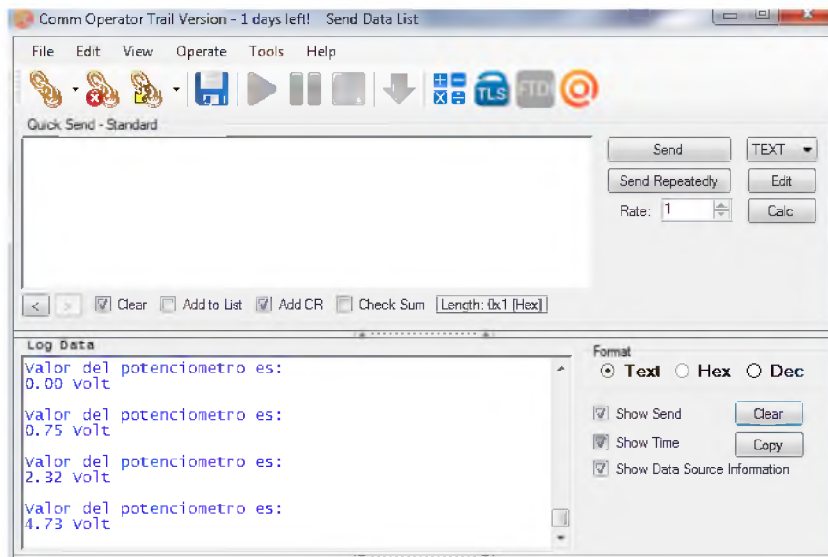


Figura 4.13: Visualización del voltaje adquirido por el potenciómetro.

## 4.5. Servicio Web implementado en el módulo WiFly

Concluida las pruebas del sistema, se creó una interfaz que permitirá visualizar los datos adquiridos por el sistema. Para crear dicha interfaz se utilizaron los lenguajes HTML y PHP [18, 11]. Se sabe que cada lenguaje tiene una estructura,

en HTML la estructura para construir su hoja de código está constituida por la cabeza (**head**) en el cual se introduce el encabezado, temas, título, etc y su segunda parte es el cuerpo (**body**). En esta sección se inicia la programación.

El código 4 muestra la interfaz final (Figura 4.14). La tabla contiene un recuadro llamado hora. En ese recuadro se colocará el reloj introducido por medio de la programación PHP de tal forma que este se vaya actualizando cada cierto tiempo.

Código 4: Interfaz del módulo WiFly en HTML.

```
1 <html>
2   <head>
3     <META HTTP-EQUIV="Refresh" CONTENT="0.5" />
4     <title> PHP: Tutorial </title>
5 </head>
6 <body>
7 <h1 style="color:red"> INTERFAZ MODULO WiFly</h1>
8 <h4 style="color:green"> Muestreo de temperatura mediante
9 el modulo WiFly y la placa Arduino Mega. </h4>
10 <table border=1 align= 'center' >
11 <tr>
12 <td colspan=4><h3 style="text-align:center" >
13   TABLA DE TEMPERATURA EN &deg;C </h3></td>
14 </tr>
15 <tr>
16 <td colspan=2 ><h4>Temperatura</h4></td>
17 <td colspan=2 ><h4>Hora</h4></td>
18 </tr>
19 <tr>
20 <td colspan=2 ><h4></h4><?php
21   echo $personas ['medida'] ['buf']; ?></td>
22 <td colspan=2 ><h4></h4><?php
23   echo $personas ['tiempo'] ['hora']; ?></td>
24 </tr>
25 </table>
26 </body>
27 </html>
```

Como se observó en los códigos anteriores, se utilizaron etiquetas como lo son `</h4>`, `</td>`, `</tr>`, todas estas fueron utilizadas para darle formato, tal como es, el tamaño de la letra, colocar un salto de línea, el estilo y color, al igual que la dirección del texto [18].



Figura 4.14: Interfaz para mostrar los datos.

La programación PHP sirve para darle vida a las páginas web, ya que este lenguaje permite manipular el contenido de la página web en el servidor justo antes de que la página sea expuesta en el explorador del cliente. Un programa PHP cuando se ejecuta en el servidor, altera o genera un código HTML. La etiqueta `<?php` abre una sección, desde aquí en adelante se trata de código PHP por lo menos hasta llegar a la etiqueta `?>` de cierre (código 5).

Código 5: Estructura de inicio de la programación en PHP.

```
<?php
//Aqui se escribe el codigo PHP
?>
```

Para realizar la programación PHP de la comunicación, debemos utilizar las funciones llamadas sockets. Una conexión típica de red está constituida por 2 sockets, uno realizando el papel del cliente, y otro realizando el papel del servidor. Para realizar la comunicación con el módulo y la interfaz es necesaria la función `socket_create`, de acuerdo a su descripción, esta función crea y devuelve un recurso socket, también referido como un extremo de comunicación (código 6).

