



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DISEÑO E INTEGRACIÓN DE UN SISTEMA DE
ADQUISICIÓN DE DATOS MEDIANTE EL USO DE
ARDUINO Y RASPBERRY-PI**

TESIS PROFESIONAL

PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:
NÚÑEZ RODRÍGUEZ JUAN ALFREDO

DIRECTOR DE TESIS:
DR. VÍCTOR MANUEL GUERRA ORTIZ

MÉXICO D.F. 2014



Agradecimientos

Son muchas las personas que me gustaría agradecerles por el apoyo que me dieron para cumplir este objetivo, les agradezco por sus consejos, amistad y ánimo que ofrecieron a lo largo de la carrera.

Principalmente me gustaría agradecer a mi familia por inculcarme valores que sirvieron como base para seguir estudiando, a mis padres porque me brindaron todo el apoyo que necesité para la culminación de esta etapa, a mis abuelos por la educación y consejos que me brindaron en el transcurso de la vida, a mis hermanos por la tolerancia y comprensión que tuvieron en los buenos y malos momentos.

A la Universidad Nacional Autónoma de México, Facultad de Ingeniería, así como a los profesores por ofrecerme la formación y herramientas suficientes para seguir adelante.

A la Unidad de Servicios de Cómputo Académico y a la Dirección General de cómputo y de Tecnologías de Información y Comunicación, por permitirme formar parte de la generación de becarios.

Contenido

CAPÍTULO 1: GENERALIDADES.....	1
1.1 INTRODUCCIÓN	1
1.2 OBJETIVO.....	3
1.2.1 Objetivo general.....	3
1.2.2 Objetivos particulares.....	3
1.3 MARCO TEÓRICO Y ESTADO DEL ARTE	4
CAPÍTULO 2: CONCEPTOS PRINCIPALES	11
2.1 SISTEMA DE ADQUISICIÓN DE DATOS.....	11
2.1.1 Definición.....	11
2.1.2 Elementos.....	11
2.1.3 Características	13
2.2 MICROCOMPUTADORAS Y MICROCONTROLADORES	15
2.2.1 Definición.....	15
2.2.2 Diferencia.....	15
2.2.3 Arquitectura.....	17
2.2.4 Elementos.....	18
2.3 ARDUINO	23
2.3.1 Definición.....	23
2.3.2 La placa de Arduino Uno	23
2.3.3 Microcontroladores	24
2.3.4 Elementos del Arduino.....	24
2.3.5 Aplicaciones.....	30
2.4 RASPBERRY-PI.....	31
2.4.1 Definición.....	31
2.4.2 La placa de la Raspberry-Pi.....	31
2.4.3 Elementos de la Raspberry-Pi	34
2.4.4 Aplicaciones.....	36
2.5 ALMACENAMIENTO DE DATOS	37
2.5.1 Dispositivos ópticos	37
2.5.2 Memorias de estado sólido.....	37
2.5.3 Dispositivos magnéticos.....	37

2.5.4	Bases de datos	38
2.5.5	Aplicaciones.....	40
CAPÍTULO 3: DESARROLLO DEL PROYECTO.....		41
3.1	INSTALACIÓN DE PLATAFORMA ARDUINO	44
3.1.1	Instalación de Arduino en Linux.....	44
3.1.2	Instalación de Arduino en Windows	44
3.2	INSTALACIÓN DEL SERVIDOR	45
3.2.1	Instalación del servidor Apache con php y MySQL en Linux	45
3.2.2	Instalación del servidor Apache con php y MySQL en Windows	47
3.2.3	Creación de un servidor web en Arduino.....	50
3.3	INSTALACIÓN DE LA BASE DE DATOS.....	53
3.3.1	Creación de la base de datos en Linux	53
3.3.2	Creación de la base de datos en Windows.....	54
3.4	CREACIÓN DEL PROYECTO.....	56
3.4.1	Creación de archivos del proyecto	56
3.4.2	Conexión	62
CAPÍTULO 4: ANÁLISIS DE RESULTADOS		65
4.1	APLICACIÓN DEL SISTEMA EN UN ENTORNO REAL	65
4.2	RESULTADOS.....	73
CAPÍTULO 5: CONCLUSIONES Y TRABAJO A FUTURO.....		75
APÉNDICE A: ARCHIVOS DEL SAD GENERAL		77
A.1	TesisSAD.ino	77
A.2	conectaBD.php.....	79
A.3	index.php.....	80
A.4	muestraDB.php.....	81
A.5	pruebaQuery.php.....	82
A.6	probarDB.php.....	82
A.7	ardaDB.php	83
A.8	guardaArc.php.....	84
A.9	cerrarDB.php.....	84
A.10	getValoresArduino.php	84
APÉNDICE B: ARCHIVOS DEL SAD DEL ENTORNO REAL.....		85
B.1	servidorParaCaja.ino	85

B.2	dirCajas.php	88
B.3	conectaDB.php.....	89
B.4	guardaArcTotal.php.....	89
B.5	index.php.....	89
B.6	getValoresArduino.php	91
B.7	cerrarDB.php.....	91
B.8	recupera.php	92
B.9	guardaArc.php.....	93
APÉNDICE C: OTROS CONCEPTOS		95
C.1	Transductores	95
C.1.1	Transductores resistivos	96
C.1.2	Transductores de reactancia variable	98
C.1.3	Transductores con semiconductores.....	98
C.1.4	Transductores pasivos	100
C.1.5	Transductores termoelectrónicos.....	102
C.1.6	Transductores ultrasónicos.....	103
C.1.7	Transductores digitales.....	103
C.2	Actuadores.....	104
C.2.1	Actuadores hidráulicos	104
C.2.2	Actuadores neumáticos	107
C.2.3	Actuadores eléctricos	107
C.3	Interferencias.....	110
C.3.1	Efectos ambientales.....	110
C.3.2	Interferencias electrostáticas	111
C.3.3	Interferencias electromagnéticas	111
C.3.4	Diafonía.....	112
C.3.5	Rechazo de ruido.....	112
TABLA DE ILUSTRACIONES.....		113
BIBLIOGRAFÍA Y REFERENCIAS		116

CAPÍTULO 1: GENERALIDADES

1.1 INTRODUCCIÓN

En el desarrollo de este trabajo de tesis se respeta el idioma nativo u original en algunas palabras que no permiten una traducción, identificando dichas palabras con un tipo de letra cursiva.

Hoy en día existen programas como LabVIEW que permiten hacer funciones similares a las del sistema que se desarrolla en este trabajo de tesis, ya que LabVIEW es una herramienta de pruebas de control y diseño, es utilizado para adquirir información del medio, mediante *hardware* diseñado específicamente para ello, sin embargo su alto costo los hace poco accesible para el desarrollo de actividades académicas o incluso para las pequeñas empresas.

En el caso del sector académico, en la Dirección General de cómputo y de Tecnologías de Información y Comunicación (DGTIC), no se cuenta con un sistema de adquisición de datos (de *software* libre y económico), que sea capaz de recolectar información del medio, la almacene, tome medidas de acuerdo a los valores capturados y pueda generar un reporte; además de ser consultado en tiempo real, desde cualquier dispositivo que cuente con conexión a Internet.

Cuando se trabaja en el área industrial o en el área de la investigación, muchas veces se requiere estar monitoreando variables externas en un sistema, para poder actuar sobre él de alguna manera o simplemente para llevar un registro de los datos capturados, es por ello que en este trabajo se desarrolló un sistema de adquisición de datos de propósito general para poder visualizar información en un entorno web y poder almacenarla en una base de datos, para luego generar un reporte que será de gran utilidad para los procesos que se desarrollan en estos campos. En el desarrollo de este sistema, también se contempla el uso de actuadores, los cuales nos permitirán, si es el caso, realizar actividades tales como: cerrar o abrir algún tipo de válvula, cortar o activar el suministro de corriente, mover motores, etc.

Este sistema a desarrollar, no sólo puede utilizarse en el área de la investigación, sino también en el área tecnológica, ya que puede ser implementado en muchas partes, adecuando el sistema con los sensores y actuadores necesarios, es decir se está diseñando un sistema práctico y escalable que se acople a las necesidades que se presenten. Por ejemplo se podría utilizar en el campo de la domótica (conjunto de sistemas que se utilizan para automatizar una vivienda), utilizando la conexión con cable Ethernet para comunicar el servidor web con el Arduino y se pueda manipular un *switch* (un relevador por ejemplo) que encienda las luces de una vivienda, entre otros.

Este proyecto permite integrar tecnologías que usualmente se usan de manera individual, pero al ser unidas dan como resultado el desarrollo de aplicaciones, que son capaces de capturar, analizar y almacenar datos de variables físicas del medio, y cuya flexibilidad les permitirá adaptarse y ser usadas en cualquier entorno que requiera de este tipo de registros.

Este sistema está diseñado para dar una implementación específica a las microcomputadoras y microcontroladores que han surgido en los últimos años. Se pretende utilizarlas para automatizar los métodos de recolección de datos, ya que existen infinidad de procesos que deben ser monitoreados constantemente para ubicar alguna falla en los sistemas en cualquier momento, con esto se impulsa a la utilización de las Tecnologías de la Información y Comunicación (TIC), que permiten automatizar la medición de variables físicas que puedan representar datos o información. Los Sistemas de Adquisición de Datos (SAD), son sistemas automatizados para obtener y almacenar dicha información, así como alertar al usuario cuando alguna de las variables físicas sobrepasa los límites o rangos establecidos.

Este trabajo de tesis está pensado y desarrollado de tal manera que cualquier persona que no sea experto en la materia de ciencias de la computación pueda utilizar el sistema para la captura y almacenamiento de datos provenientes de mediciones de variables físicas, tal es el caso de investigadores, técnicos y profesionistas que tengan el interés de llevar un registro del estudio de fenómenos físicos de manera automatizada. Debido a lo anterior en el Capítulo 2 “*Conceptos principales*” se desarrolla una investigación de los conceptos básicos utilizados en este trabajo, tales como; características principales de un SAD, microcomputadoras y microcontroladores (Arduino y Raspberry-Pi), incluyendo los elementos externos necesarios para captura y análisis de señales (transductores) y actuadores responsables de llevar a cabo una acción cuando el sistema o el usuario lo requiera.

En el capítulo 3 “*Desarrollo del proyecto*” se explica a detalle cómo se desarrolló el SAD, incorporando todos los elementos responsables para que el sistema funcione adecuadamente. Se pretende que el SAD pueda ser replicado por personas que no tienen conocimientos de la administración de los sistemas operativos Linux y Windows, debido a esto, se describe paso a paso la instalación de las herramientas necesarias para desarrollar el SAD en los sistemas operativos mencionados anteriormente. Cabe destacar que el sistema que se describe en este capítulo es moldeable, dependiendo de las variables físicas a medir y los transductores utilizados, es decir un sistema base, o en otros términos un sistema de propósito general, que es posible modificar de acuerdo a la señal proveniente de los transductores seleccionados. El sistema es capaz de alertar al usuario si es que, alguna de las variables sobrepasa los rangos permitidos, para que el usuario pueda tomar una decisión, que le permita controlar las variables, además de un relevador que se activa o desactiva mediante la aplicación web y un motor a pasos que el usuario puede mover a libertad.

Para aplicar el sistema desarrollado en el trabajo de esta tesis en el Capítulo 4 “*Análisis de resultados*” se implementó el sistema descrito en el Capítulo 3 “*Desarrollo del proyecto*”. Para este ejemplo de funcionamiento del sistema se utilizaron transductores de temperatura y luz, y se midieron variables eléctricas (voltajes y corrientes) para su estudio.

Por último en el Capítulo 5 “*Conclusiones y trabajo a futuro*” se muestran las conclusiones del trabajo de tesis, así como las posibles mejoras que puede tener el SAD con base en la investigación realizada y a las nuevas y existentes tecnologías.

1.2 OBJETIVO

1.2.1 Objetivo general

Diseñar e integrar un sistema de adquisición de datos mediante el uso de microcomputadoras Arduino y Raspberry-Pi que se pueda controlar y monitorear mediante conexiones a internet y que sea útil al estudio de fenómenos físicos.

1.2.2 Objetivos particulares

- Explicar las características de los dispositivos Arduino y Raspberry-Pi.
- Desarrollar una aplicación web. Que le permita al usuario comunicarse con el sistema, visualizar las variables físicas, guardarlas y consultarlas desde la base de datos y generar un documento que contenga la información de estas variables físicas para un estudio posterior.
- Impulsar el uso de tecnologías como Arduino y Raspberry-Pi y facilitar a la comunidad la adquisición de datos provenientes de variables físicas.

1.3 MARCO TEÓRICO Y ESTADO DEL ARTE

Con el surgimiento de las nuevas Tecnologías de la Información y la Comunicación (TIC) y la microelectrónica ha sido el fundamento de un avance en la humanidad extraordinario. Esta revolución ha permeado todas las actividades humanas, hoy en día la mayoría de disciplinas, profesiones, y en general, muchas actividades utilizan la ciencia de la computación.

De esta misma manera las microcomputadoras como las que se tratan en este trabajo de tesis, en pocos años serán el instrumento informático más utilizado, no sólo en los nichos actuales, sino que expandirá su penetración a prácticamente todo lo conocido.

Muchas de las aplicaciones de hoy en día no serían posibles sin el uso de sensores, ya que ellos tienen la capacidad de informarnos de los cambios en magnitudes físicas, para su conocimiento y control. La utilización de sensores es indispensable en la automatización de industrias, por lo que un gran número de dispositivos de medición utiliza un principio eléctrico básico para cualquier etapa en la recolección de datos, ya sea captura, acondicionamiento, procesamiento, almacenamiento, etc.

“...El proceso de medición generalmente requiere el uso de un instrumento como medio físico para determinar la magnitud de una variable. Los instrumentos constituyen una extensión de las facultades humanas y en muchos casos permiten a las personas determinar el valor de una cantidad desconocida la cual no podría medirse utilizando solamente las facultades sensoriales...” [3]

Con el avance de la tecnología hoy en día contamos con infinidad de instrumentos especializados en adquirir señales, estudiarlas e interpretarlas de manera más óptima, estos instrumentos denominados instrumentos de medición.

Como lo señalan Copper, W. & Helfrick, A., para el trabajo de medición se emplean las siguientes definiciones:

- *Instrumento: dispositivo para determinar el valor o la magnitud de una cantidad variable*
- *Exactitud: aproximación con la cual la lectura de un instrumento se acerca al valor real de la variable medida.*
- *Precisión: medida de la reproducibilidad de las mediciones; esto es dado el valor fijo de una variable, la precisión es la medida del grado con el cual las mediciones sucesivas difieren una de otra.*
- *Sensibilidad: relación de la señal de salida o respuesta del instrumento respecto al cambio de la entrada o variable medida.*
- *Resolución: cambio más pequeño en el valor medido al cual responde el instrumento.*
- *Error: desviación a partir del valor real de la variable medida*

Exactitud y precisión

“...Exactitud se refiere al grado de aproximación o conformidad al valor real de la cantidad medida. Precisión es el grado de concordancia dentro de un grupo de mediciones o instrumentos...” [3].

Las cifras significativas proporcionan información acerca de la magnitud real de las mediciones. Mientras más aumenta el número de cifras significativas es mayor a precisión de la medición.

Error

De acuerdo con el sitio web de la Real Academia Española (RAE) el error es definido como:

“Diferencia entre el valor medido o calculado y el real.”[22]

Cuando realizamos mediciones es imposible que sea perfecta porque siempre están involucrados los errores que llegan a afectar la medición.

Dependiendo del tipo de error se puede clasificar en:

Error humano

Al realizar una medición es indispensable conocer el funcionamiento del instrumento, así como el resultado que éste pueda arrojar, para poder registrar y calcular adecuadamente los resultados de las mediciones, la persona se debe de hacer responsable de calibrar adecuadamente el instrumento de medición para que arroje resultados reales, este tipo de error se debe al mal uso del instrumento, el desconocimiento del mismo y descuidos, siempre hay que considerar en primer término este tipo de errores ya que son los más comunes debido a los malos hábitos y se considera que son difíciles de evitar.

Error sistemático

Cuando estamos realizando mediciones y llevando a cabo la calibración de un instrumento es indispensable que el valor medido permanezca constante a condiciones iguales, los errores sistemáticos son debidos a la modificación de las condiciones, por lo que los resultados no resultan acordes a dichos cambios. Este tipo de errores no sólo depende del instrumento sino del método utilizado para efectuar la medición.

- Errores Instrumentales: Errores propios del instrumento de medición muchas veces debido a fallas mecánicas, o errores de fabricación. También se deben cuando el instrumento no responde a la rapidez con la que la variable a medir está cambiando o incluso cuando no cumple con el rango de medición.
- Errores ambientales: El medio ambiente es muy impredecible y cuando cambian las condiciones climáticas es imposible evitar o controlar los cambios en el medio ambiente.

Errores aleatorios

Los errores aleatorios son los que permanecen una vez eliminadas las causas de errores sistemáticos. Se manifiestan cuando se mide repetidamente la misma magnitud, con el mismo instrumento y el mismo método. La presencia de errores aleatorios hace que después de realizar una o varias medidas de una determinada magnitud se tenga una incertidumbre sobre el verdadero valor de ésta (valor exacto).

Morón, J. menciona que el análisis de sistemas puede dividirse en tres aspectos: [10]

- *El desarrollo de un modelo matemático apropiado para el problema físico bajo consideración. Esta parte del análisis se dedica a la obtención de “ecuaciones dinámicas”, condiciones iniciales o de frontera, valores de parámetros, etc. En este proceso es donde el juicio, la experiencia y la experimentación se combinan para lograr el desarrollo de un modelo apropiado. En esta forma, este primer aspecto es el más difícil de desarrollar formalmente.*
- *Después de obtener un modelo apropiado, se resuelven las ecuaciones resultantes para encontrar soluciones de diversas formas.*
- *Luego, la solución del modelo matemático se relaciona o interpreta en función del problema físico.*

Cuando se habla de sistemas, especialmente de un Sistema de Adquisición de Datos forzosamente se tiene que hablar de las señales que abundan en la naturaleza las cuales necesitamos capturar para analizar y estudiarlas en forma de datos, a continuación se explica el tipo de señales que se pueden encontrar para su análisis.

Señales continuas y discretas

Una señal es una función en el tiempo $x(t)$ la cual generalmente describe fenómenos naturales, esta señal es la que se analizará en el SAD diseñado en este trabajo.