Código 6: Estructura del socket create en PHP.

```
1 <?php
2 resource socket_create (int $domain, int $type, int $protocol)
3 ?>
```

El parámetro **domain** tabla 4.3 especifica la familia de protocolos que va a usar por el socket .

Dominio	Descripción
AF_INET	Protocolos basados en Internet IPv4. TCP y UDP son protocolos comunes de esta familia de protocolos.
AF_INET6	Protocolos basados en Internet IPv6. TCP y UDP son protocolos comunes de esta familia de protocolos.
AF_UNIX	Familia de protocolos de comunicación local. Alta eficacia y baja sobrecarga la hacen una gran forma de IPC (Interprocess Communication - Comunicación entre procesos).

Tabla 4.3: Familia de protocolos

El parámetro **type** tabla 4.4 selecciona el tipo de comunicación que va a usar el socket .

Tipo	Descripción
SOCK_STREAM	Proporciona flujos de bytes orientados a conexión, secuenciados, fiables y full-duplex. Puede soportar un mecanismo de transmisión fuera de banda. El protocolo TCP está basado en este tipo de socket.
SOCK_DGRAM	Soporta datagramas (no orientado a conexión, mensajes no fiables de una longitud máxima). El protocolo UDP está basado en este tipo de socket.
SOCK_SEQPACKET	Proporciona una ruta de transmisión de datos orientada a conexión secuenciada, fiable, de dos direcciones para datagramas de longitud máxima fija; se requiere un consumidor para leer un paquete entero con cada llamada de lectura.
SOCK_RAW	Proporciona acceso al protocolo de red sin formato. Este tipo especial de socket se puede usar para construir manualmente cualquier tipo de protocolo. Un uso común para este socket es realizar solicitudes ICMP (como ping).
SOCK_RDM	Proporciona una capa de datagramas fiable que no garantiza el orden. Lo más seguro es que esto no esté implementado en su sistema operativo.

Tabla 4.4: Tipos de sockets disponibles

El parámetro **protocol** (tabla 4.5) establece el protocolo dentro del **domain** especificado la familia de protocolos que usara al comunicarse con el socket devuelto. El valor apropiado puede ser recuperado por su nombre usando **getprotobyname()**. Si el protocolo deseado es TCP o UDP, también se pueden usar las constantes SOL\_TCP, y SOL\_UDP correspondientes.

Nombre	Descripción
icmp	Internet Control Message Protocol (Protocolo de Mensajes de Control de Internet) es usado principalmente por pasarelas y hosts para notificar errores en la comunicación por datagramas. El comando "ping" (presente en la mayoría de los sistemas operativos modernos) es un ejemplo de la aplicación del ICMP.
udp	User Datagram Protocol (Protocolo de Datagramas de Usuario) es un protocolo no orientado a conexión, no fiable, con longitud de registros fija. A causa de estos aspectos, UDP requiere una cantidad mínima de sobrecarga del protocolo.
tcp	Transmission Control Protocol (Protocolo de Control de Transmisión) es un protocolo fiable orientado a conexión, orientado a flujo, full duplex. TCP garantiza que todos los paquetes de datos serán recibidos en el orden en el que fueron enviados. Si algún paquete se pierde por alguna razón durante la comunicación, TCP automáticamente retransmitirá el paquete hasta que el host destino admita ese paquete. Por razones de fiabilidad y rendimiento, la implementación de TCP decide por sí misma los límites apropiados de los octetos de la capa de comunicación de datagramas subyacente. Por lo tanto, las aplicaciones TCP deben permitir la posibilidad de transmisión de registros parciales.

Tabla 4.5: Protocolos comunes

La función de `socket_create()` devuelve un recurso socket en caso de éxito, o `FALSE` en caso de error. El código de error real se puede recuperar llamando a `socket_last_error()`. Este código de error se puede pasar a la función `socket_strerror()` para obtener una explicación textual del error, como se muestra en el código 7.

Código 7: Socket last error para obtener una explicación del error.

```
1 <?php
2     $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
3     if ($socket == false)
4     {
5         echo "socket_create() _ fallo : _ razon : " .
6         socket_strerror(socket_last_error()). "</br>";
7     }
8     else
9     {
10        echo "OK.</br>";
11    }
12 ?>
```

La descripción en el código 8 del `socket_connect` inicia con una conexión hacia la dirección **address** usando el recurso **socket**, el cual debe ser un recurso socket válido creado con `socket_create()`.