Generalizando todas las posibles señales que pueden existir se pueden clasificar en dos tipos: las señales analógicas o de tiempo continuo y las señales digitales o de tiempo discreto, las señales en tiempo continuo o analógicas son las que existen en los fenómenos naturales, son aquellas que pueden tomar cualquier valor en cualquier instante de tiempo, en cambio las señales digitales o discretas son aquellas que tienen valores en fragmentos de tiempo específicos, este tipo de señales no existen en los fenómenos naturales sin embargo son una transformación de las señales analógicas para poder comprender los fenómenos que representan estas y así poder manipularlas electrónicamente de tal manera que los dispositivos o sistemas electrónicos puedan estudiar el fenómeno natural, esto es debido a que en la electrónica digital sólo se comprenden 2 estados lógicos (*high* y *low*). Entonces el tipo de señal está determinada de acuerdo a la forma de la variable independiente que en este caso es el tiempo, es decir si el tiempo de muestreo es discreto entonces la señal será digital, y si es continuo la señal será analógica [Figura 1.1].

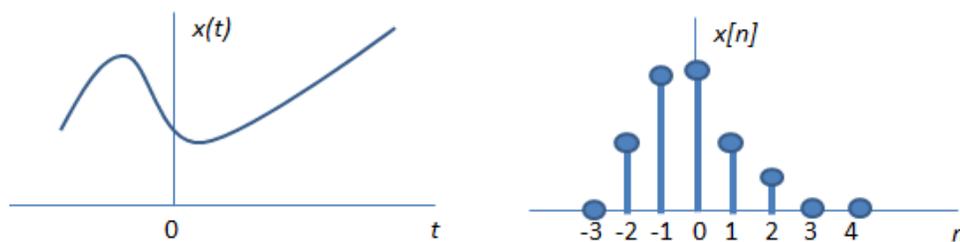


Figura 1.1: Señales en tiempo continuo y en tiempo discreto.

Morón, J. , dice que: *Una señal de tiempo discreto $x[n]$ puede representar un fenómeno para el cual la variable independiente es inherentemente discreta. Una señal de tiempo discreto, $x[n]$, también puede obtenerse mediante el muestreo de una señal de tiempo continuo $x(t)$ para obtener los valores.*

$$x(t_0), x(t_1), \dots, x(t_n), \dots$$

a los valores x_n se les denomina muestras; el intervalo de tiempo entre muestras se llama el intervalo de muestreo. Cuando estos intervalos son iguales (muestreo uniforme), entonces:

$$x_n = x[n] = x[nT_s]$$

Donde T_s es el intervalo de muestreo. Un dispositivo que convierta información analógica a forma digital mediante redondeo se denomina un convertidor analógico-digital. [10]

Señales periódicas y no-periódicas

Las señales pueden ser clasificadas de acuerdo a la periodicidad con la que se presenta la información y aunque parezca imposible la mayoría de los fenómenos físicos y químicos que existen en la naturaleza presentan algún tipo de periodicidad y es raro encontrar algún fenómeno que sea aperiódico.

Oppenheim, A. & Willsky, A. [12] dicen que: *“...Una señal periódica de tiempo continua $x(t)$ tiene la característica de que existe un número positivo T para el cual:”*

$$x(t) = x(t + T)$$

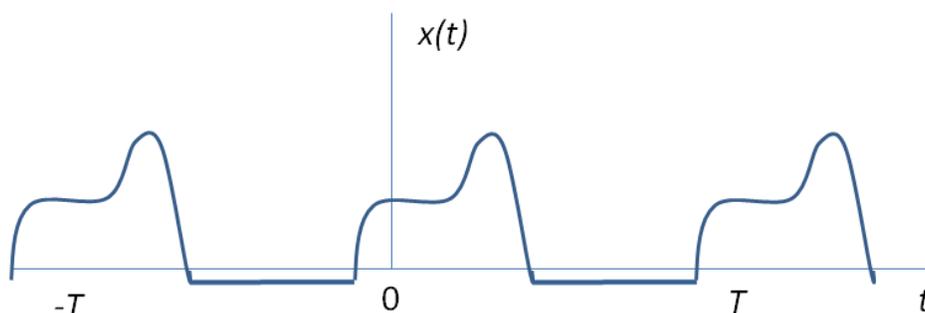


Figura 1.2: Ejemplo de una señal periódica.

En la figura 1.2 se puede apreciar una señal periódica con periodo T , esta se repite con un mismo patrón durante un tiempo múltiplo de T y continúa haciéndolo por tiempo infinito, pero esto sólo ocurre cuando la función $x(t)$ es variable. De esta manera la frecuencia fundamental está denotada por: $1/T$, y su unidad de medida son los Hertz (ciclos por segundo). A las señales que no se repitan cíclicamente en un cierto periodo de tiempo se les llama aperiódicas o simplemente señales no periódicas.

Acondicionamiento de señal

Las señales de salida de muchos transductores son señales con amplitudes y corrientes muy pequeñas, como por ejemplo, las señales provenientes de transductores bio-eléctricos. Estas señales se requieren transmitir para ser manejadas adecuadamente por el sistema, pero debido a su amplitud pueden adquirir ruido del medio ambiente o del propio equipo de adquisición de datos, para esto se utilizan instrumentos especializados en adecuar la señal a una señal más óptima para su manejo, por ejemplo para reducir el ruido se pueden implementar filtros de segundo orden, implementar un amplificador operacional cuando se requiere amplificar la señal, o incluir ambas opciones utilizando un amplificador de instrumentación, que no sólo limpia la señal del ruido sino que además la amplifica.

- Amplificador de instrumentación

Los amplificadores de instrumentación tienen una ganancia finita, tienen una entrada diferencial de alta impedancia, una característica particular de los amplificadores de instrumentación es que poseen un rango de voltaje alto y un rechazo en modo común también alto.

Configuración de un amplificador de instrumentación [Figura 1.3]

En su libro Copper, W. & Helfrick, A. [3] describen el funcionamiento de la configuración del amplificador de instrumentación como se muestra a continuación:

El voltaje en la entrada inversora del amplificador de entrada inferior es $v_{entrada^-}$ y por lo tanto v_1 es:

$$v_1 = \left(\frac{R_1}{R_2}\right)(v_{entrada^-} - v_{entrada^+})$$

De forma semejante v_2 es:

$$v_2 = \left(\frac{R_1}{R_2}\right)(v_{entrada^+} - v_{entrada^-})$$

La etapa de salida es un amplificador diferencial por lo tanto v_{salida} es:

$$v_{salida} = \left(\frac{R_3}{R_4}\right)(v_2 - v_1) = 2\left(\frac{R_1R_3}{R_2R_4}\right)(v_{entrada^+} - v_{entrada^-})$$

Para reducir la captación de voltaje de ruido en las conexiones entre el transductor y amplificador de instrumentación, las terminales al transductor serán tan cortas como sea posible y la señal amplificada se transmite a la distancia requerida.

Existen situaciones donde el ruido ambiental es tan crítico que los amplificadores convencionales no soportan los niveles de ruido que ahí se encuentra, en estos casos se puede utilizar un amplificador de aislamiento para prevenir las peligrosas señales de ruido de alto voltaje que llegan al equipo de adquisición de datos.

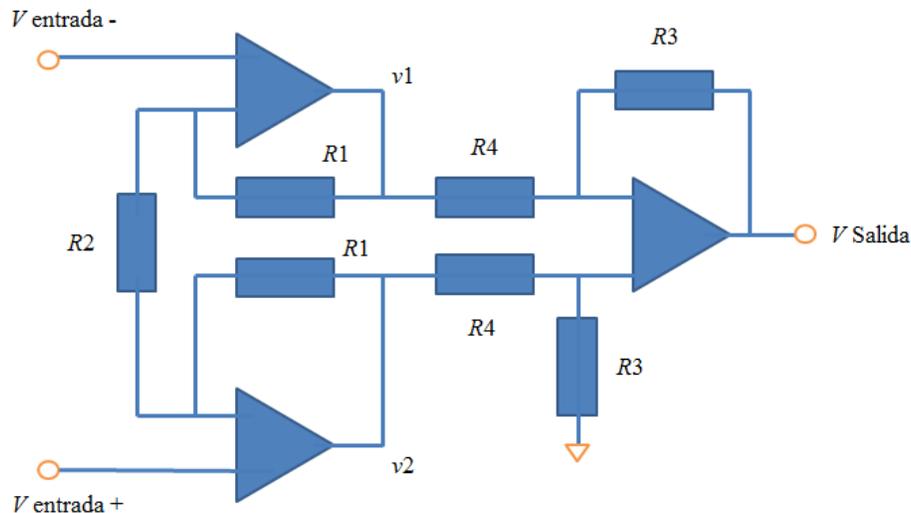


Figura 1.3: Configuración de un amplificador de instrumentación.

Con base en los conceptos anteriores se puede decir que un sistema de adquisición de datos mide señales provenientes de fenómenos físicos los cuales son interpretados por una computadora responsable de mostrar dichos datos de manera que los podamos entender.

Hoy en el mercado en el campo de la instrumentación existen diferentes compañías encargadas de elaborar y dar mantenimiento a los SAD, estas compañías desarrollan *software* y *hardware* especializado para obtener señales provenientes del medio físico y procesarlas, tales como LabVIEW, Measurement Studio para Visual Studio, LabWindows/CVI para Lenguaje C, MATLAB. La mayoría del *software* por lo general debe de ser adquirido con el *hardware* del mismo, tal es el caso de la compañía National Instruments que se especializó en los SAD o como ellos lo nombran “*Data acquisition*” (DAQ), por otra parte el *software* MATLAB no es un *software* que tenga como fin los SAD sin embargo y debido a su gran poder con el manejo de matrices, se utiliza para hacer cálculos que impliquen una gran precisión o un gran número de elementos involucrados, este *software* resulta importante para este trabajo por el complemento *Simulink*, el cual permite simular y procesar señales a nivel de *software* en un entorno de programación gráfico, para después ser procesadas en MATLAB.

Según el sitio oficial web de National Instruments LABVIEW cuenta con más de 30 años de experiencia, DAQ de National Instruments es el *hardware* de medidas basado en PC (*Personal Computer*) más confiable [28], sin importar el tipo de aplicación, ya sea para medidas básicas o sistemas complejos. Este sistema puede ser una alternativa si se desea adquirir un SAD respaldado por una compañía que se dedica a el desarrollo de este tipo de sistemas, sin embargo el costo y mantenimiento de estos dispositivos es exageradamente alto para personas que sólo buscan medir algunas señales y llevar un registro de éstas, por lo cual en el trabajo de esta tesis se utilizan herramientas de vanguardia tecnológica, relativamente nuevas en el mercado (tal es el caso de la Raspberry-Pi) además de ser de los dispositivos más utilizados es sistemas de control (tal es el caso de Arduino). Estas tecnologías serán tratadas más a fondo en el Capítulo 2 “*Conceptos principales*”.

Conforme avanza la tecnología cada vez se hace más rápido el tiempo de respuesta de los dispositivos, así como el procesamiento de información, también con este avance las computadoras se están convirtiendo en computadoras de bolsillo, es decir ya no tenemos que sentarnos frente a un escritorio a trabajar en la computadora, ni siquiera en la actualidad tenemos que llevar una *laptop* para poder realizar trabajos o tareas, ya que desde los dispositivos móviles (celulares y tabletas) podemos realizar dichas tareas, es decir las computadoras convencionales (computadoras de escritorio), se están quedando atrás, una prueba de esto es este trabajo, en el que se desarrollara una aplicación con los dispositivos Arduino y Raspberry-Pi que tienen un bajo costo, y debido a su tamaño y diseño tienen una excelente portabilidad y comunicación con otros dispositivos.

CAPÍTULO 2: CONCEPTOS PRINCIPALES

En este capítulo se describirán los conceptos principales: Sistema de Adquisición de Datos, microcomputadoras y microcontroladores; También los dispositivos y herramientas utilizadas en el desarrollo del trabajo: Arduino, Raspberry-Pi y bases de datos.

2.1 SISTEMA DE ADQUISICIÓN DE DATOS

2.1.1 Definición

“Los sistemas de adquisición de datos, son dispositivos creados para adquirir y almacenar, procesar o transmitir grandes cantidades de información de una forma autónoma.” [9].

Todos los sistemas de adquisición de datos se utilizan para capturar señales del mundo exterior, para analizar su contenido, para realizar alguna acción o simplemente para llevar un registro, estas señales pueden provenir de dos tipos de fuentes:

- Las que provienen de fuentes directas que principalmente generan señales eléctricas.
- Las que provienen de transductores, estos convierten alguna señal del exterior en una señal que pueda ser analizada por el sistema (generalmente son convertidas en señales eléctricas).

De acuerdo al tipo de señales que analiza un sistema, se pueden clasificar en dos tipos, sistemas analógicos: son aquellos que analizan señales continuas las cuales pueden tener valores que están dentro del campo de los números reales y sistemas digitales: son aquellos que analizan señales de forma discreta, en este tipo de señales se obtienen una sucesión de valores discretos los cuales son representados por el sistema de acuerdo al tiempo en el que fueron capturados, este tipo de sistemas solamente reconocen dos tipos de valores: voltaje de corriente directa (V_{CD}) representado por una carga de $5[V]$ y voltaje de tierra (V_{GND}) donde el valor teórico es $0[V]$.

2.1.2 Elementos

En su libro D. Copper & D. Helfrick dice que: *“...Un sistema de adquisición de datos analógico consta de algunos o de todos los elementos siguientes...”*:

- *Transductores para la transformación de parámetros físicos en señales eléctricas.*
- *Acondicionadores de señales para la amplificación, modificación o selección de ciertas partes de estas señales.*
- *Dispositivos de presentación visual para el monitoreo continuo de las señales de entrada. Estos dispositivos pueden incluir osciloscopios, panel de medidores, desplegados numéricos, etc.*

- *Instrumentos de registro de gráficas para obtener un registro permanente de los datos de entrada. Como memorias o bases de datos.*

Un sistema de adquisición de datos digital:

- *Transductor. Transforma parámetros físicos en señales eléctricas aceptables para el sistema de adquisición.*
- *Acondicionador de señal. Por lo general incluye la circuitería de soporte para el transductor. Esta circuitería puede proporcionar la energía de excitación, circuito de equilibrio y elementos de calibración.*
- *Explorador o multiplexor. Acepta múltiples señales analógicas y las conecta secuencialmente en un instrumento de medición.*
- *Convertidor de señal. Transforma la señal analógica en una forma aceptable para el convertidor analógico digital.*
- *Convertidor analógico digital(A/D). Convierte el voltaje analógico a su forma digital equivalente.*
- *Equipo auxiliar. Contiene instrumentos para funciones de programación de sistemas y procesamiento digital de los datos.*
- *Registrador digital. Responsable de registrar los datos para que se pueda realizar un estudio o análisis de la información. [3]*

Debido al tipo de señal que se puede analizar en un SAD, los sistemas analógicos son utilizados cuando no se requiere precisión en el dato capturado o por el ancho de banda amplio de la señal a analizar, es decir que el rango de valores de la señal es extenso. A diferencia de los sistemas digitales que toma valores discretos de señales con un ancho de banda angosto por lo que la precisión de estos sistemas es indispensable para analizar la señal. [Figura 2.1]

En el libro D. Copper & D. Helfrick se comparan los sistemas digitales y analógicos: “...Los sistemas digitales varían en complejidad desde sistemas de un solo canal de medición y registro de voltajes de cd hasta sistemas automáticos de múltiples canales, los cuales miden un gran número de parámetros de entrada, los comparan con respecto a condiciones o límites preestablecidos y llevan a cabo cálculos y toman decisiones sobre la señal de entrada. Los sistemas digitales en general son más complejos que los analógicos, tanto en términos de volumen y complejidad de los datos de entrada que pueden manejar...”. [3]

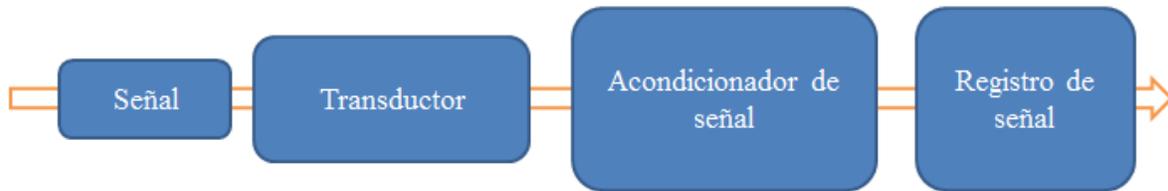


Figura 2.1: Sistema básico de medición.

2.1.3 Características

A continuación se describen algunas características que deberían tener los SAD sin importar el propósito para el cual fueron diseñados:

Canales de adquisición:

Los canales de adquisición son vías por las cuales un SAD obtiene información del exterior, dependiendo del tipo de canal, es decir del tipo de señal que capta, ya sea una señal discreta o una señal continua, pueden ser canales digitales o canales analógicos. Es importante considerar el número de canales disponibles para la adquisición de datos y el número de canales disponibles para la comunicación de la información entre el dispositivo y el entorno (por ejemplo alguna interfaz, almacenamiento en memoria, etc.). El SAD deberá tener el número de canales necesarios para la adquisición de información, es decir, de acuerdo a las necesidades el sistema deberá contar con un número de canales analógicos que soporten en la entrada del sistema la señal de salida de un sensor o transductor.

Autonomía

Es la capacidad del sistema de funcionar sin la intervención del ser humano, está determinada por factores como el consumo de electricidad, capacidad y límite de almacenamiento, efectividad de canales y medios de transmisión, capacidad de corrección de errores (propios del sistema, por ejemplo que el *software* caiga en un *loop* infinito) etc., es decir si alguna de estas características falla es necesario la intervención del ser humano para corregir errores o dar mantenimiento al sistema.

Capacidad de almacenamiento

La capacidad de almacenamiento se refiere al volumen de información que un sistema puede almacenar, generalmente los sistemas utilizan memorias de acceso secuencial, sin embargo en el SAD desarrollado en este trabajo, la parte de almacenamiento de información se dejará aparte, dejándolo sólo como instrumento de captura y envío de información, para después utilizar una base de datos para el almacenamiento de dicha información.

Capacidad de comunicación

Es la capacidad de transmitir la información requerida a otros entornos o sistemas, por ejemplo un archivo de texto, una memoria MicroSD, un servidor de base de datos u otro sistema. Por lo general

un SAD captura datos y recibe instrucciones para almacenarlos en alguna parte (archivos, memorias, o en base de datos), sin embargo en este trabajo se pretende que estas instrucciones sean invisibles para el usuario y que el sistema se encuentre continuamente adquiriendo y almacenando información, con lo que se pretende que este sea un sistema en tiempo real. El SAD deberá tener la capacidad para comunicarse con otros dispositivos, es decir deberá utilizar un estándar para la comunicación al exterior, esto con la finalidad que otros dispositivos que utilicen dicho estándar puedan obtener la información e interpretarla adecuadamente, por ejemplo se puede estandarizar la velocidad de transmisión de datos, por ejemplo el número de *baudios* transmitidos por segundo.