Código 8: Estructura del socket de conexión.

```
1 <?
2 bool socket_connect(resource $socket, string
3 $address [, int $port=0])
4 ?>
```

Los parámetros que utiliza la función `socket_connect()` son `socket`, este es el resultado que genera el `socket_create()`; El parámetro **address** es una dirección IPv4 en notación decimal con puntos (127.0.0.1), el parámetro **port** sólo se usa y es obligatoria al conectarse a un socket con una dirección IPv6 válida o a un socket con el nombre de ruta de un socket de dominio Unix, y designa el puerto del **host** remoto al que debería de hacerse una conexión.

Este socket devuelve TRUE en caso de éxito o FALSE en caso de error código 9. El código de error se puede recuperar de la misma manera que el `socket_create()`.

Código 9: Conexión del socket a partir de una IP y un puerto.

```
1 <?
2 echo "Intentando conectar a '$address' en el
3 puerto '$service_port'.";
4 $result = socket_connect($socket, $address, $service_port);
5 if ($result === false)
6 {
7     echo "socket_connect() fallo.</br>Razon:( $result)".
8     socket_strerror(socket_last_error($socket)) . "</br>";
9 }
10 else
11 {
12     echo "OK.</br>";
13 }
14 ?>
```

Por último solo queda estudiar la función `socket_recvmsg`, esta función se encarga de leer el mensaje que es transmitido por el medio inalámbrico, y su estructura para ser utilizada se muestra en el código 10:

Código 10: Parametros de socket recvmsg.

```
1 <?
2 socket_recvfrom($socket, $buf, 50, 0, $address, $service_port);
3 ?>
```

Teniendo la información necesaria de cómo trabajar con las funciones de los sockets solo falta establecer los parámetros necesarios que utilizaremos para construir el programa que funcione correctamente y se realice la conexión del módulo con la interfaz, de manera que permita visualizar los datos que se irán adquiriendo [11].

## 4.6. Funcionamiento

En este apartado se describen las aplicaciones realizadas con el sistema. A fin de ejecutar los programas con extensión (.php) fue necesario utilizar un servidor con aplicación web llamada *Apache* (Figura 4.15), este servidor utilizado ejecuta programación PHP. Para esto es necesario instalar *Apache* y configurarlo para que ejecute los codigos PHP, *XAMPP*® es un software que incluye en un solo clic esta instalación, otros que pueden ser utilizados son MySQL, FileZilla, entre otras.

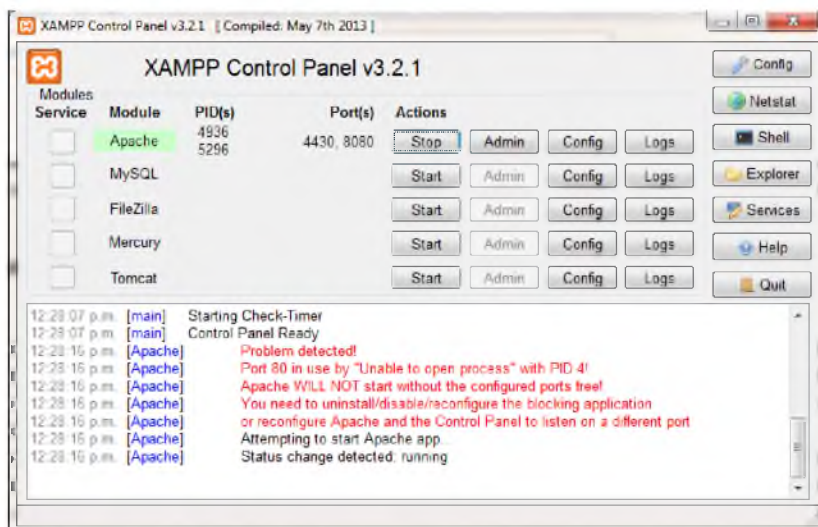


Figura 4.15: Software XAMPP para inicializar la conexión localhost.

La aplicación inicial implementada en el módulo fue la de medir la intensidad de luz mediante una fotorresistencia. Se buscó una aplicación sencilla, de manera que generara una señal analógica. Una de las configuraciones más comunes de las fotorresistencias es el divisor de voltaje. En la Figura 4.16 se muestra dos posibles conexiones. Estos circuitos muestran una fotorresistencia dispuesta de tal manera que formen un divisor de voltaje, con un voltaje de entrada  $V_{in}$  y una resistencia fija  $R$ . La diferencia entre estas configuraciones consiste, que cuando la luz incide sobre la fotorresistencia, el voltaje se ve disminuido (como se observa en el circuito a), mientras que para el circuito (b) cuando se hace incidir luz sobre la fotorresistencia, el voltaje  $V_{out}$  se ve aumentado.

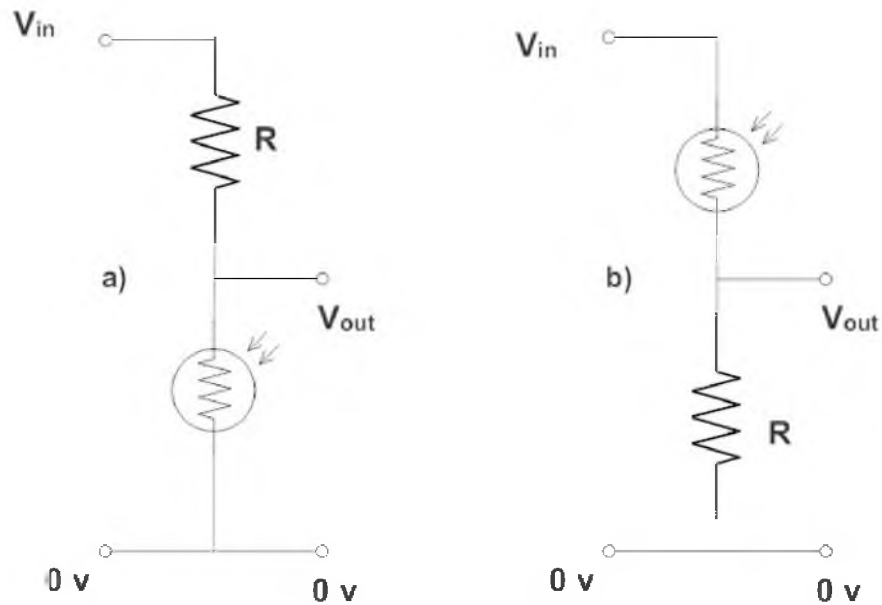


Figura 4.16: Divisor de voltaje usando una fotorresistencia.

El  $V_{out}$  será adquirida por medio del Arduino y transmitida por el módulo. Para ello en la Figura 4.17 se muestra la conexión de la fotorresistencia y el Arduino.

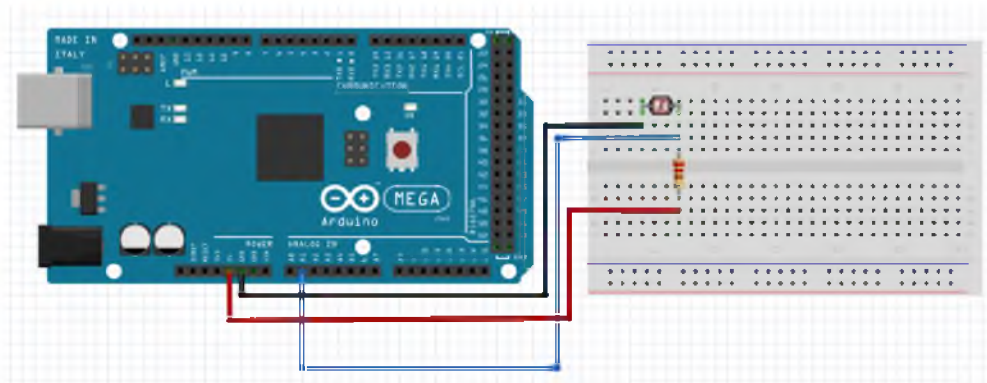


Figura 4.17: Circuito de la fotorresistencia donde la salida disminuye al aumentar la intensidad luminosa.

A continuación se realiza la descripción detallada del código que sirvió para la adquisición de la señal analógica de la fotorresistencia. En este código se lee el valor del voltaje entre una fotorresistencia y una resistencia fija (línea 3), el valor leído se envía por los pines de comunicación para que se transmita por el módulo (línea 6), de forma que cada segundo se repita este proceso (línea 10), como se presenta en el código 11.