Capacidad de procesamiento, decisión y acción

Los SAD están diseñados exclusivamente para obtener información del exterior e interpretarla, para luego enviarla a una computadora para que sea almacenada, estudiada o procesada. Debido al avance en microcontroladores y microcomputadoras en el mismo SAD la información puede ser procesada y con base a los resultados que se obtengan el sistema pueda determinar qué acciones tomar para evitar daños, corregir errores o informar a una persona para que este tome la decisión adecuada.

Velocidad

La velocidad es el tiempo en que el sistema puede tomar valores fiables de la señal medida, procesarla y arrojar un resultado integro. Generalmente esta velocidad está determinada por un cristal de cuarzo que tiene una oscilación natural que es utilizada como ciclo de trabajo del sistema.

Resolución

La resolución se refiere al rango de valores que el sistema puede capturar y procesar.

Costo

El costo es muy importante, debido a que es un factor crítico en la industria. El costo de la automatización de algún sistema debe ser una inversión rentable para que se considere su implementación. Para el caso de la comunidad académica, este no deja de ser factor de gran importancia, dado que cualquiera que desee elaborar un sistema para estudiar algún fenómeno debe considerar un costo no muy elevado para poder implementarlo.

Tamaño físico y portabilidad

El tamaño y la portabilidad también son muy importantes ya que deberá ser desarrollado de tal manera que pueda ser transportado y utilizado en cualquier parte.

2.2 MICROCOMPUTADORAS Y MICROCONTROLADORES

2.2.1 Definición

¿Qué es una computadora?

“La computadora es un dispositivo capaz de aceptar información, almacenarla, aplicarle un proceso y registrar los resultados de ese proceso” [2]

¿Qué es una microcomputadora?

Una microcomputadora es una computadora pequeña que utiliza un microprocesador como CPU (Unidad Central de Proceso), es pequeña a comparación de los servidores, estaciones de trabajo o *mainframes*.

¿Qué es un microcontrolador?

Es un circuito integrado programable capaz de ejecutar las instrucciones grabadas en su memoria, por lo general se utilizan para realizar tareas repetitivas y sencillas.

Debido a que la funcionalidad principal de una computadora es procesar datos para convertirlos en información se puede tomar la definición de computadora para definir al Arduino y a la Raspberry-Pi, ya que engloba perfectamente a las microcomputadoras y microcontroladores, sin embargo cada uno de estos dispositivos tiene características individuales que los diferencian uno del otro.

A continuación se describirá la diferencia entre ambos dispositivos (microcomputadoras y microcontroladores).

2.2.2 Diferencia

La principal diferencia entre estos dispositivos es que en un microcontrolador primero se graban las instrucciones que se desea que realice en una memoria que ya tiene integrada (no requiere de una memoria exterior para que funcione), es decir este es un dispositivo para propósito específico; en cambio una microcomputadora debe de contener un procesador o un microprocesador que sea capaz de ejecutar instrucciones en el momento que estas lleguen, las cuales no necesitan grabarse en el dispositivo para ser ejecutadas [Figura 2.2] .

Definitivamente la capacidad de las microcomputadoras supera por mucho a las de un microcontrolador, ya que este tipo de componentes es de propósito más general y no se tiene que grabar en cada momento las instrucciones que se desea ejecutar, sin embargo, se debe contar con una memoria de sólo lectura o ROM (*Read Only Memory*), y un bus de datos para que haya comunicación entre el procesador y las instrucciones que puedan ser ejecutadas y almacenadas en la memoria ROM, también se debe de contar con puertos de entrada y salida para que pueda haber comunicación con el exterior. En cambio un microcontrolador es un dispositivo capaz de grabar una serie de instrucciones en memoria para luego ejecutarlas cuando sean requeridas, debe de contar

con un CPU, una memoria, y canales de comunicación con los periféricos. El microcontrolador [Figura 2.3] deberá tener la capacidad de grabar instrucciones en una memoria interna programable EEPROM (*Electrical Erasable Programmable Read Only Memory*) por sus siglas en inglés o en una memoria flash interna, ya que ambas tienen la característica de borrar la información en menos de un segundo mediante corriente eléctrica, esto facilita las correcciones en el programa grabado.

El programa que se graba en memoria es el resultado de un proceso de compilación de algún lenguaje de más alto nivel cómo Java, C, e incluso en un lenguaje ensamblador, después de este proceso resultan una serie de instrucciones representadas por números hexadecimales que a baja escala son interpretados cómo unos y ceros por el microcontrolador, por lo general estas instrucciones tienen una extensión “.hex”.

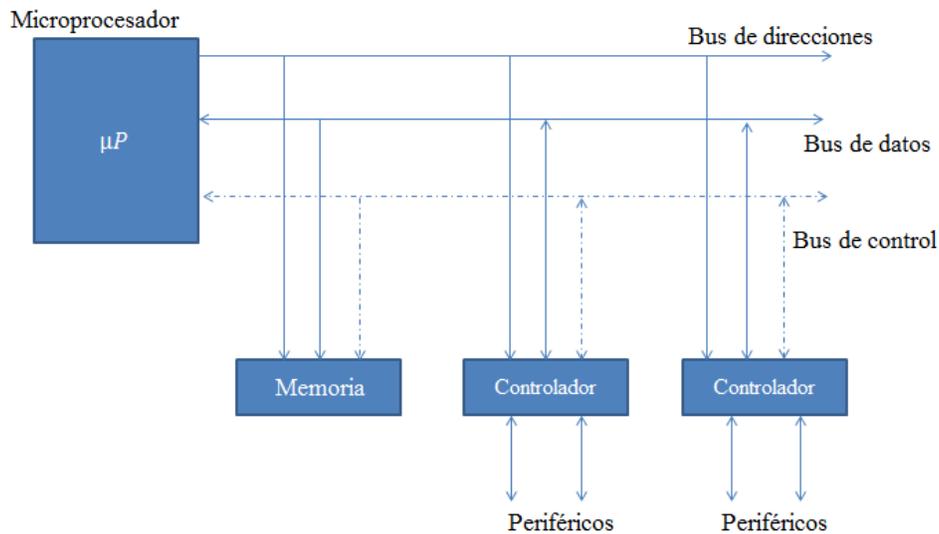


Figura 2.2: Estructura de un microprocesador.

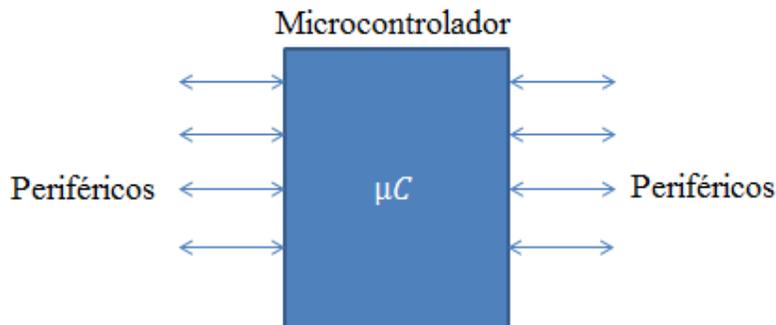


Figura 2.3: Estructura de un microcontrolador.

El funcionamiento de los dispositivos Arduino y Raspberry-Pi está basado en microcontroladores y microcomputadoras respectivamente. Arduino está basado en un microcontrolador (esa es la definición que se le da al circuito integrado ATmega), pero este dispositivo va más allá de ser un microcontrolador, ya que gracias a su amplia gama de librerías le es posible comunicarse al medio exterior mediante una gran variedad de placas a las que se denominan *shields*; Mientras que

Raspberry-Pi se describe como una microcomputadora, capaz de correr sistemas operativos basados en Linux.

La Raspberry-Pi se acopla más a la definición de una microcomputadora, sin embargo la gran capacidad de Arduino le hace acercarse también a las características que posee una microcomputadora por lo que en el desarrollo de este trabajo se englobarán bajo el mismo concepto (microcomputadoras).

Otra diferencia de estos dispositivos es la arquitectura de funcionamiento al momento de operar y ejecutar instrucciones, a continuación se describe los tipos de arquitectura de las computadoras en general y cual aplica para el Arduino y la Raspberry-Pi.

2.2.3 Arquitectura

Arquitectura Von Newmann

Arquitectura tradicional en donde el CPU, está conectado mediante un único bus de datos a una memoria en donde se guardan las instrucciones y datos del programa. Tal es el caso del ATmega el microcontrolador del Arduino, donde solamente existe un único bus de comunicación por lo que se reduce el tiempo de respuesta de estos dispositivos, debido que no puede buscar en memoria una nueva instrucción mientras no analicen las transferencias de datos de alguna otra instrucción, este tiempo es insignificante cuando se trata de un número reducido de instrucciones, y puede tener algunas limitantes. [Figura 2.4]

Aguayo, P. dice que: *Las principales limitaciones de la arquitectura Von Neumann son:*

- *La limitación de la longitud de las instrucciones por el bus de datos, que hace que el microprocesador tenga que realizar varios accesos a memoria para buscar instrucciones complejas*
- *La limitación de la velocidad de operación a causa del bus único para datos e instrucciones que no deja acceder simultáneamente a unos y otras, lo cual impide superponer ambos tiempos de acceso. [1]*

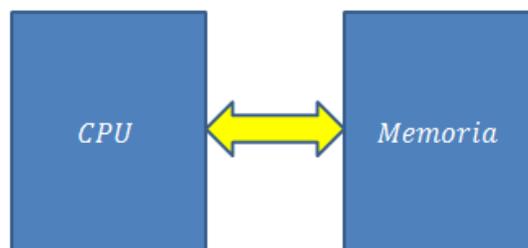


Figura 2.4: Arquitectura Von Neumann.

Arquitectura Harvard

Esta arquitectura fue diseñada para reducir el tiempo de respuesta de los dispositivos programables ya que cuenta con CPU, conectado a dos memorias: una encargada de las instrucciones y otra encargada de los datos. Ambas memorias cuentan con un bus de datos independiente para la comunicación con el CPU, estos buses independientes pueden tener distinta anchura en la palabra de bits, al contar con buses independientes el CPU puede acceder a los datos simultáneamente cuando está accediendo a las instrucciones y así reduciendo el tiempo de respuesta del dispositivo. Tal es el caso de la Raspberry-Pi que al ser una microcomputadora más compleja cuenta con este tipo de arquitectura que tiene ventajas sobre la arquitectura Von Neumann. [Figura 2.5]

Aguayo, P. menciona que: *algunas de las Ventajas de esta arquitectura son:*

- *El tamaño de las instrucciones no está relacionado con el de los datos, y por lo tanto puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria de programa, logrando así mayor velocidad y menor longitud de programa.*
- *El tiempo de acceso a las instrucciones puede superponerse con el de los datos, logrando una mayor velocidad en cada operación. [1]*

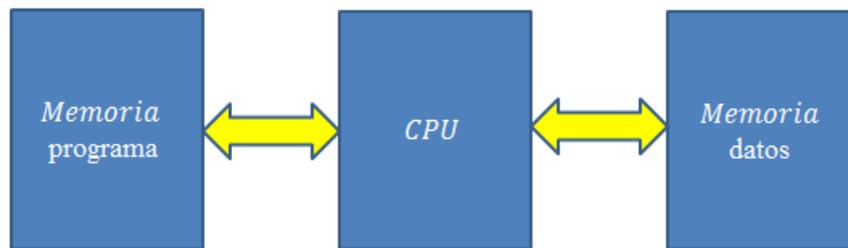


Figura 2.5: Arquitectura Harvard.

A continuación se describen elementos relevantes que se considera deben de tener las microcomputadoras.

2.2.4 Elementos

El procesador

Es el elemento más importante de las microcomputadoras ya que dictaminan las características de esta. Es el responsable de codificar y ejecutar instrucciones, direccionar a localidades de memoria los datos, comunicación de los periféricos, almacenamiento de datos. Específicamente se habla del ARM1176JZF-S para la Raspberry-Pi y los ATmega328, ATmega168 y ATmega8 dependiendo de la tarjeta de Arduino que se elija.

ALU

“...Una Unidad Lógica Aritmética (ALU) es una función multioperación digital de lógica combinacional. Esta puede realizar un conjunto de operaciones aritméticas básicas y un conjunto de operaciones lógicas...” [11]

Parte lógica del CPU incluso se considera el núcleo de este, su funcionamiento se basa en una serie de líneas de control, las cuales seleccionan que operación debe realizar, así toma el valor de las dos entradas y devuelve el resultado de la operación. Además la ALU proporciona información acerca del resultado y lo ubica en registros específicos:

Z (*Zero*): El resultado es cero.

N (*Negative*): El resultado fue negativo de acuerdo al valor del bit más significativo

O (*Overflow*): La operación produjo un *overflow* o desbordamiento.

C (*Carry*): La operación generó un acarreo.

Registros

Los registros pueden ser el origen o el destino de una operación, incluso pueden ser registros dedicados, similares a los que se explicaron en la definición de la ALU.

Todos los registros pueden ser utilizados por el CPU, incluso estos valores pueden ser almacenados en la memoria, pero el acceso a los registros es mucho más rápido que el acceso a memoria.

Stack Pointer

El *Stack Pointer* o pila del programa es una memoria consecutiva que almacena la dirección de memoria de la siguiente instrucción a ejecutar esto cuando se utilizan *loops* o ciclos que le permite al CPU conocer cuál es la siguiente instrucción sin perder la continuidad de la lógica de programación.

Unidad de control

Es responsabilidad de la unidad de control determinar qué operación se va a ejecutar después de conocer la ruta de datos de manera secuencial, para esto se ocupa un registro llamado el Contador de Programa (PC), en este registro se almacena la dirección de la siguiente instrucción a ejecutar, así este registro trabaja a la par con el Registro de Instrucción (IR) en el cual se carga la instrucción en donde está apuntando el PC, se decodifica la instrucción y se establece la ruta secuencial para ejecutarse. Cuando se inicia la ejecución del programa o en un reset el PC tiene la dirección de memoria \$0000.

Set de instrucciones

- CISC (*Complex Instruction Set Computing*)

Los procesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas entre operandos situados en la memoria o en los registros internos.

- RISC (*Reduced Instruction Set Computer*)

En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo. La sencillez y rapidez de las instrucciones permiten optimizar el *hardware* y el *software* del procesador.

- SISC (*Simple Instruction Set Computing*)

Las instrucciones que utilizan este tipo de procesador son muy concretas especialmente destinadas a propósitos muy concretos.

Memoria

Aguayo, P. establece que: “...la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos...”. [1]

Se pueden distinguir los siguientes tipos de memoria.

- ROM (*Read Only Memory*) con máscara: Es un tipo de memoria no volátil en la cual su contenido es generado durante la fabricación de la memoria.
- OTP (*One Time Programmable*): Memoria no volátil cuyo contenido se graba sólo una vez. Para grabarla se utiliza un grabador en donde eléctricamente se introduce las instrucciones del programa.
- UVPROM (*Ultraviolet Programmable Read Only Memory*): Memoria no volátil que puede ser grabada y borrada muchas veces, para grabarla se utiliza un grabador en donde eléctricamente se introduce las instrucciones del programa y para borrarla cuenta con una ventana de cristal en la cual si se deja pasar rayos UV (rayos ultravioleta provenientes del sol), estos borrarán los datos grabados.
- EEPROM (*Electrical Erasable Programmable Read Only Memory*): Memoria no volátil que puede ser grabada y borrada muchas veces, para grabarla y borrarla se utiliza un grabador en donde eléctricamente se introduce las instrucciones del programa.
- Flash: Memoria no volátil que puede funcionar como ROM o como RAM, pero tiene la ventaja de consumir menos electricidad y espacio, debido a estas características se están

sustituyendo las memorias ROM por este tipo de memorias que son más rápidas, de mayor densidad y más compactas.

Puertos E/S

“...Los puertos de entrada y salida permiten comunicar al procesador con el mundo exterior, a través de interfaces, o con otros dispositivos. Son la principal utilidad de los pines de un microcontrolador...” [1]

Por lo general cuentan con entradas digitales y analógicas, en donde las entradas digitales capturan el valor de la señal y lo representan como un “0” o “1” lógico según sea el caso, en cambio los puertos analógicos capturan el valor real de la señal, esto si no sobrepasa los límites de valores permitidos en los pines de entrada.

Estos puertos también pueden utilizarse como puertos de comunicación entre otros dispositivos, como por ejemplo el puerto UART (puerto serie asíncrona), USART (puerto serie síncrono y asíncrono), USB (Bus Universal Serial), etc.

Reloj principal

Para el funcionamiento de las microcomputadoras se requiere un ciclo de reloj para que puedan funcionar adecuadamente y sincronizar las operaciones del sistema, por lo general este ciclo es generado por un cristal de cuarzo, ya que estos tienen la propiedad de tener características piezoeléctricas y frecuencias muy altas.

Conversores A/D y D/A

La mayoría de las microcomputadoras contienen un convertidor analógico-digital (A/D) para procesar señales continuas, en ocasiones al contar con un solo convertidor A/D es recomendable tener un multiplexor que selecciona la señal a analizar y procesar; el convertidor digital-analógico (D/A) funciona de manera inversa, para mostrar los datos procesados (señales digitales) en una señal analógica, la cual puede utilizarse en actuadores junto a la electrónica de potencia.

Interrupciones

Las interrupciones son mecanismos de las microcomputadoras para responder a cambios de una variable crítica, que sirven para ejecutar instrucciones programadas previamente cuando la variable cambie. La respuesta puede ser tan sencilla desde el incremento de una variable hasta reiniciar la microcomputadora. Toda interrupción debe programarse y activarse en el programa principal. Mientras no haya cambio en la variable crítica el programa principal se ejecutara, cuando ocurra un cambio en la variable crítica la ejecución del programa pasara a ejecutar la interrupción.

Gridling, G. & Weiss Be. explican que: “...las microcomputadoras, microcontroladores y microprocesadores deben de ser desarrollados para sistemas que tienen que reaccionar a eventos...tan pronto como se produce el evento, el sistema responde llamando a una rutina de interrupción de servicio (ISR) que controla el evento. El ISR debe ser programada por el programador de la aplicación...” [8]

Temporizador

Se utiliza para llevar a cabo acciones cuando se desborda un contador o llega a cero. Para utilizarlo primero se declara el contador, su sentido (incremental o decremental), y el evento que dispara el contador (por ejemplo un cambio flanco en la señal). También funcionan para llevar a cabo retardos o *delay* con un tiempo preciso.

WatchDog

El *watchdog* o perro guardián es un sistema que evita que el *software* programado caiga en bucles infinitos o errores; consiste en un contador, que cuando se satura la variable que emplea reinicia el sistema.

Protección ante fallo de alimentación o *Brownout*

Es un circuito que resetea el sistema cuando el voltaje de alimentación es inferior al mínimo voltaje (*Brownout*).

Estado de reposo

Consiste en instrucciones que se pueden programar para que el sistema no consuma energía a menos que la requiera: por lo general los microcontroladores disponen de la instrucción *sleep* para indicarle al microcontrolador que entrara en el estado de reposo y bajo consumo.

2.3 ARDUINO

2.3.1 Definición

En el sitio web oficial, Arduino se describe como una interfaz entre la computadora y el mundo exterior [25]; sin embargo Arduino también se puede programar para que realice acciones de manera autónoma y eficiente. El microcontrolador de Atmel® usado por Arduino es también utilizado en los autos modernos de hoy en día para realizar acciones automatizadas, programadas previamente dentro de este microcontrolador. Dadas estas características, se puede definir como una microcomputadora de propósito específico para tomar decisiones de control para procesos industriales, análisis de señales, automatización de procesos repetitivos e interacción con el exterior.

2.3.2 La placa de Arduino Uno

En la figura 2.6 se puede observar la placa de Arduino Uno:

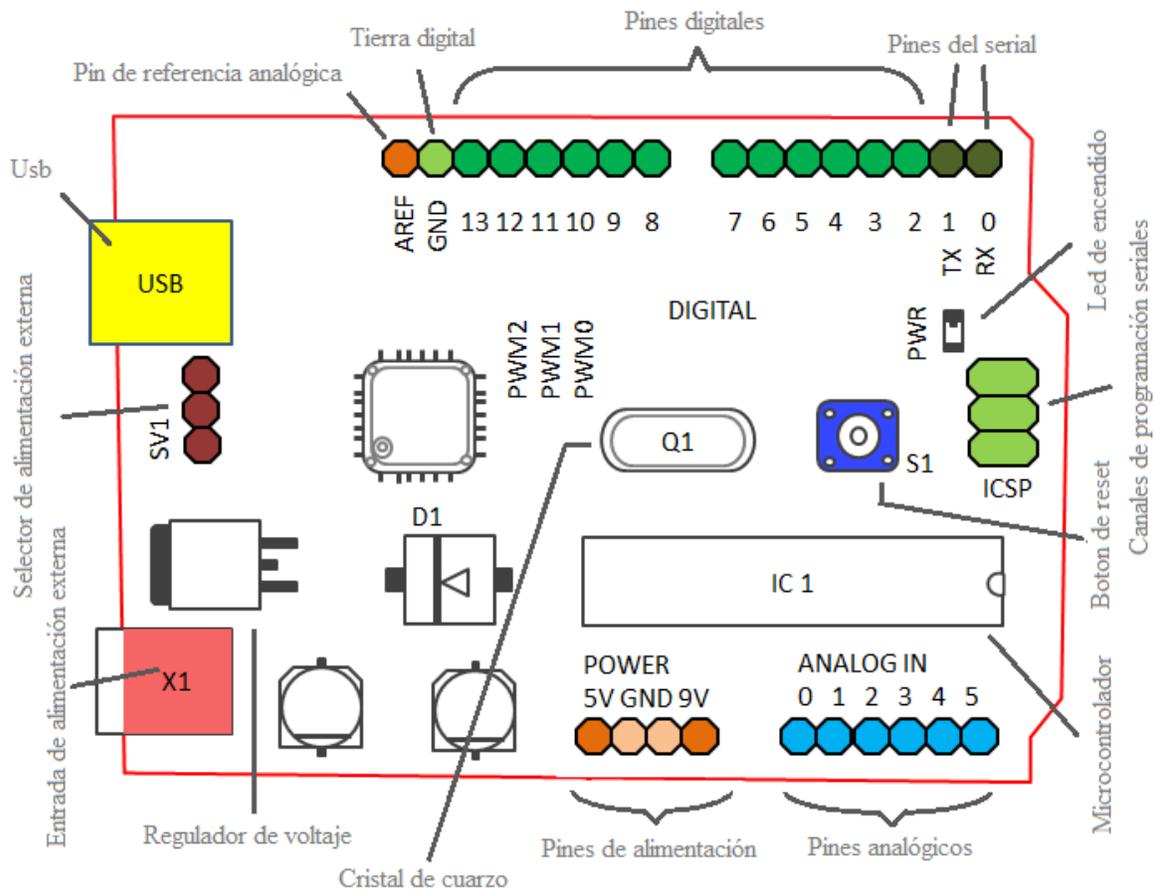


Figura 2.6: Esquema de la placa de Arduino Uno.

2.3.3 Microcontroladores

Las características de cada microcontrolador de los modelos de Arduino son las siguientes:

ATmega328 (usado en placas más recientes)

Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6 (DIP) or 8 (SMD)
DC Current per I/O Pin	40 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1KB

ATmega168 (usado en Arduino Diecimila y en Arduino Duemilanove)

Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6 (DIP) or 8 (SMD)
DC Current per I/O Pin	40 mA
Flash Memory	16 KB
SRAM	1 KB
EEPROM	512 bytes

ATmega8 (usado en placas de Arduino antiguas)

Digital I/O Pins	14 (of which 3 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
Flash Memory	8 KB
SRAM	1 KB
EEPROM	512 bytes

2.3.4 Elementos del Arduino

Pines digitales

Los puertos digitales pueden ser utilizados como entradas o salidas de propósito general a través de las funciones *pinMode()*, *digitalRead()*, y *digitalWrite()*. Cada terminal tiene una resistencia *pull-up* que puede activarse o desactivarse utilizando la función *digitalWrite()* (con un valor de *high* y *low*, respectivamente) cuando el pin está configurado como entrada. La corriente máxima por salida es 40 [mA].

- *Serial: 0 (RX) y 1 (TX)*. Utilizado para recibir (RX) y transmitir (TX) datos serie TTL. En el Arduino Diacemila, estas terminales están conectadas a las correspondientes pines del circuito integrado conversor FTDI USB a TTL serie. En el Arduino BT, están conectados a las terminales correspondientes del módulo Bluetooth WT11. En el Arduino Mini y el

Arduino LilyPad, están destinados para el uso de un módulo serie TTL externo (por ejemplo el adaptador Mini-USB).

- *Interrupciones externas: 2 y 3.* Estas terminales pueden ser configuradas para disparar una interrupción con un valor bajo, un pulso de subida o bajada, o un cambio de valor. Con la función *attachInterrupt()*.
- *PWM: 3, 5, 6, 9, 10, y 11.* Proporcionan salidas PWM (*Pulse Width Modulation*) de 8 bit con la función *analogWrite()*. En placas con ATmega8, las salidas PWM sólo están disponibles en los pines 9, 10, y 11.
- *Reset BT: 7.* (sólo en Arduino BT) Conectado a la línea de reset del módulo *bluetooth*.
- *SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).* Estas terminales soportan comunicación SPI. Aunque esta funcionalidad esta proporcionada por el *hardware*, no está incluida actualmente en las librerías estándar de Arduino.
- *LED: 13.* En el Diacemila y el LilyPad hay un led en placa conectado al pin digital 13. Cuando el pin tiene valor *high*, el led está encendido, cuando el pin está en *low*, está apagado.

Pines analógicos

Los pines de entrada analógicos soportan conversiones analógico-digital (A/D) de 10 bit utilizando la función *analogRead()*. Las entradas analógicas pueden ser también usadas como pines digitales: entrada analógica 0 como pin digital 14, hasta la entrada analógica 5 como pin digital 19. Las entradas analógicas 6 y 7 (presentes en el Mini y el BT) no pueden ser utilizadas como pines digitales.

- *I2C: 4 (SDA) y 5 (SCL).* Soportan comunicaciones I2C (TWI) utilizando la librería *Wire*.

Pines de alimentación

- *VIN* (a veces marcada como 9[v]). Es el voltaje de entrada a la placa Arduino cuando se está utilizando una fuente de alimentación externa (diferente de los 5[v] de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar voltaje a través de este pin. Las placas aceptan distintos rangos de voltaje de entrada. Pero el Arduino LilyPad no tiene pin VIN y acepta sólo una entrada regulada.
- *5V.* La alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador en placa o ser proporcionada por USB u otra fuente regulada de 5V.
- *3V3.* (sólo en el Diacemila) Una fuente de 3.3 voltios generada por el chip FTDI de la placa.

- *GND*. Pines de tierra.

Otros Pines

- *AREF*. Referencia de voltaje para las entradas analógicas. Utilizada con la función *analogReference()*.
- *Reset*. (sólo en el Diacemila) Se tiene que poner esta línea a *low* para resetear el microcontrolador. Utilizada típicamente para añadir un botón de reset a *shields* que bloquean el de la placa principal.

Hardware

- Duemilanove. Esta es la última revisión de la placa Arduino USB básica. Se conecta al ordenador con un cable USB estándar y contiene todo lo necesario para programar la placa. Se puede ampliar con gran variedad de *shields*: placas de extensión con funcionalidades específicas.
- Diecimila. Esta es la versión anterior de la placa USB básica.
- Nano. Una placa compacta diseñada para usar directamente en placas de desarrollo, el Nano se conecta al ordenador con un cable Mini-B USB.
- Mega. La más grande y potente placa Arduino, compatible con los *shields* de Duemilanove y Diecimila.
- Bluetooth. El Arduino BT contiene un módulo *bluetooth* que permite comunicarse y programarse sin cables. Es compatible con los *shields* de Arduino.
- LilyPad. Diseñado para aplicaciones sobre prendas, esta placa puede ser cosida a la ropa, es de color purpura.
- Fio. Diseñada para aplicaciones inalámbricas. Incluye una ranura para conectar el módulo XBee, un conector para baterías LiPo y electrónica para cargar baterías.
- Mini. La placa Arduino más pequeña. Funciona perfectamente en una placa de desarrollo o en aplicaciones donde el espacio es primordial. Se conecta al ordenador usando el adaptador Mini USB.
- Adaptador Mini USB. Esta placa convierte una conexión USB en 5 volts, toma tierra, líneas TX y RX que se puede conectar al Arduino Mini o a otro micro controlador.
- Pro. Esta placa está diseñada para aquellos que quieran dejar la placa incrustada en el proyecto: es más barata que la Diecimila y se puede alimentar fácilmente con baterías, pero requiere de componentes extra y montaje.
- Pro Mini. Como la Pro, la Pro Mini está diseñada para usuarios avanzados que requieren de bajo costo, menor tamaño y dispuestos a un poco de trabajo extra.
- Serial. Placa básica que utiliza una interfaz RS232 como comunicación con la computadora para programar o intercambiar datos. Esta placa es fácil de montar incluso como ejercicio didáctico.

En la tabla 2.1 se pueden comparar las placas de Arduino.

Tabla 2.1: Características de las placas de Arduino.

Nombre	µControlador	Voltaje de operación	Velocidad (CPU)	Entradas y salidas analógicas
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0
Micro	ATmega32u4	5 V/7-12 V	16 Mhz	12/0
Mini	ATmega328	5 V/7-9 V	16 Mhz	8/0
Nano	ATmega168 ATmega328	5 V/7-9 V	16 Mhz	8/0
Ethernet	ATmega328	5 V/7-12 V	16 Mhz	6/0
Esplora	ATmega32u4	5 V/7-12 V	16 Mhz	-
ArduinoBT	ATmega328	5 V/2.5-12 V	16 Mhz	6/0
Fio	ATmega328P	3.3 V/3.7-7 V	8 Mhz	8/0
Pro (168)	ATmega168	3.3 V/3.35-12 V	8 Mhz	6/0
Pro (328)	ATmega328	5 V/5-12 V	16 Mhz	6/0
Pro Mini	ATmega168	3.3 V/3.35-12 V 5 V/5-12 V	8 Mhz 16Mhz	6/0
LilyPad	ATmega168V ATmega328V	2.7-5.5 V/2.7-5.5 V	8 Mhz	6/0
LilyPad USB	ATmega32u4	3.3 V/3.8-5V	8 Mhz	4/0
LilyPad Simple	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0
LilyPad SimpleSnap	ATmega328	2.7-5.5 V/2.7-5.5 V	8 Mhz	4/0

Nombre	Entradas y salidas digitales	EEPROM [K Bytes]	SRAM [K Bytes]	FLASH [K Bytes]	USB	UART
Uno	14/6	1	2	32	Regular	1
Due	54/12	-	96	512	2 Micro	4
Leonardo	20/7	1	2.5	32	Micro	1
Mega 2560	54/15	4	8	256	Regular	4
Mega ADK	54/15	4	8	256	Regular	4
Micro	20/7	1	2.5	32	Micro	1
Mini	14/6	1	2	32	-	-
Nano	14/6	0.512 1	1 2	16 32	Mini-B	1
Ethernet	14/4	1	2	32	Regular	-
Esplora	-	1	2.5	32	Micro	-
ArduinoBT	14/6	1	2	32	-	1
Fio	14/6	1	2	32	Mini	1
Pro (168)	14/6	0.512	1	16	-	1
Pro (328)	14/6	1	2	32	-	1
Pro Mini	14/6	0.512	1	16	-	1
LilyPad	14/6	0.512	1	16	-	-
LilyPad USB	9/4	1	2.5	32	Micro	-
LilyPad Simple	9/4	1	2	32	-	-
LilyPad SimpleSnap	9/4	1	2	32	-	-

- *Shields*

Los *shields* son placas que se colocan encima de Arduino y que amplían funciones para que sean controladas desde Arduino, para controlar diferentes aparatos o adquirir datos.

- *Shield Xbee*: Este *shield* permite conectar inalámbricamente varios Arduino a distancias de 100 pies en edificios y 300 pies en el exterior usando el módulo Maxstream Xbee Zigbee.
- *Shield Motors*: Este *shield* permite a Arduino controlar motores eléctricos de corriente continua, servos y motores paso a paso y leer *encoders*.
- *Shield Ethernet*: Este *shield* permite a una placa Arduino conectarse a una red Ethernet y tener acceso a Internet.

Existen infinidad de *shields* para Arduino, incluso es posible desarrollar *shields* para tareas específicas.

Software

El *software* del que dispone Arduino es una serie de librerías que proveen funcionalidad extra al *sketch* o programa.

Librerías estándar

- EEPROM - Librería para leer y escribir en memorias.
- Ethernet - Librería para conectar a internet usando el Ethernet *Shield*.
- Firmata - Librería para comunicarse con aplicaciones en la computadora usando un protocolo estándar Serial.
- LiquidCrystal - Librería para controlar *displays* de cristal líquido (LCD)
- Servo - Librería para controlar servomotores.
- SoftwareSerial - Librería para la comunicación serial de cualquier pin digital.
- Stepper - Librería para controlar motores paso a paso (*stepper motors*).
- Wire - Interfaz de dos cables, o Two Wire Interface (TWI/I2C), para enviar y recibir datos a través de una red de dispositivos y sensores.
- Matrix - Librería para manipular *displays* de matrices de LED básicas.
- Sprite - Librería básica para manipulación de *sprites* para usar en animaciones con matrices de leds.

Librerías contribuidas

Estas librerías fueron desarrolladas por los fabricantes de *shields* específicos o programas de aplicaciones.

Comunicación (networking y protocolos)

- Messenger - Para procesar mensajes de texto desde la computadora.
- NewSoftSerial - Versión mejorada de la librería SoftwareSerial.

- OneWire - Controla dispositivos (de Dallas Semiconductor) que usan el protocolo One Wire.
- PS2Keyboard - Lee caracteres de un teclado PS2.
- Simple Message System - Envía mensajes entre Arduino y la computadora.
- SSerial2Mobile - Envía mensajes de texto o emails usando un teléfono móvil (vía comandos AT a través de software serial)
- Webduino - Librería de web server extendible (para usar con Arduino Ethernet *Shield*)
- X10 - Librería para enviar señales X10 a través de líneas de corriente AC.
- XBee - Librería para comunicaciones entre XBees en modo API.
- SerialControl - Librería para controlar remotamente otras Arduino a través de una conexión serial.

Sensores:

- Capacitive Sensing - Convertir dos o más pines en sensores capacitivos.
- Debounce - Para lectura de entradas digitales con ruido (por ejemplo botones).

Displays y LEDs:

- GLCD - Grafica rutinas para LCD basados en el chipset KS0108.
- LedControl - Para controlar matrices de leds o *displays* de siete segmentos con MAX7221 o MAX7219.
- LedControl - Alternativa a la librería Matrix para controlar múltiples leds con chips Maxim.
- LedDisplay - Control para el *shield* de LED HCMS-29.

Motores y PWM:

- TLC5940 - Controlador de PWM de 16 canales y 12 bits.

Medición de Tiempo:

- DateTime - Librería para llevar registro de fecha y hora actual en el *software*.
- Metro - Útil para cronometrar acciones en intervalos regulares.
- MsTimer2 - Utiliza timer2interrupt para disparar una acción cada N milisegundos.

Utilidades:

- TextString, también conocido como String - Maneja *strings*.
- PString - Liviana clase para imprimir en búfer.
- Streaming - Método para simplificar declaraciones de impresión.

La información fue obtenida desde el sitio web oficial de Arduino [15] que tiene licencia Creative Commons.

2.3.5 Aplicaciones

Existen infinidad de aplicaciones y usos que se le pueden dar a los Arduino: en la página oficial se muestran algunos ejemplos:

En la educación:

Toot es un juguete interactivo de sonido activo diseñado para niños de edades entre los 3 y 6 años de edad que requieran mejorar sus habilidades auditivas, de música y de idiomas. Fue desarrollado por Federico Lameri como su proyecto de tesis de Maestría en Diseño de Interacción en Supsi y prototipo utilizando Arduino Leonardo. [21]

En la música:

El *InfraHarp* es un generador de sonidos que utiliza detectores infrarrojos y que puede reproducir diferentes teclas en las principales escalas melódicas y armónicas, de menor importancia, con dos opciones de octava [20] [27] es una forma de crear música a partir de reproducir tonos al detectar señales infrarrojas.

En la Química:

AW(*Water activity meter*) es un medidor de cantidad de agua “libre” que pueden contener los alimentos, según el autor es una aplicación para verificar el estado de los alimentos, basándose en la teoría de que los hongos crecen y se desarrollan en ambientes húmedos, el medidor verifica qué tanta humedad contienen los alimentos [16] [19].

En Mecatrónica:

Mirror-hand es un proyecto en el cual se transmite el movimiento de una mano humana a una mano robótica. Para ello, la mano humana es colocada dentro de un guante diseñado especialmente para este proyecto, utilizando sensores de presión, motores, y Arduino [18] [26].

En Tecnología (Drones)

Este proyecto aún está en fase de pruebas, consiste en un robot que es capaz de controlarse autónomamente con la ayuda de Arduino, esto abre las puertas a vehículos autónomos (especialmente para aplicaciones militares) [17]

2.4 RASPBERRY-PI

2.4.1 Definición

De acuerdo con el sitio web oficial de Raspberry-Pi, esta microcomputadora se describe como un sistema de bajo costo, con una tarjeta cuya implementación permite conectarse a un monitor de computadora o a una televisión, además de ser capaz de usar un teclado y un mouse estándar [29]. Es un dispositivo pequeño que permite a las personas de cualquier edad explorar la computación, para aprender a programar en lenguajes como Python, C y Java. Es capaz de hacer todo lo que se espera de una computadora, desde navegar por Internet hasta reproducir vídeo de alta definición. También se puede utilizar para trabajar en hojas de cálculo, procesadores de texto, y jugar.