Código 11: Obtención del valor analógico de la fotorresistencia.

```
1 void loop()
2 {
3   valorAnalogico=analogRead(1);
4   volt=(valorAnalogico*5)/1023;
5
6   wifiSerial.print(" Valor del sensor Analogico=");
7   wifiSerial.println(valorAnalogico);
8   wifiSerial.println();
9
10  delay(1000);
11 }
```

En la Figura 4.18 se muestra la interfaz gráfica, en esta se observa que la salida de voltaje baja aproximadamente 1 volt al aumentar la intensidad luminosa que se incide en la fotorresistencia, este tipo de configuración puede ser modificada, de acuerdo a los circuitos mostrados anteriormente.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/tutorial/interfaz\_led.php'. Below the address bar, the text 'TCP/IP Connection' is visible. The connection log shows the following steps: 'OK. Intentando conectar a '169.254.1.1' en el puerto '2000'...', 'OK. Se inició la conexión en la dirección 169.254.1.1 En el puerto 2000', and 'Leyendo respuesta de la wifly:'. Below the log, the heading 'INTERFAZ MODULO WiFly' is displayed in red. Underneath, there is a green text line: 'Muestreo de intensidad de luz con el modulo WiFly y la placa Arduino Mega.' and a black text line: 'La salida disminuye al aumentar la iluminación'. To the right of the text, there is a table with the title 'TABLA DE INTENSIDAD DE LUZ'.

Señal	Hora
1.00	01:13:23

Figura 4.18: Interfaz donde se muestra el cambio de la salida cuando se aumenta la intensidad luminosa de la fotorresistencia.

En la Figura 4.19 se presenta la conexión física del módulo WiFly en conjunto con el Arduino y la fotorresistencia.

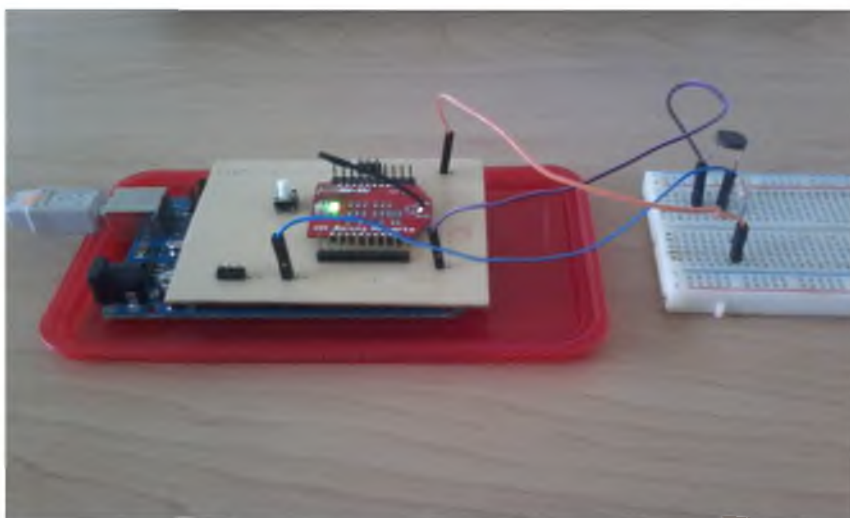


Figura 4.19: Interacción del módulo WiFly con la fotoresistencia.

La segunda aplicación que se realizó permite adquirir por medio del módulo WiFly y el integrado LM35 la temperatura y así ser enviada vía red LAN a cualquier computadora conectada a ella.

Con las características descritas en el capítulo 2 del sensor de temperatura LM35, se tiene un valor de escala de  $+ 10.0 \text{ mV}/^\circ\text{C}$ , esto quiere decir, si el sensor detecta 1 grado centígrado a la salida del sensor obtendríamos 10 mV. Por ejemplo si tenemos  $36.4^\circ\text{C}$  se tendrá 364 mV que sería equivalente a 0.364 V, por otra parte sabemos que el convertidor de analógico a digital del *Arduino* es de 10 bits de resolución y los valores variarán entre 0 y 1023; entonces la fórmula que se empleó es la siguiente.

$$V_{out} = \frac{((5v)(outpin))(100)}{1023}$$

A continuación se da una breve explicación del código empleado para adquirir la temperatura. Se declararon las variables que fueron utilizadas para almacenar los valores (**Sensor**, **Pinlm**, **tempC**). En la función **void loop** se lee el valor del sensor (línea 2), se convirtió el valor obtenido del sensor, es decir, la lectura del voltaje se convirtió a temperatura con la fórmula descrita anteriormente (línea 3) del código 12.

En las líneas 5 y 6 se envía el dato para que sea transmitido, y después de cada segundo se vuelve a repetir el proceso, como se presenta en el código 12.

Código 12: Adquiriendo la señal del sensor LM35.

```
1 void loop()
2 {
3   Sensor=analogRead(1);
4   tempC=(5.0*Sensor*100.0)/1023.0;
5
6   wifiSerial.print(tempC);
7   wifiSerial.print("Grados Celsius \n");
8   delay(1000);
9 }
```

Para la construcción del circuito fue necesario hacer una revisión de la hoja de datos del sensor para observar su configuración (Figura 4.20a), de acuerdo a esta se realizó el circuito de la Figura 4.20b el cual conecta el sensor LM35 al *Arduino*.