El primer acercamiento con la Raspberry-Pi es impactante, ya que las computadoras que se conocen tienen tamaños moderadamente grandes a comparación con la Raspberry (que tiene un tamaño de una tarjeta de crédito), exceptuando los celulares inteligentes y tabletas que hoy en día existen en los hogares, la capacidad que tiene este dispositivo es la de una computadora convencional, es escalable, fácil de utilizar (para usuarios con experiencia en Linux), y sobre todo tiene un bajo costo.

Una ventaja que tiene radica en que la distribución de *software* es de código abierto, con un sistema operativo basado en Linux. Es perfecta para la configuración de servidores, lo que significa que puede utilizarse para aplicaciones locales en donde involucre este tipo de equipo, sin necesidad de adquirirlos, (dado que el costo de un servidor es muy elevado) se da la posibilidad de desarrollar aplicaciones concretas, como una Raspberry-Pi funcionando como servidor de base de datos en una pequeña empresa. La Raspberry-Pi está diseñada para promover la educación y el conocimiento, perfecta para nosotros los universitarios. Esto se puede constatar en el sitio web que dice:

“La Fundación Raspberry Pi es una organización educativa benéfica registrada con sede en el Reino Unido. El objetivo de nuestra Fundación es promover la educación de los adultos y los niños, en particular en el campo de los ordenadores, informática y temas relacionados”. [29]

2.4.2 La placa de la Raspberry-Pi

La placa de la Raspberry-Pi tiene unas dimensiones aproximadas de aproximadamente de 8.5 x 5.6 [cm] con una ligera sobresaliente para insertar la microSD, y pesa aproximadamente 45[g], lo que la hace perfecta para su portabilidad. Tiene medidas similares a las de una tarjeta de crédito.

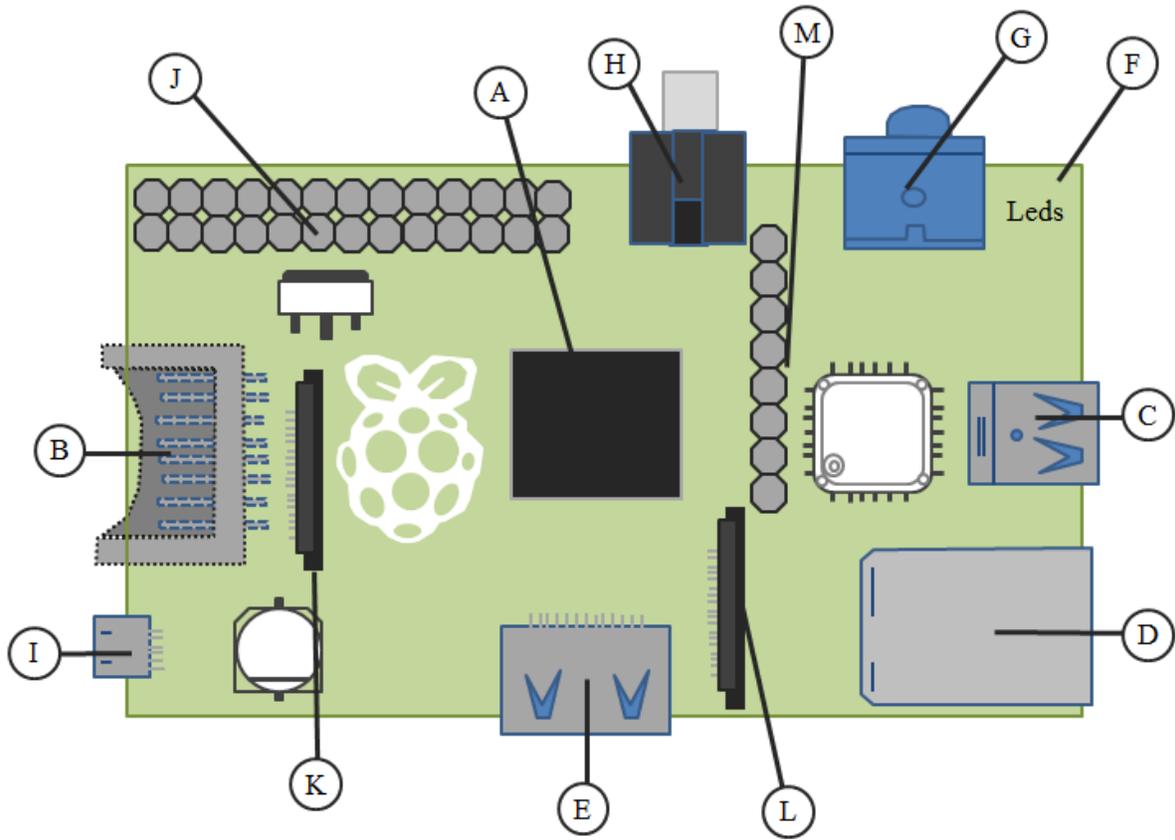


Figura 2.7: Esquema de la placa de la Raspberry-Pi.

En la Figura 2.7 se muestra la estructura de la placa de Raspberry-Pi, según la estructura de Richardson, M. & Wallace, S. (2013) [13].

A continuación se describen las partes que integran esta placa:

A-El procesador.

Es el corazón de la Raspberry Pi es el mismo procesador que podemos encontrar en el iPhone 3G y el Kindle 2, con esto se da una idea de las capacidades de la Raspberry Pi, en comparación con estos dispositivos. Es un chip que trabaja a un velocidad de 700[MHz] de 32 bits basado en la arquitectura ARM11. Estos chips tienen una gran variedad de arquitecturas con diferentes capacidades y diferentes precios. El Modelo "B" tiene 512 MB de memoria RAM y el modelo "A" tiene 256 MB. (La primera edición del Modelo B sólo tenía 256 MB de RAM.)

B- La ranura para la MicroSD.

Debido a que no existe disco duro la información se almacena en una memoria MicroSD.

C-Puerto USB.

En el modelo B, hay dos puertos USB 2.0, pero sólo uno en el Modelo A. Los primeros modelos de la Raspberry-Pi tienen limitantes en la corriente con la que operan, ya que algunos dispositivos USB

pueden demandar hasta 500[mA]. La placa original de la Raspberry-Pi daba soporte a 100[mA] o menos, pero las versiones más recientes cumplen con las especificaciones de USB 2.0. Se puede utilizar un Hub externo si se requiere en la demanda de más corriente.

D- Puerto Ethernet.

El modelo B tiene un puerto RJ45 Ethernet estándar. El Modelo A, carece de este, pero se puede conectar a una red cableada con un adaptador Ethernet USB (el puerto en el Modelo B es en realidad una adaptación del puerto USB a Ethernet). Aunque también existe la posibilidad de conectarse mediante conexión Wi-Fi.

E-Puerto HDMI.

El puerto HDMI ofrece vídeo digital y salida de audio, soporta 14 diferentes resoluciones, esta puede una desventaja porque las pantallas antiguas no cuentan con la entrada HDMI, pero se puede comprar un adaptador para que haya compatibilidad en la conexión; También se puede utilizar el puerto RCA.

F- Leds del status

ACT	Verde	Enciende cuando hay transmisión de la MicroSD
PWR	Rojo	Muestra que el sistema este alimentado con 3.3[v]
FDX	Verde	Enciende si hay conexión de tipo <i>full dúplex</i>
LNK	Verde	Muestra la actividad de la red
100	Amarillo	Enciende si hay conexiones de 100Mbps

G- Salida de audio analógica.

Cumpliendo con el estándar de 3.3 [mm] mini Jack de audio analógico, maneja cargas de alta impedancia como bocinas. Con audífonos no sonara muy bien pero se debe a un problema en el controlador que aún está en evolución.

H-Puerto RCA.

Proporciona señales PAL o NTSC. Este formato tiene una muy baja resolución en comparación con HDMI.

I-Power Input.

Entrada del suministro de corriente, esta tiene la entrada de una MicroUSB.

J- Entradas y salidas de propósito general (GPIO) y otros pines.

K- Interfaz de monitor serial (DSI). Este conector soporta 15 pines y puede ser utilizado para comunicarse con un LCD o un OLED.

L- Interfaz de cámara serial (CSI).

Puerto diseñado para conectar el módulo de la cámara directamente.

M- Pines para P2 y P3.

Son dos columnas de pines que sirven para realizar pruebas a los procesadores Bradcom (P2) y al LAN9512 (chip de conexión de red) (P3).

2.4.3 Elementos de la Raspberry-Pi

Hardware

Actualmente existen 3 modelos de la Raspberry-pi el modelo A, el modelo B y el modelo B+. El modelo A es la primera versión de la Raspberry-pi cuenta con 256MB de RAM, un puerto USB y no cuenta con puerto Ethernet, no consume tanta energía como el modelo B; el modelo B cuenta con 512 de memoria RAM, 2 puertos USB y también se le integra el puerto Ethernet; el modelo B+ es una mejora del modelo B, este cuenta con 4 puertos USB y se puede decir que es una versión con más velocidad que la anterior, esta versión salió en Julio del 2014.

Debido a que la Raspberry es un producto prácticamente nuevo en el mercado no cuenta con muchas expansiones o accesorios para ésta, o por lo menos desarrollados por los fabricantes de la Raspberry-Pi, el que destaca es la cámara que soporta 080p30, 720p60 y VGA90 para video.

Existen fabricantes como Adafruit que cuenta con diferentes módulos para esta, como por ejemplo el Mini Kit PiTFT que es una pequeña pantalla táctil de 2.8 pulgadas.

Debido a que la Raspberry es una microcomputadora, se le puede añadir como accesorio un teclado táctil inalámbrico, así como también un HUB de USB para que se le puedan conectar más dispositivos USB a esta.

Software

La Raspberry-Pi corre sobre el núcleo de Linux y sin embargo no cuenta con todas las características de algún sistema operativo Linux, debido a que su procesador está diseñado para dispositivos móviles. El procesador requiere especial atención en muchas instrucciones (*binary blob*) para drivers, y dispositivos especiales que no vendrán en distribuciones Linux estándar, otra limitante es la memoria RAM por lo cual están diseñadas distribuciones de Linux especiales para la Raspberry-Pi [Figura 2.8], algunas de estas son:

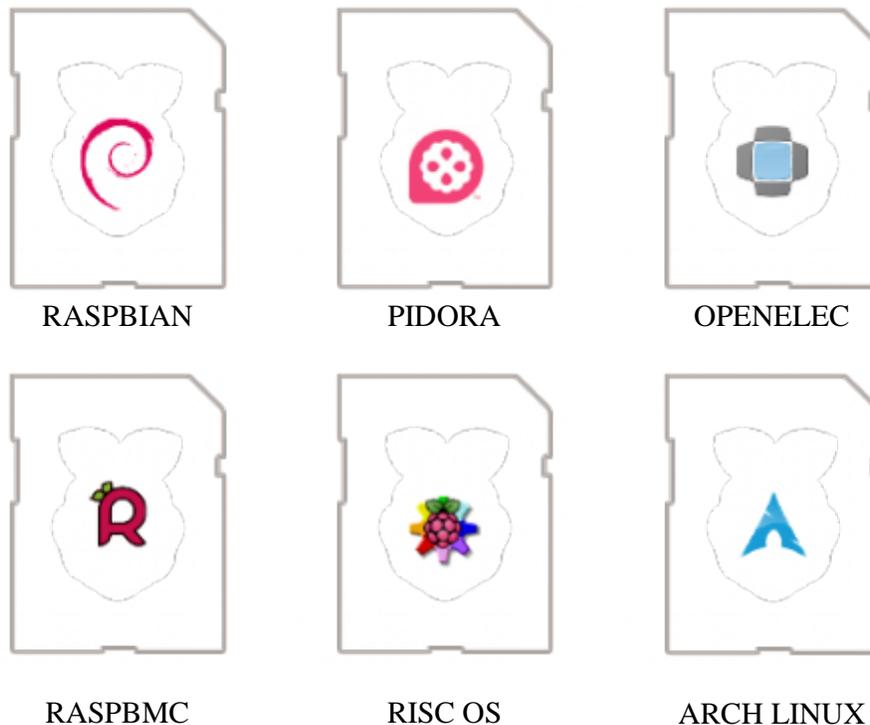


Figura 2.8: Sistemas operativos de la Raspberry-Pi.

Raspbian- Es la distribución recomendada para instalar en la Raspberry-Pi, es un sistema operativo basado en Debian optimizado para el *hardware* de la Rasperry-Pi. [24]

Pidora- Es una modificación optimizada de Fedora para la Raspberry-Pi. [23]

Openelec- Distribución Linux especializada para la Raspberry-Pi

Raspbmc- Es una distribución de Linux para la Raspberry-Pi basado en Debian.

Risc OS- Es un Sistema operativo británico diseñado especialmente para los procesadores ARM.[30]

Arch Linux- Sistema operativo diseñado para procesadores ARM. [14]

Los sistemas descritos anteriormente son distribuciones que recomienda la página oficial de la Raspberry-Pi, sin embargo existen muchos otros como por ejemplo: Adafruit Raspberry Pi y Educational Linux que es una versión de Adafruit que incluye librerías para manejar sus dispositivos.

Cuando Adquirimos la Raspberry se nos entrega sin sistema operativo, el sistema operativo se tiene que cargar una MicroSD donde después se colocará dentro de la ranura para esta. Para cargar el sistema operativo existen dos opciones: la primera es descargar el sistema de la página oficial y cargarlo a la tarjeta o utilizar un programa que nos brinde soporte para hacerlo, para el segundo caso se conocen dos, el primero es BerryBoot, y el otro es Noobs, ambos se deben de cargar en la MicroSD y nos guiarán para poder instalar el sistema operativo que deseemos.

La paquetería de la Raspberry, es compatible con *software* libre que podemos encontrar en cualquier distribución de Linux, tal como: procesadores de texto (Open Office), *software* de cálculo y gráfico (*Mathematica*), *software* de programación (Python), y *software* de desarrollo para aplicaciones de *hardware* (Arduino).

2.4.4 Aplicaciones

Las aplicaciones que se le quieran dar a la Raspberry dependen del alcance del proyecto. Debido a que tiene las funciones de una computadora con algunas limitantes (por ejemplo la memoria RAM disponible), es posible utilizarla para diversas aplicaciones como las que se describen a continuación:

Educación:

La principal aplicación que se desea dar a la Raspberry (según los desarrolladores) es para el campo de la educación y el desarrollo tecnológico, llevando estos dispositivos a escuelas para que los alumnos tengan un acceso a herramientas computacionales y se interesen en el campo de la computación.

Matemáticas e Ingeniería:

Se puede utilizar la paquetería que viene en el sistema operativo para hacer cálculos y operaciones matemáticas, por ejemplo se puede utilizar la paquetería *Mathematica* para realizar cálculos y visualizar gráficas.

Seguridad informática:

Debido a que el sistema operativo es un sistema basado en Linux se pueden instalar paquetería o herramientas para monitoreo y seguridad en la red, por ejemplo un escaneo de puertos como “*nmap*”, o incluso utilizar scripts basados en algún lenguaje como Python o Perl para realizar análisis en la red, en pocas palabras se puede hacer un “*pen-testing*” a otros equipos.

Servidor:

Si se desea que la Raspberry funcione como un servidor, ya sea servidor web, servidor de base de datos, servidor DNS, u otro, sólo se tiene que instalar los paquetes necesarios y dar de alta los servicios adecuados para que esta realice las funciones precisas.

Herramienta de programación y desarrollo:

La Raspberry tiene la capacidad de interpretar distintos lenguajes de programación, para la compilación y ejecución de programas, tales como Java, C, C++, Python, Perl, SWI-Prolog, etc., así como en el desarrollo de este trabajo de tesis para programar otros dispositivos (Arduino) con el IDE adecuado pero con la restricción de la memoria RAM, ya que algunos IDE requieren demasiados recursos para su ejecución.

2.5 ALMACENAMIENTO DE DATOS

Cuando se trabaja con computadoras forzosamente se requieren dispositivos que almacenen la información procesada, de manera que se pueda volver a utilizar en cualquier momento, esto para que los datos no se pierdan cuando se interrumpe la corriente eléctrica. En una memoria RAM los datos que se procesan se mantienen hasta que se interrumpe la corriente eléctrica, ya que este tipo de memoria requiere alimentación para almacenar la información. Sin dispositivos de almacenamiento en no se podría guardar la información de forma permanente y sería necesario cargar todos los programas y los datos cada vez que se encendiera la computadora o dispositivo. De ahí la importancia de dispositivos como la memoria ROM.

A continuación se muestra algunos ejemplos de dispositivos de almacenamiento en donde comúnmente se almacena información:

2.5.1 Dispositivos ópticos

Los dispositivos ópticos emplean una luz láser en lugar de un imán para leer y escribir bits de datos en una capa reflectante de silicio. Esta capa está protegida por una superficie de plástico permeable a la luz (CD, DVD, Blue-Ray).

2.5.2 Memorias de estado sólido

Son dispositivos cuyo uso ha ido en aumento en los últimos años debido a su bajo costo y a su gran capacidad para almacenar información.

Son capaces de almacenar la información sin la necesidad de que estén alimentadas por corriente eléctrica, a pesar de que estas características las hacen ideales para utilizarlas como memoria principal en las computadoras, no es así ya que tienen un límite en las veces que se puede escribir en estas, por lo que es difícil que reemplace a las memorias RAM.

2.5.3 Dispositivos magnéticos

Este tipo de dispositivos usan partículas cargadas eléctricamente sobre una superficie para almacenar información, y en función de cómo se encuentran orientadas pueden representar un uno o un cero lógico. Para poder leer o escribir información estos dispositivos cuentan con imanes que se encargan de dichas tareas.

Uno de los primeros dispositivos de este tipo fueron las cintas magnéticas que servían para almacenar grandes cantidades de información, sin embargo el inconveniente que presentaban es que accedían a la información de manera secuencial.

Por lo que las cintas se reemplazaron por los disquetes que fueron los primeros discos capaces de acceder de manera aleatoria a la información.

Hoy en día el dispositivo de almacenamiento en las computadoras es el disco duro, que está formado por una pila de discos rígidos metálicos magnetizados en cuyas superficies se concentra la información. Esta pila de discos se encuentra encerrada en una carcasa metálica para protegerla del exterior, ya que la más mínima partícula de polvo puede dañarlos. Cuando se depura la información, es decir cuando se borran datos del disco duro, quedan huecos que se ocupan al momento de guardar otros datos, de esta forma el disco duro no pierde tiempo en estar seleccionando el mejor lugar para almacenar la información.

Para el tema de este trabajo se hará más énfasis en el registro de los datos en el disco duro, en una base de datos de manera organizada y eficaz, para poder consultarlos y si es el caso, para poder realizar algún reporte o análisis de los datos capturados.

2.5.4 Bases de datos

“...Las bases de datos y los sistemas de bases de datos son un componente esencial de la vida cotidiana en estos tiempos. Actualmente, la mayoría de nosotros nos enfrentamos a diversas actividades que implican cierta interacción con una base de datos...”. [5]

La principal aplicación de las bases de datos es mantener la información ordenada y resguarda, para poder consultarla en cualquier momento, la información que se resguarda por lo general son cadenas de texto y variables numéricas. Sin embargo, con el avance de las tecnologías se puede resguardar más que eso, incluso las bases de datos se ocupan para controlar procesos industriales y de fabricación, junto con un sistema inteligente que haga consultas a ésta y se tome la decisión adecuada.

Algunos ejemplos del uso las bases de datos son: retirar dinero del banco, realizar alguna recepción, acceder a algún catalogo como por ejemplo el de una biblioteca, comercio, etc. Estas son actividades que requieren un sistema que interactúe con la base de datos y se obtengan las consultas deseadas.

Las bases de datos se utilizan en cualquier ámbito de estudio, ya sea para investigación, como por ejemplo que se almacenen datos estadísticos para un estudio posterior, se utilizan en las empresas para resguardar sus datos, en mismo internet especialmente en las redes sociales constantemente se están haciendo consultas a la base de datos de algún sitio, etc. En general la computación y las bases de datos van siempre de la mano.

“...Una base de datos es una colección de datos relacionados. Con la palabra datos nos referimos a los hechos (datos) conocidos que se pueden grabar y que tienen un significado implícito...”. [5]

Por ejemplo se pueden tener a los contactos con nombre, dirección y teléfono ubicados en algún programa o se pueden tener en algún *Directorio Telefónico*. Esta colección de datos relacionados con un significado implícito es una base de datos.

Elmasri, R. & Navathe, S. mencionan que una base de datos tiene las siguientes propiedades:

- Una base de datos representa algún aspecto del mundo real, lo que en ocasiones se denomina *minimundo* o *universo de discurso* (UoD, *Universe of discourse*). Los cambios introducidos en el *minimundo* se reflejan en la base de datos.
- Una base de datos es una colección de datos lógicamente coherente con algún tipo de significado inherente. No es correcto denominar base de datos a un *surtido aleatorio* de datos.
- Una base de datos se diseña, construye y rellena con datos para un propósito específico. Dispone de un grupo pretendido de usuarios y algunas aplicaciones preconcebidas en las que esos usuarios están interesados.
- Una base de datos debe ser una representación de un mundo más pequeño que forzosamente deberá tener usuarios que estén consultando información acerca de este *minimundo*, por lo que la base de datos siempre deberá de estar disponible, deberá brindar información fiable y deberá estar constantemente actualizada a los cambios que se presenten, otra característica importante es que deberá estar diseñada para que pueda ser escalable. [5]

Una base de datos se puede generar y mantener manualmente o estar computarizada. Un sistema de administración de datos DBMS (*Database Management System*) es una colección de programas que permite a los usuarios crear y mantener una base de datos. El DBMS es un programa que facilita los procesos de definición, construcción, manipulación y compartición de bases de datos entre varios usuarios y aplicaciones.

En la definición de una base de datos se especifican los tipos de datos, estructuras y restricciones que serán almacenados en la base de datos. La construcción implica en almacenar información en la base de datos para que esta pueda ser consultada. La manipulación consiste en consultar la base de datos para recuperar información específica, actualizar la base para hacer modificaciones y generar informes o reportes con base a la información recuperada. Por último se comparte la base de datos con los usuarios (que no sean administradores de ésta) mediante aplicaciones que puedan generar consultas y obtener resultados a partir de estas. [Figura 2.9]

Una aplicación puede acceder a la base de datos enviando consultas o solicitudes de registros al DBMS. Una consulta normalmente provoca la recuperación de los datos solicitados. Para el desarrollo de este trabajo se utilizara una base de datos de distribución libre (MySQL).

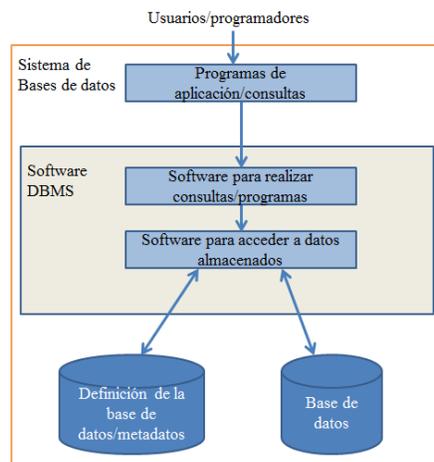


Figura 2.9: Entorno de un sistema de base de datos simplificado.

2.5.5 Aplicaciones

El almacenamiento de la información ha sido un tema de interés desde que se comenzaron a desarrollar las primeras computadoras, por lo que sería redundante especificar aplicaciones de cada uno de los dispositivos de almacenamiento descritos anteriormente. Una de las partes esenciales de este trabajo es el uso de una base de datos, por lo que a continuación se mencionan algunas aplicaciones:

Aplicaciones financieras:

En estos días es difícil encontrar a organismos financieros que guarden nuestro dinero sin que utilicen una base de datos, responsable de que se lleven a cabo las transacciones de forma segura y eficiente, así como para el almacenamiento de los datos como clientes.

Educación:

Prácticamente todas las instituciones educativas tienen o están mudando a una base de datos la información de sus estudiantes, ya que es una forma rápida de consulta de los registros que se tienen.

Empresas:

Un negocio que cuenta con una base de datos tiene mucha ventaja a un negocio que no la tiene porque se tiene de manera ordenada el registro de todos sus productos, así como todas las transacciones que se realizan en compra-venta de artículos.

Investigación

Un ejemplo concreto que se le da a las bases de datos es en este trabajo de tesis, donde se ocupa una base de datos para almacenar información de variables físicas para un estudio posterior, se ocupa para tener la información concentrada y luego hacer peticiones a ésta para que devuelva sólo los datos solicitados, por ejemplo, obtener los datos de algún día en específico, mostrar los registros en donde alguna variable física se salió del rango permitido, etc.

Prácticamente las bases de datos se utilizan en sectores donde se requiera almacenar información de manera ordenada, y más cuando el número de registros es enorme, como por ejemplo en hospitales, sectores gubernamentales, bancos etc.

CAPÍTULO 3: DESARROLLO DEL PROYECTO

En este capítulo se describirá cómo se desarrolló el SAD, incluyendo la instalación de las herramientas necesarias para el mismo, se describirá la instalación en dos sistemas operativos; en Windows porque es el sistema operativo comercial que muchas personas tienen a su alcance; y en “Raspbian” que es el sistema más utilizado en la Raspberry-Pi. Por consiguiente se describen los pasos que se hicieron en Raspbian y también se describe la alternativa que se pudiera tomar en Windows.

Se optó por desarrollar el sistema también en Windows porque es la manera en que se podían hacer pruebas y desarrollo de una manera rápida y eficiente.

Para el desarrollo del SAD en este capítulo se requiere principalmente tres partes fundamentales:

- Los sensores
- Los actuadores
- El servidor web y/o de base de datos

Todas estas partes funcionan en conjunto para tener un desarrollo con un propósito general, si se desea tener sólo una parte del sistema se puede omitir alguna parte. Por ejemplo el Arduino puede funcionar adquiriendo datos físicos y como servidor web mostrando los resultados de los valores obtenidos vía Ethernet, y almacenar los valores capturados en una MicroSD; La Raspberry-Pi puede funcionar adquiriendo datos físicos (controlando los pines de propósito general), o sólo como servidor web y base de datos; Debido a que la mayoría de los transductores son diseñados para que respondan a cambios del medio en forma de señales eléctricas, los transductores se pueden utilizar por separado, pero en conjunto con un osciloscopio para visualizar los cambios en el medio.

El propósito de usar las microcomputadoras Arduino y Raspberry es diseñar un SAD que pueda ser utilizado para el estudio de fenómenos físicos, a continuación se muestra en la figura 3.1 un esquema del SAD de forma sencilla.

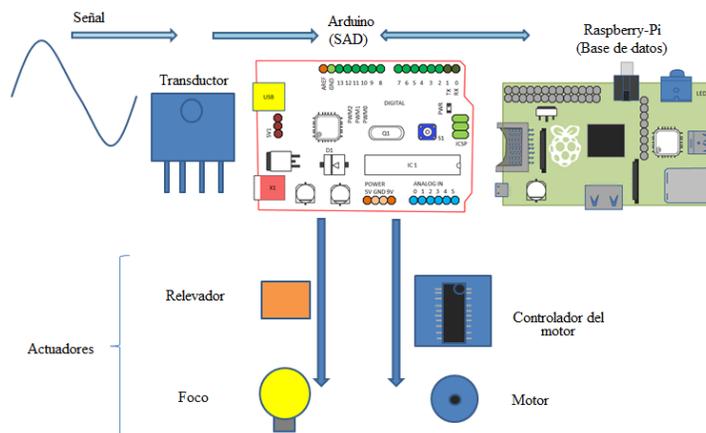


Figura 3.1: Esquema simple del SAD.

Para este caso el Arduino funciona adquiriendo datos y como servidor web ya que es el que proporciona la página web para visualizar los datos o para el control digital (Este tipo funcionamiento de Arduino se describe en el apartado 3.2.3 de este capítulo), la conexión al Arduino se hace con la Raspberry-Pi mediante el cable de red, aunque si se desea una conexión más allá de un red local (es decir una conexión utilizando Internet) se puede conectar las microcomputadoras a un dispositivo de red (HUB, *switch* o *router*) que esté conectado a Internet.

El siguiente diseño es un ampliación del primer diseño.

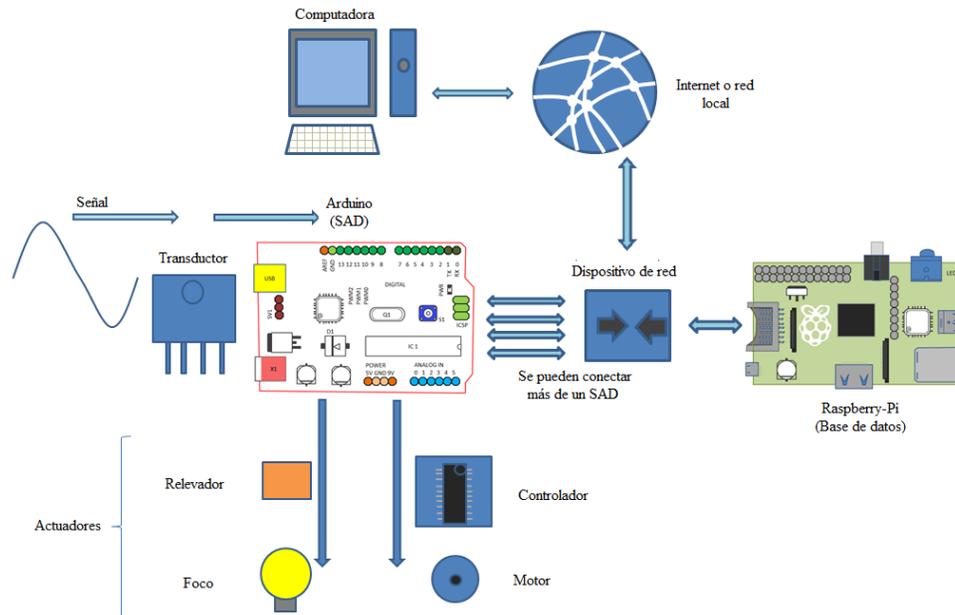


Figura 3.2: Ampliación y diseño general de un SAD.

Con el anterior diseño mostrado en la figura 3.2 se le deja la responsabilidad a la Raspberry-Pi de monitorear datos de diferentes tipos de SAD (establecidos en los Arduino) funcionando la Raspberry-Pi como servidor web y como servidor de base de datos, esto es lo menos recomendable por cuestiones de seguridad, se sugiere tener separado el servidor web y el de base de datos para evitar que el servidor sea atacado desde la aplicación web, sin embargo para cuestiones prácticas se desarrolló de esta manera.

Los dispositivos se pueden conectar a un dispositivo de red ya que la comunicación de estos se hace por el puerto Ethernet. El costo de este diseño sube considerablemente ya que se deberá contar con un Arduino por cada SAD que se desee, junto con su *shield* de internet, además de los sensores y actuadores específicos para cada medición. Debido a que la comunicación con el SAD es por el puerto Ethernet no sólo se requiere una dirección *IP* para cada SAD, en el primer o segundo diseño se deberá programar al Arduino con alguna dirección *IP* (una *IP* privada es lo más recomendable para que la *IP* pública quede alojada en la Raspberry-Pi) que se tenga libre, para esto se debe consultar con el administrador de la red.

La ventaja del segundo diseño con el primero es que si se cuenta con una *IP* pública se puede conectar al sistema desde cualquier parte del mundo utilizando internet y así monitorear y controlar los actuadores como sea requerido, en cambio en el primer diseño sólo tendrán acceso dispositivos que estén dentro de la red en donde se encuentran las microcomputadoras.

Antes de que se comenzara con la instalación del servidor Apache, es importante decir que la instalación fue hecha desde consola y que no se hizo en un entorno gráfico, esto es porque la mayoría de los servidores se administran de manera remota, utilizando herramientas que nos ayudan a hacer dicha conexión remotamente como por ejemplo la herramienta SSH Secure Shell para Windows.

Para conectarse remotamente se abre alguna herramienta que permita hacer la conexión al equipo (desde la Raspberry-Pi se también se puede hacer la conexión abriendo una terminal y se utiliza el comando `ssh`), se coloca la dirección *IP* y el usuario del mismo, luego se pedirá la contraseña de usuario para la conexión remota [Figura 3.3].

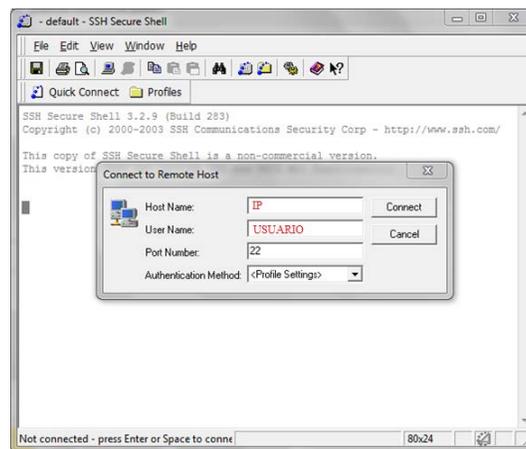


Figura 3.3: Secure Shell.

Antes de proceder fue necesario actualizar las librerías de descarga en el equipo, en este caso particular se utilizó un equipo Raspberry-Pi y se trabajó sobre el sistema operativo Raspbian, para fines prácticos se instalaron todas las herramientas desde súper usuario (usuario “*root*”), esto para evitar que hubiera alguna denegación del sistema a causa de permisos, o en su defecto se pudo anteponer el comando reservado “*sudo*” en las líneas de instalación, esto para lograr obtener los permisos necesarios.

Para esto primero se cambió a súper usuario, y se actualizaron las librerías:

```
USUARIO@raspberrypi:~$ su -
Password: █
root@raspberrypi:~# apt-get update
```

A continuación se describe como se instalaron las herramientas necesarias para el funcionamiento correcto del SAD.

3.1 INSTALACIÓN DE PLATAFORMA ARDUINO

3.1.1 Instalación de Arduino en Linux

Para la instalación de Arduino en el sistema operativo Linux fue necesario instalar las librerías y el IDE de Linux, con esto se instaló todo el entorno de desarrollo de Arduino es decir el IDE, a continuación se muestra el comando utilizado:

```
root@raspberrypi:/# apt-get install arduino
```

Otra opción pudo haber sido instalar solamente el compilador, las librerías responsables de la conexión con Arduino, el cargador del programa al Arduino y los archivos que interpretan instrucciones “*make*” (utilizadas para una compilación y carga de programa en el Arduino desde línea de comandos). Con esta opción no es necesario el entorno gráfico de Arduino.

```
root@raspberrypi:/# apt-get install gcc-avr avr-libc avrdude arduino-mk
```

3.1.2 Instalación de Arduino en Windows

Para la instalación del entorno de desarrollo de Arduino en Windows es necesario descargar el instalador del sitio web oficial: <http://arduino.cc/en/main/software> en este caso la versión del ejecutable que se puede a descargar es arduino-1.0.5-r2-windows.exe, al terminar la descarga se ejecuta y se procede a la instalación.

El *Wizard* apoya en la instalación, y solamente se tiene que aceptar los términos de licencia, seleccionar la carpeta en donde se instalara el IDE Arduino y continuar con la ejecución del instalador [Figura 3.4].

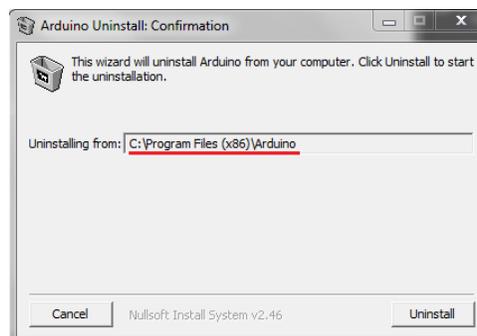


Figura 3.4: *Wizard* del instalador de Aduino.

3.2 INSTALACIÓN DEL SERVIDOR

3.2.1 Instalación del servidor Apache con php y MySQL en Linux

El servidor se instaló con el siguiente comando:

```
root@raspberrypi:/# apt-get install apache2
```

Se verificó que el servidor quedara funcionando correctamente, para esto se abrió un navegador y se escribió la dirección *IP* del equipo en donde quedó instalado el servidor apache [Figura 3.5].

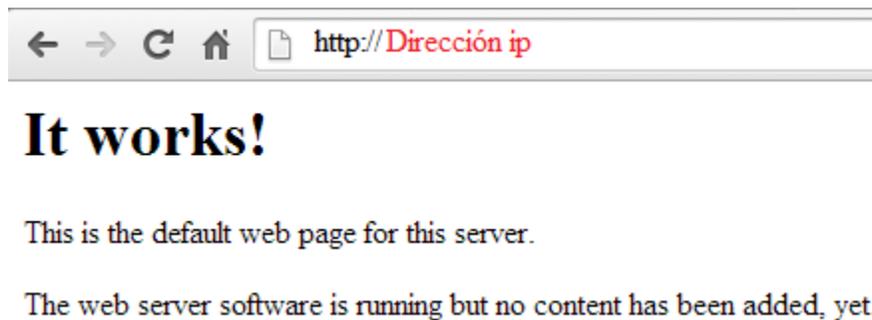


Figura 3.5: Funcionamiento correcto del servidor Apache.