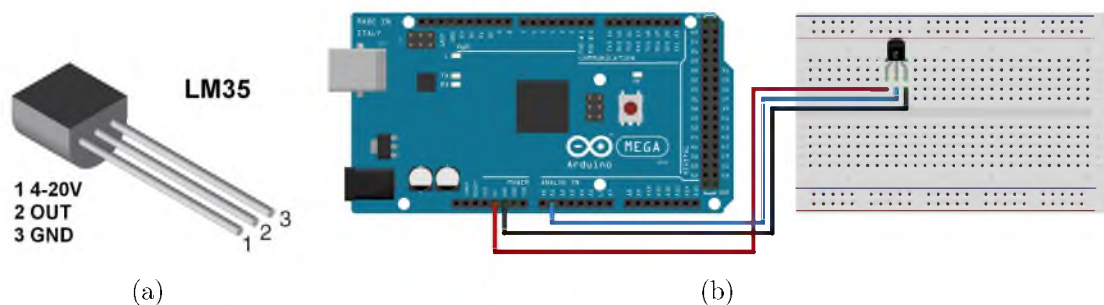


Figura 4.20: En la Figura (a) se muestra la configuración del sensor de temperatura y en la Figura (b) circuito del sensor LM35 con el *Arduino*.

En la Figura 4.21 se muestra el resultado final de la interfaz. En ella se observa el valor adquirido por el sensor LM35. En la Figura 4.22 se presenta el acoplamiento físico del conjunto WiFly *Arduino* y el sensor.

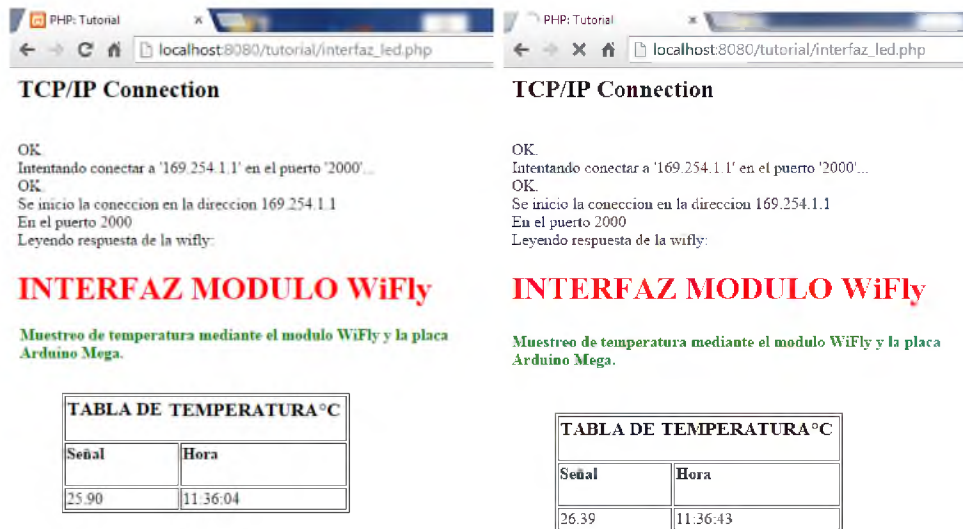


Figura 4.21: Temperatura adquirida por el sensor en un salón y mostrada en la interfaz.

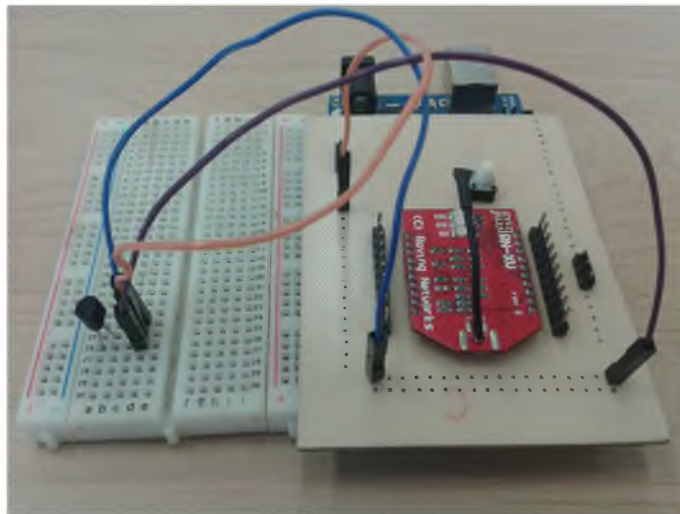


Figura 4.22: Circuito real del sensor junto con el Arduino y el módulo WiFly.