Con el servidor funcionando, se instalaron después php y las librerías del mismo para su correcto funcionamiento en apache:

```
root@raspberrypi:/# apt-get install php5 libapache2-mod-php5
```

Para verificar el correcto funcionamiento de php se tuvo que crear un archivo con la extensión php, después se tuvo que mover de carpeta y crear el archivo con algún editor de texto (ya sea *vi* o *nano* que son los más comunes en el entorno de Linux).

```
root@raspberrypi:~# cd /var/www/
root@raspberrypi:/var/www# nano testphp.php
```

Se escribió en el archivo una función de php, en este caso se creó el archivo con nano [Figura 3.6].

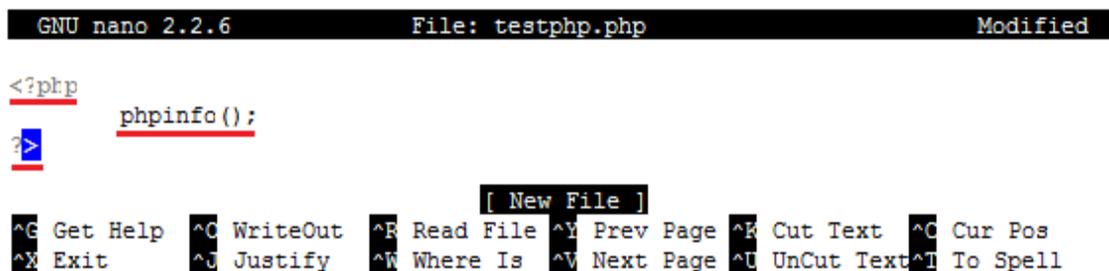


Figura 3.6: Archivo de prueba php.

3.2.2 Instalación del servidor Apache con php y MySQL en Windows

Quizá para un desarrollo de aplicaciones web en un entorno familiarizado (sistema operativo Windows) es recomendable utilizar herramientas que ya configuran todo el entorno del servidor web local para realizar pruebas. Existen infinidad de herramientas las cuales se pueden utilizar como alternativas para el desarrollo del proyecto, algunas de los cuales se pueden utilizar son: XAMPP, WAMP Server, BitNami, Mamp, EasyPhp, NMP Server, Portable WebServer, UwAmp, Microsoft WebMatrix, etc.

La ventaja de esto es que todo es administrado en un entorno gráfico amigable para el usuario, el paquete de herramientas que se seleccionó fue la de XAMPP.

Instalación de XAMPP:

Para la instalación de la herramienta XAMPP es necesario crear una carpeta en donde se ubicara el servidor. En este caso estará ubicada en C:\Program Files (x86)\XAMPP

Se descarga el instalador del sitio web: <https://www.apachefriends.org/es/index.html>, luego se ejecuta el instalador [Figura 3.9].

Como se pretende instalar XAMPP en la carpeta C:\Program Files (x86)\XAMPP será necesario deshabilitar por un momento el antivirus, así como también el UAC (*User Account Control*) de Windows, esto es debido a que al momento de instalar XAMPP escribirá archivos necesarios para su instalación.

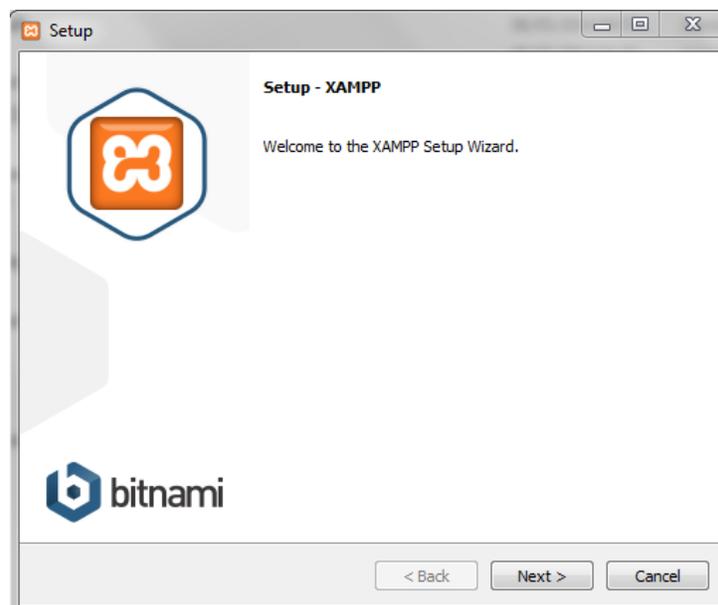


Figura 3.9: Instalador de XAMPP.

La mayoría de los instaladores configura todo el entorno automáticamente, y este no es la excepción, así que solamente se seleccionan las herramientas a instalar y el lugar en donde se instalaran, por ejemplo: C:\Program Files (x86)\XAMPP.

Se selecciona lo que se desee instalar, sólo es necesario instalar el servidor apache, php, y MySQL, en dado caso también se puede instalar phpMyAdmin para la administración desde un entorno web. [Figura 3.10]:

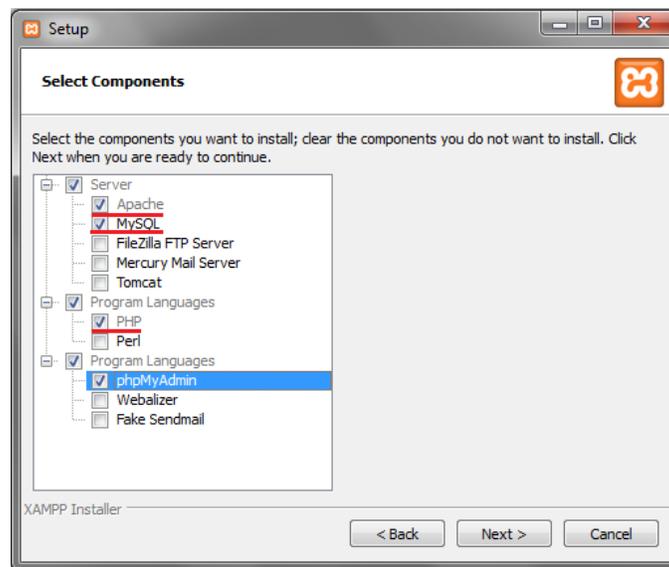


Figura 3.10: Herramientas a instalar en XAMPP.

Se selecciona la carpeta de instalación de XAMPP [Figura 3.11].

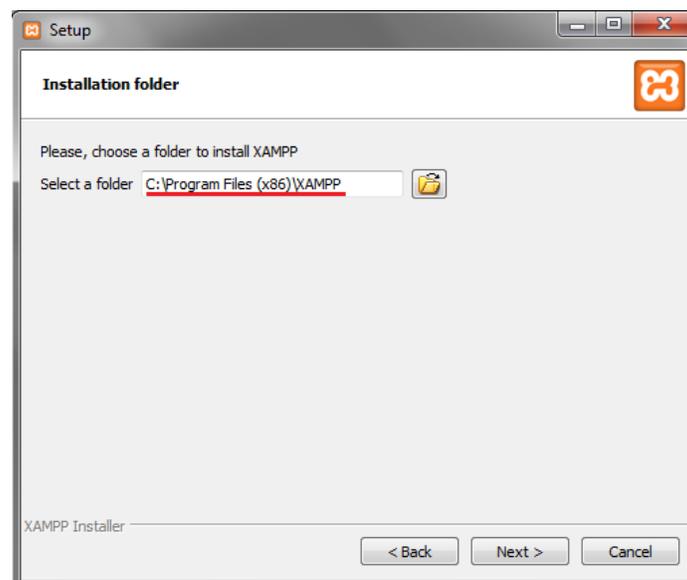


Figura 3.11: Carpeta de instalación de XAMPP.

Se ejecuta el panel de XAMPP y se inicia el servidor Apache con MySQL [Figura 3.12]



Figura 3.12: Panel de XAMPP.

Se verifica que funcione correctamente el servidor creando con un editor de texto para Windows un archivo de prueba en el directorio C:\Program Files (x86)\XAMPP\htdocs (en donde se instaló XAMPP) con extensión “.php” por ejemplo nombrándolo “phpinfo.php” y se escribe dentro del archivo una función en php.

```

1 <?php
2     phpinfo();
3 ?>
    
```

Por último se verifica que funcione correctamente abriendo un navegador web, y escribiendo en él la URL del servidor local, por lo general se instala con el nombre de “localhost”, pero en su defecto se puede utilizar la dirección IP del servicio local “127.0.0.1”, seguido del nombre del archivo php [Figura 3.13].

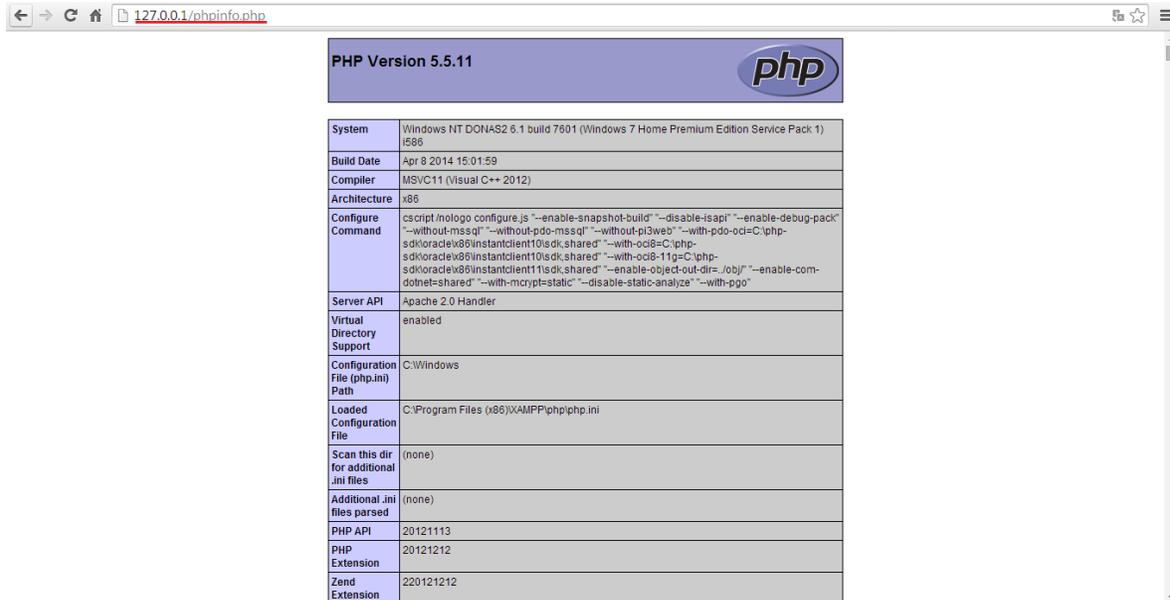


Figura 3.13: Prueba de XAMPP.

3.2.3 Creación de un servidor web en Arduino

Programación de Arduino remota desde Linux

Algo que impide el control dinámico del programa es la necesidad de estar forzosamente en contacto con la computadora y ésta que esté conectada al Arduino para hacer pruebas del programa, para estar monitoreando su comportamiento de manera continua, pero qué sucede si el servidor de base de datos y el SAD (Raspberry-Pi y Arduino) se encuentra en un entorno de difícil acceso, lo que se puede hacer es programar de manera remota al Arduino. Se programa remotamente mediante el protocolo SSH, la computadora que tiene conexión directa con el Arduino establece dicha conexión, en este caso es la Raspberry-Pi, una vez establecida la conexión primero se debe de verificar que efectivamente haya conectado un dispositivo Arduino en el sistema, para esto se utiliza el comando que muestra los dispositivos USB conectados a el sistema, esto es muy importante porque se debe de conocer qué tipo de Arduino se está manejando.

```
root@raspberrypi:~# lsusb
Bus 001 Device 002: ID 0424:9512 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 1a40:0201 Terminus Technology Inc. FE 2.1 7-port Hub
Bus 001 Device 005: ID 046d:c534 Logitech, Inc.
Bus 001 Device 006: ID 2341:0042 Arduino SA Mega 2560 R3 (CDC ACM)
```

Creación del servidor remotamente

Como se había mencionado anteriormente no se requiere forzosamente todos los elementos para que el sistema cumpla su objetivo (recolectar información del medio y mostrarla para su estudio posterior), así que se puede dejar de lado el servidor de base de datos como elemento externo y programar el Arduino para que funcione como servidor web, de esta manera sólo mostrará las variables capturadas pero no se podrá almacenar en una base de datos.

Para crear un servidor en Arduino se requiere forzosamente el *shield* de Arduino responsable de la conexión de a internet.

Se busca el ejemplo en el sistema operativo con su nombre: “*WebServer.ino*” (instalado previamente junto con las herramientas de desarrollo de Arduino):

```
root@raspberrypi:~# find / -name WebServer.ino
/usr/share/arduino/libraries/Ethernet/examples/WebServer/WebServer.ino
```

Se modifica el archivo conforme a las necesidades, se copia el archivo en el “*sketchbook*”, y se deberá crear una carpeta con el nombre del archivo.

```
root@raspberrypi:~# mkdir /home/pi/sketchbook/WebServer
root@raspberrypi:~# cp /usr/share/arduino/libraries/Ethernet/examples/WebServer/
WebServer.ino /home/pi/sketchbook/WebServer/
```

Se debe de crear un archivo llamado “*Makefile*” responsable de la compilación del archivo “.ino” [Figura 3.14]:

```
root@raspberrypi:/home/pi/sketchbook/WebServer# nano Makefile
```

```
GNU nano 2.2.6 File: Makefile Modified

#BOARD SETTINGS
BOARD_TAG = mega2560
ARDUINO_PORT = /dev/ttyACM*
ARDUINO_DIR = /usr/share/arduino/
ARDMK_DIR = /usr/local
AVR_TOOLS_DIR = /usr
ARDUINO_LIBS = Ethernet Ethernet/utility SPI

include ../Arduino.mk

[ Read 9 lines ]
^G Get Help ^C WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^M To Spell
```

Figura 3.14: Archivo Makefile.

Se edita el archivo de acuerdo a las necesidades, en este caso sólo se debe de cambiar la dirección *IP* asociada al dispositivo Arduino:

```
root@raspberrypi:/home/pi/sketchbook/WebServer# nano WebServer.ino
```

Se carga el archivo “.hex” al Arduino:

```
root@raspberrypi:/home/pi/sketchbook/WebServer# make upload
```

Como se utilizó un Arduino Mega, para cargar programas a este dispositivo se tiene que habilitar la opción de la memoria flash (-D), ya que si no se hace marcará un error y no se cargará el programa [Figura 3.15].

```
/usr/bin/avr-objcopy -O ihex -R .eeprom build-cli/WebServer.elf build-cli/WebServer.hex
/usr/bin/avrdude -q -V -p atmega2560 -C /etc/avrdude.conf -c stk500v2 -b 115200
-P /dev/ttyACM0 \
-U flash:w:build-cli/WebServer.hex:i

avrdude: AVR device initialized and ready to accept instructions
avrdude: Device signature = 0x1e9801
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: stk500v2_command(): command failed

avrdude done. Thank you.

make: *** [raw upload] Error 255
root@raspberrypi:/home/pi/sketchbook/WebServer#
```

Figura 3.15: Problema de carga de programa.

Por lo que bastara con cargarlo manualmente copiando la instrucción y agregando la opción -D a esta [Figura 3.16]:

```
root@raspberrypi:/home/pi/sketchbook/WebServer# /usr/bin/avrdude -q -V -D -p atm
ega2560 -C /etc/avrdude.conf -c stk500v2 -b 115200 -P /dev/ttyACM0 -U flash:w:bu
ild-cli/WebServer.hex:i

avrdude: AVR device initialized and ready to accept instructions
avrdude: Device signature = 0x1e9801
avrdude: reading input file "build-cli/WebServer.hex"
avrdude: writing flash (13192 bytes):
avrdude: 13192 bytes of flash written

avrdude: safemode: Fuses OK

avrdude done. Thank you.
```

Figura 3.16: Solución de carga de programa.

Se prueba el correcto funcionamiento del servidor abriendo en un navegador de internet y se localiza la IP que se asocia al Arduino [Figura 3.17].



Figura 3.17: Prueba de Arduino funcionando como servidor web.

A continuación se describen las instrucciones para compilar manualmente los programas:

Instrucciones de make.

<i>make:</i>	compila el programa con extensión “.ino”
<i>make upload:</i>	compila el programa y carga el programa al Arduino
<i>make clean:</i>	borra las dependencias creadas
<i>make depends:</i>	actualiza las dependencias
<i>make reset:</i>	aplica un reset al Arduino mediante le puerto serial
<i>make raw_upload:</i>	carga el programa sin aplicar un reset al Arduino
<i>make show_boards:</i>	muestra la lista de placas de Arduino definidas en boards.txt

3.3 INSTALACIÓN DE LA BASE DE DATOS

3.3.1 Creación de la base de datos en Linux

Para crear la base de datos se debió de instalar previamente todas las herramientas necesarias de MySQL, se establece la conexión a MySQL con el siguiente comando, y después de introducir el password abrirá MySQL [Figura 3.18].

```
root@raspberrypi:~# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.37-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figura 3.18: Acceso a MySQL.

Se muestran las bases de datos que existen en el servidor [Figura 3.19]

```
mysql> SHOW DATABASES;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema      |
+-----+
3 rows in set (0.00 sec)
```

Figura 3.19: Muestra de bases de datos.

Se crea la base de datos nombrándola preferentemente con el nombre de la aplicación, en este caso la aplicación será para recaudar datos provenientes de Arduino, se llamara “logs”.