Terminada la etapa de aplicaciones, se observó el funcionamiento planeado en conjunto con el Arduino y los sensores, de esta manera se pudieron adquirir los datos censados y así poder ser mostrados en la interfaz diseñada.

## 5. CONCLUSIONES

En esta tesis se diseñó y construyó un sistema de adquisición de datos con capacidad de acceso a una red computacional vía comunicación inalámbrica. Para tal sistema se utilizaron distintos sensores (fotorresistencia, LM35), una tarjeta Arduino Mega (como procesador de información) y un módulo de acceso a red inalámbrica WiFly RN XV 171. El sistema en conjunto adquiere señales físicas para ser enviadas vía comunicación inalámbrica a un servidor. Este pone a disposición la información obtenida de los sensores a cualquier cliente que la solicite, conformándose en una conexión cliente servidor.

El sistema de adquisición de datos está constituido por el módulo de censado, módulo de transmisión y la interfaz computacional. El módulo de censado convierte una magnitud física en una señal eléctrica. En este trabajo se realizó un estudio, para este módulo, sobre las propiedades, características y configuración de dispositivos fotorresistivos y el circuito integrado LM35 diseñado para la medición de temperatura. La señal eléctrica de estos sensores es adquirida en el Arduino Mega por medio de una terminal AD. Este procesa la información y la transmite, en forma serial al Módulo WiFly. Se investigó el funcionamiento, conexión y operación del módulo a fin de obtener la conexión necesaria a la red local. El módulo fue configurado para crear una red ADHOC y tener una conexión punto a punto. Una vez creada la red se tuvo la necesidad de programar la interfaz para el usuario. Dicha interfaz fue programada utilizando los lenguajes de programación HTML y PHP. Como resultado se obtuvo una interfaz que logra la comunicación inalámbrica desde un sistema de adquisición hasta desplegar la información en una computadora vía red Lan.

Se considera esta propuesta interesante debido al acceso de la información vía web y en su caso, el cual no está considerado en este trabajo, activar o desactivar actuadores. Por ejemplo sería práctico apagar lámparas, encender un aire acondicionado, apagar una estufa, o controlar dispositivos desde cualquier punto con conexión a Internet o a la red local. Esta idea puede ser utilizada en el desarrollo de distintos proyectos, por mencionar el control de un robot móvil vía web.

## A. Códigos *Arduino*<sup>®</sup>

Es este apartado se muestra los códigos que se utilizaron.

```
#include <SoftwareSerial.h>
SoftwareSerial wifiSerial(14,15);

void setup()
{
    Serial.begin(115200);
    wifiSerial.begin(9600);
}
void loop()
{
    wifiSerial.println();
    wifiSerial.println("Hola Mundo");
    delay(1000);
}
```

### Código del potenciómetro

```
include <SoftwareSerial.h>
SoftwareSerial wifiSerial(14,15);
int valorPot=0;

void setup()
{
    Serial.begin(115200);
    wifiSerial.begin(9600);
}
void loop()
{
```

```
valorPot=analogRead(2);
wifiSerial.println();
wifiSerial.println("Valor del potenciómetro es:");
Serial.println(valorPot);
wifiSerial.println(valorPot);
delay(1000);
}
```

### Código del sensor de temperatura LM35

```
#include <SoftwareSerial.h>
SoftwareSerial wifiSerial(14,15);

int Sensor = 0.0;
float tempC = 0.0;

void setup()
{
  Serial.begin(115200);
  wifiSerial.begin(9600);
}
void loop()
{
  Sensor = analogRead(1);
  tempC = (5.0 * Sensor * 100.0)/1023.0;
  wifiSerial.print(tempC);
  Serial.print(tempC);
  delay(1000);
}
```

## B. Código HTML en conjunto con PHP

Código de la interfaz para la visualización de la información obtenida por el conjunto Arduino-WiFly.

```
<?php
date_default_timezone_set('America/Mexico_City');
$hora=date('h:i:s');

//empieza el socket

error_reporting(E_ALL);

echo "<h2>TCP/IP Connection</h2></br>";

/* Obtener el puerto para el servicio WWW. */
//$service_port = getservbyname('tcp');
$service_port = 2000;

/* Obtener la direccion IP para el host objetivo. */
//$address = gethostbyname('localhost');
$address = "169.254.1.1";

/* Crear un socket TCP/IP. */
$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
if ($socket === false) {
echo"socket_create() fallo: razon:".socket_strerror(socket_last_error())."</br>";
} else {
    echo "OK.</br>";
}

echo "Intentando conectar a '$address' en el puerto '$service_port'...</br>";
$result = socket_connect($socket, $address, $service_port);
if ($result === false) {
    echo "socket_connect() fallo.</br> Razon: ($result) " .
    socket_strerror(socket_last_error($socket))."</br>";
```

```

} else {
    echo "OK.</br>";
}

echo "Se inicio la conexion en la direccion $address </br>" ;
echo "En el puerto $service_port </br>" ;
echo "Leyendo respuesta de la wifly:</br>";

$i = 1;
while ($i <= 2) {
    socket_recvfrom($socket, $buf, 50, 0, $address, $service_port);
    $i++;
    //echo "Se recibio $buf </br>" . PHP_EOL;
}
$personas=[
    'medida'=>['buf'=> $buf],
    'tiempo'=>['hora'=> date('h:i:s')],
];
?>

<!DOCTYPE html>
<html>

<head>
<META HTTP-EQUIV="Refresh" CONTENT="0.5" />
<title> PHP: Tutorial </title>
</head>
<body>
<h1 style="color:red"> INTERFAZ MODULO WiFly</h1>
<h4 style="color:green"> Muestreo de temperatura mediante el modulo
WiFly y la placa Arduino Mega. </h4>
<table border=1 align= 'center'>
<tr>
<td colspan=4><h3 style="text-align:center" >TABLA DE TEMPERATURA EN &deg;C </h3>
</td>
</tr>

```

```
<tr>
<td colspan=2 ><h4>Temperatura</h4></td>
<td colspan=2 ><h4>Hora</h4></td>
</tr>
<tr>
<td colspan=2 ><h4></h4><?php echo $personas['medida']['buf']; ?></td>
<td colspan=2 ><h4></h4><?php echo $personas['tiempo']['hora']; ?></td>
</tr>
</table>
</body>
</html>
```

## Bibliografía

- [1] Arduino, “<http://www.arduino.cc>.” 4.3
- [2] Areny, Ramon Pallas. *Adquisición y distribución de señales*. Barcelona, 1993. 2.4
- [3] Asenjo, Jorge Sánchez. *Servidores de Aplicaciones Web*. 2011. 3.5
- [4] Gallardo, Armando Román, et al. “Plataforma Domótica para la Interconexión de Objetos Físicos a Internet Empleando Conectividad WiFi.” 2014. 3.6.1, 3.6.1
- [5] Guillen, Xavier Vilajosana, et al. *Mecanismos de Comunicación*. 2014. 3.4
- [6] Leon W. Couch, II. *Sistemas de comunicación digitales y analógicos*. (Séptima edición Edition). University of Florida, Gainesville, 2008. 1.1
- [7] Montañana. *Redes de Comunicación*. España, 2004-2009. 3.1
- [8] Muñoz, Miguel Angel Sánchez. *Robot auxiliar de limpieza de suelos doméstico*. PhD dissertation, Universitat Oberta de Catalunya, Junio 2014. 3.6.2
- [9] Ortiz, Rodríguez, et al. “Aplicación de la Tecnología ZigBee en la Comunicación Inalámbrica Digital con un Controlador Lógico Programable para el Control Industrial en Zonas de Alto Riesgo.” Instituto Tecnológico de San Juan del Rio, 2014. 3.6.4
- [10] Patrón, Guillermo Abraham Viana, et al. *Desarrollo de sistema de telemetría para implantar en prototipos móviles para el monitoreo de sitios remotos*. PhD dissertation, Universidad del Caribe, Abril 2012. 3.6.3
- [11] PHP, “<http://php.net>.” 4.5, 4.5
- [12] Ramírez, Leonel G. Corona, et al. *Sensores y Actuadores. Aplicaciones con Arduino*. México D.F., 2014.
- [13] rovingnetworks, “[www.rovingnetworks.com](http://www.rovingnetworks.com).” 4.2
- [14] Salvetti, Diego. *Redes Wireless*. Buenos Aires, 2011. 3.2

- [15] Stallings, William. *Data and Computer Communications*. Upper Saddle River, New Jersey, 2007. 2.2
- [16] Villegas, Rolando Rojas, et al. *Internet y Redes Inalambricas*. Arequipa, Perú, 2009.
- [17] Wayne, Tomasi. *Sistemas de comunicaciones electrónicas* (Cuarta edición Edition). Phoenix, Arizona, 2003. 3.1
- [18] world's largest web developer site, The, "<http://www.w3schools.com>." 4.5, 4.5