```
mysql> CREATE DATABASE logs;
Query OK, 1 row affected (0.00 sec)
```

Se selecciona la base para poder crear tablas y campos:

```
mysql> USE logs;
Database changed
```

Se crea la tabla con sus respectivos campos [Figura 3.20]:

```
mysql> CREATE TABLE IF NOT EXISTS lectura (  
-> lectura_fecha varchar(10) COLLATE latin1_spanish_ci NOT NULL,  
-> lectura_hora varchar(10) COLLATE latin1_spanish_ci NOT NULL,  
-> lector_id smallint(6) NOT NULL,  
-> lectura_f1 float NOT NULL,  
-> lectura_f2 float NOT NULL,  
-> lectura_contador varchar(10) COLLATE latin1_spanish_ci NOT NULL  
-> ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;  
Query OK, 0 rows affected (2.68 sec)
```

Figura 3.20: Creación de tabla.

3.3.2 Creación de la base de datos en Windows

El entorno que se instaló (XAMPP) en Windows automáticamente instala el servicio de base de datos, es indispensable conocer el manejador que se va utilizar, así como el lenguaje con el que se puede comunicar con la base de datos, para crear bases, tablas, y campos dentro de las tablas, en este caso el Lenguaje de Definición de Datos y el Lenguaje de Manipulación de Datos son un subconjunto de SQL (por sus siglas en inglés *Structured Query Language*). Utilizando la herramienta “*phpMyAdmin*” se puede manejar la base de datos.

Creación de una base con phpMyAdmin:

Se tiene que abrir el panel de control de XAMPP, y se selecciona el botón de “*Admin*” de la parte de MySQL, esto abrirá en el navegador una página en donde se puede administrar las bases de datos [Figura 3.21].

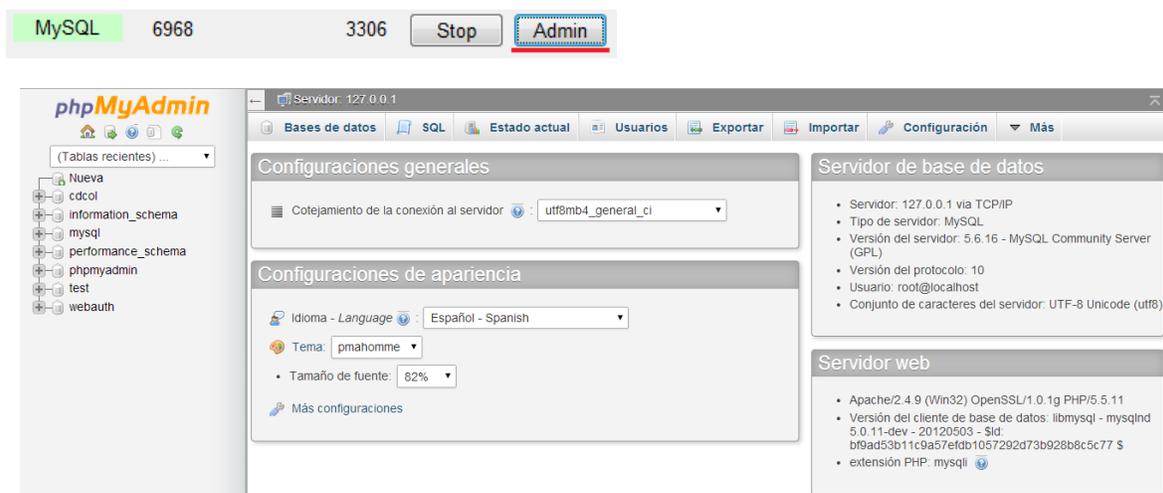


Figura 3.21: Acceso a phpMyAdmin.

Ya dentro del administrador para manipular la base se puede optar por ejecutar los comandos dentro de la pestaña “*SQL*”, o se puede hacer con el entorno de phpMyAdmin.

Por ejemplo utilizando el entorno de phpMyAdmin se crea una base de datos nueva haciendo clic en el botón “*Nueva*” ubicado en la parte izquierda de la herramienta, y se introduce el nombre que se le

dará a la base, esto es muy importante ya que se debe recordar este nombre para que después php pueda hacer la conexión a la base. Como la aplicación que se presenta en esta Tesis es de un SAD de propósito general con cualquier nombre será suficiente, en este caso se llamara “logs” [Figura 3.22].



Figura 3.22: Creación de base de datos con phpMyAdmin.

Para no dejar de lado el lenguaje de definición de datos, para crear la tabla y los campos de la misma se utilizará la pestaña SQL ubicada en la parte superior de la herramienta phpMyAdmin [Figura 3.23].

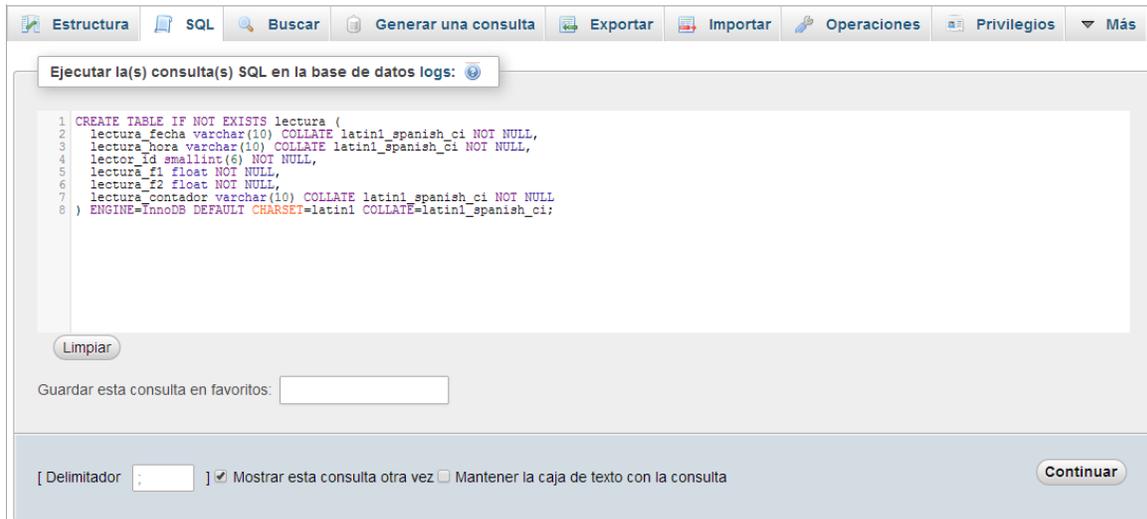


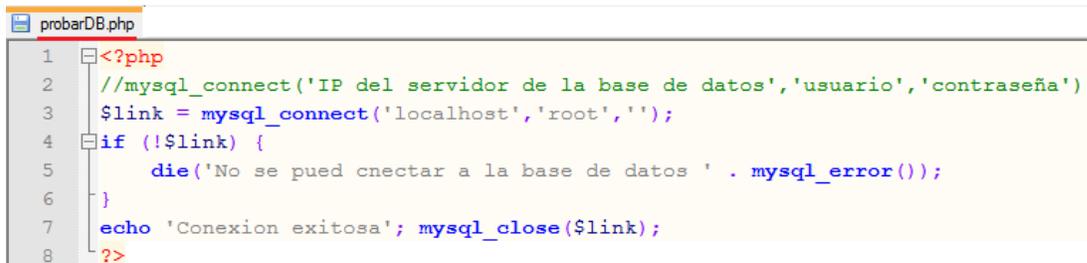
Figura 3.23: Creación de tabla desde phpMyAdmin.

Los nombres de los de los campos dentro de la base de datos serán utilizados cuando php se conecte a esta y guarde registros, por lo que será importante también recordar los nombres de estos.

3.4 CREACIÓN DEL PROYECTO

3.4.1 Creación de archivos del proyecto

Ya con el servidor Apache y MySql corriendo en el equipo se pudo verificar que desde php haya conexión a la base de datos. Para esto se creó una carpeta en donde se introduce todos los archivos “.php” para que los pueda encontrar el navegador, en este caso la carpeta tendrá el nombre de “Tesis” (la ruta en Linux es: /var/www/Tesis mientras que en Windows es C:\Program Files (x86)\XAMPP\htdocs\Tesis), enseguida se creó un programa de prueba para verificar el correcto funcionamiento, con el nombre de “probarDB.php” [Figura 3.24].



```
1 <?php
2 //mysql_connect('IP del servidor de la base de datos','usuario','contraseña');
3 $link = mysql_connect('localhost','root','');
4 if (!$link) {
5     die('No se pued cnectar a la base de datos ' . mysql_error());
6 }
7 echo 'Conexion exitosa'; mysql_close($link);
8 ?>
```

Figura 3.24: Archivo de prueba de conexión con la base de datos.

Para corroborar el funcionamiento correcto el navegador deberá mostrar un mensaje como el siguiente [Figura 3.25] cuando se ingresa la ruta en el navegador, en este caso la ruta es <http://127.0.0.1/Tesis/probarDB.php>.

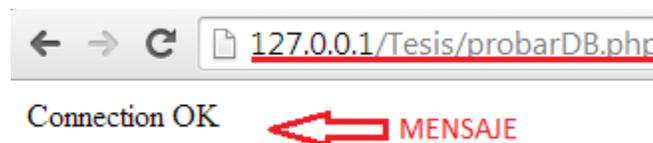


Figura 3.25: Funcionamiento correcto de archivo de prueba de base de datos.

Se procede a crear un archivo dedicado para la conexión y otro dedicado para el cierre de la base de datos [Figura 3.26].

```

conectaDB.php
1  <?php
2  /*Datos de conexion a la base de datos*/
3  $usuario="root";
4  $servidor="localhost";
5  $password="";
6  $bd="logs";
7  /*Inicia conexion a bd mysql */
8  if ($conexion=mysql_connect($servidor,$usuario,$password))
9  {
10 //print "Conexion exitosa: ".$conexion." <br> ";
11 /*Selecciona la base de datos sobre la que trabajar */
12 if (mysql_select_db($bd,$conexion))
13 {
14 //print "Conexion a la base de datos exitosa <br>".$bd;
15 }
16 else
17 {
18 //print "No se logro; realizar la conexion a la base de datos, sin privilegios para acceder a ".$bd;
19 }
20 }
21 else
22 {
23 //print "No se logro; realizar la conexion a la base de datos".mysql_get_error($conexion);
24 }
25 ?>

```

Figura 3.26: Archivo para conectar a la base de datos.

Se tuvo que utilizar el nombre exacto de la base en este caso fue “logs”, también en caso de algún error se puede editar el archivo quitado algunas líneas comentadas .

Se creó el archivo que cierra la base de datos [Figura 3.27]

```

cerrarDB.php
1  <?php
2  mysql_close($conexion);
3  ?>

```

Figura 3.27: Archivo para cerrar la conexión con la base de datos.

Estos dos archivos siempre deberán incluirse cuando se desee hacer una conexión a la base de datos.

Ya con la conexión establecida se verificó el funcionamiento con un archivo de prueba. [Figura 3.28]

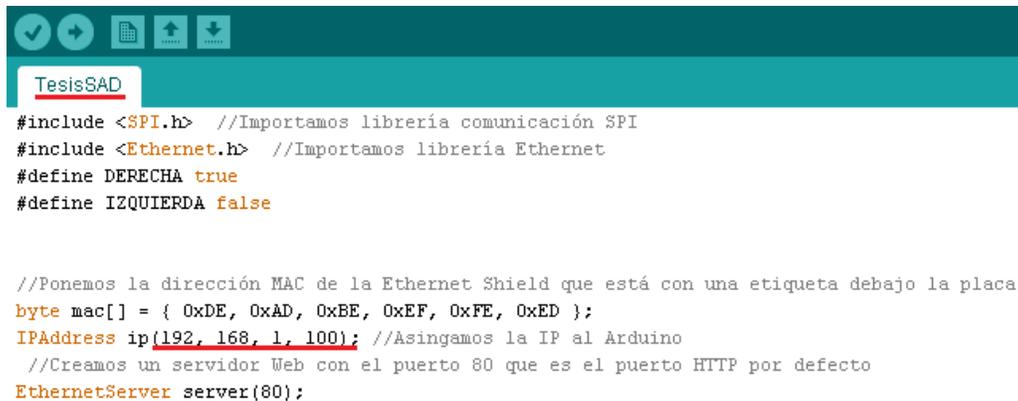


```
1 <?php
2 include "conectaDB.php";
3 $value="";
4 $lectura_fecha=date('d-m-Y');
5 $lectura_hora =date('H:i:s');
6 $lector_id =1;
7 $lectura_f1=1.0;
8 $lectura_f2 =1.0;
9 $lectura_contador=1;
10 $query="INSERT INTO lectura (
11 `lector_id`,
12 `lectura_contador`,
13 `lectura_f1`,
14 `lectura_f2`,
15 `lectura_fecha`,
16 `lectura_hora`
17 )VALUES (
18 ".$lector_id.",
19 ".$lectura_contador.",
20 ".$lectura_f1.",
21 ".$lectura_f2.",
22 ".$lectura_fecha.",
23 ".$lectura_hora.")";
24 //print $query;
25 if ($value = mysql_query($query, $conexion))
26 {
27 //print "Se inserto la medicaci&oacute;n: ".$query."<br>";
28 }else
29 {
30 //print "NO se logro insertar la medicion:".$query."<br>";
31 }
32 include "cerrarDB.php";
33 ?>
```

Figura 3.28: Archivo de prueba para manipular la base de datos.

Los campos deben de tener el mismo nombre para que pueda insertar los datos adecuadamente y se debe tener ya creados los archivos “*cerrarDB.php*” y “*conectaDB.php*”, responsables de la conexión, de igual manera si se desea verificar el funcionamiento se puede quitar el comentario de algunas líneas.

Después se creó el archivo “.ino” que se cargó al Arduino y que será responsable de capturar los datos, controlar un motor a pasos y también encender un foco [Figura 3.29].



```

TesisSAD
#include <SPI.h> //Importamos librería comunicación SPI
#include <Ethernet.h> //Importamos librería Ethernet
#define DERECHA true
#define IZQUIERDA false

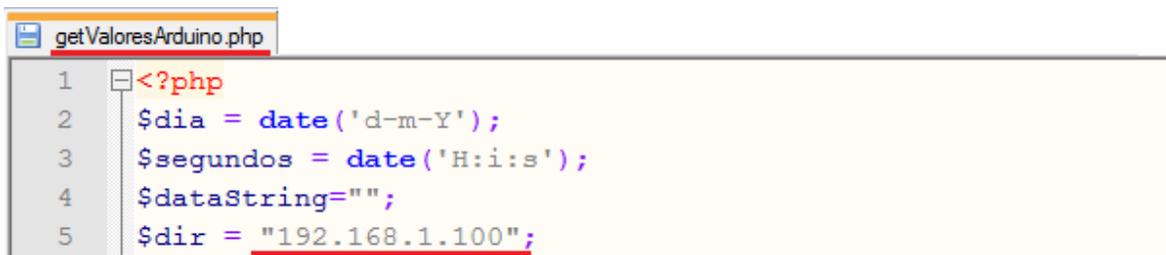
//Ponemos la dirección MAC de la Ethernet Shield que está con una etiqueta debajo la placa
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 100); //Asingamos la IP al Arduino
//Creamos un servidor Web con el puerto 80 que es el puerto HTTP por defecto
EthernetServer server(80);

```

Figura 3.29: Archivo que se cargara al Arduino.

Es importante recordar la dirección *IP* que se asigna al Arduino

Se creó después un archivo php responsable de obtener los datos provenientes del Arduino vía Ethernet [Figura 3.30].



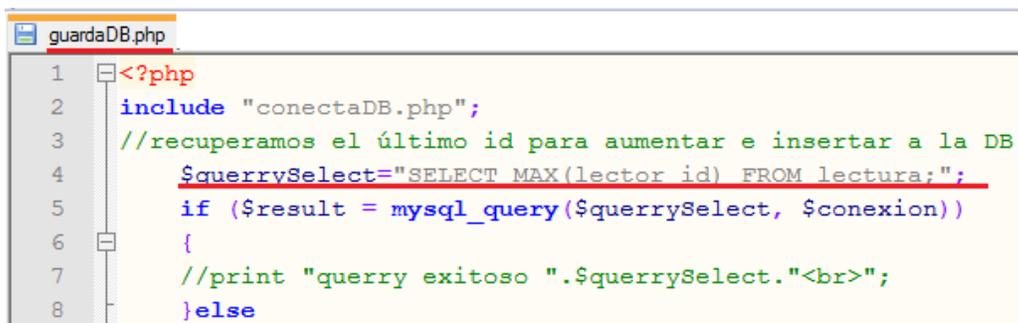
```

getValoresArduino.php
1 <?php
2 $dia = date('d-m-Y');
3 $segundos = date('H:i:s');
4 $dataString="";
5 $dir = "192.168.1.100";

```

Figura 3.30: Archivo para recuperar valores del Arduino.

Se creó el archivo para obtener los datos del Arduino y guardarlos en la base de datos [Figura 3.31]



```

guardaDB.php
1 <?php
2 include "conectaDB.php";
3 //recuperamos el último id para aumentar e insertar a la DB
4 $querySelect="SELECT MAX(lector id) FROM lectura;";
5 if ($result = mysql_query($querySelect, $conexion))
6 {
7 //print "query exitoso ".$querySelect."<br>";
8 }else

```

Figura 3.31: Archivo que almacena en la base de datos.

Para evitar que existiera un id repetido se obtiene el último registro guardado en la base de datos y se aumenta una unidad a la variable, luego se almacenan los registros en la base de datos.

Ya que se obtuvieron los registros en la base de datos se tenían que mostrar [Figura 3.32], y también se tuvo que generar un archivo que pudiera manejarse como una tabla en una hoja de cálculo.

```

muestraDB.php
1 <html lang = "es">
2 <head>
3   <meta charset="utf-8"/>
4   <meta name="autor" content="JANR"/>
5 </head>
6
7 <body background="./img/fondoTesis.jpg">
8 <?php
9   include "conectaDB.php";
10  $var=0;
11  $querySelectAll="SELECT lectura fecha, lectura hora, lectura f1,".
12  " lectura f2, lectura contador, lector id FROM lectura ORDER BY lector id DESC;";
13  $result = mysql_query($querySelectAll, $conexion);

```

Figura 3.32: Archivo que muestra el contenido de la base de datos.

Para poder visualizar el contenido de la base de datos se accede al archivo mediante la URL del navegador de internet [Figura 3.33].



FECHA	HORA	LECTURA 1	LECTURA 2	LECTURA 3
05-08-2014	03:47:00	259	248	250
05-08-2014	03:46:59	260	247	251
05-08-2014	03:46:58	264	247	255
05-08-2014	03:46:29	270	252	259
05-08-2014	03:45:43	270	252	263
05-08-2014	03:44:57	271	253	260
30-07-2014	07:50:55	246	238	241
30-07-2014	07:50:54	250	244	248
30-07-2014	07:50:53	254	243	245
30-07-2014	07:50:52	253	246	249

[GENERAR REPORTE](#)

Figura 3.33: Visualizar contenido de la base de datos.

Se obtuvieron los datos y en otro archivo se generó el documento “.csv”, esta extensión es un archivo separado por comas, que es una manera sencilla para generar reportes en formato de hoja de cálculo.

Se generó el archivo responsable de crear el reporte (Archivo con extensión “.csv”) de acuerdo a los resultados obtenidos de la consulta a la Base de Datos [Figura 3.34].

```

fwrite($archivo, $variables);
while ($row = mysql_fetch_row($result)){
    fwrite($archivo, $row[0].",".$row[1].",".$row[2].",".$row[3].",".$row[4]."\r");
    $var++;
    if($var==10)
        break;
}
fclose($archivo);

```

Figura 3.34: Archivo que genera el archivo ".csv".

Por último se creó un archivo responsable de mostrar las variables que se pueden monitorizar en el Arduino y una interfaz para controlar los puertos digitales del mismo, se ocuparon formularios para enviar la información por el método “get”, y que el Arduino responda de manera adecuada a la petición que se solicita, por ejemplo mover un motor, encender un relevador que encenderá un foco. Este archivo fue la página principal de la aplicación web por lo que fue nombrada como “index.php” [Figura 3.35].

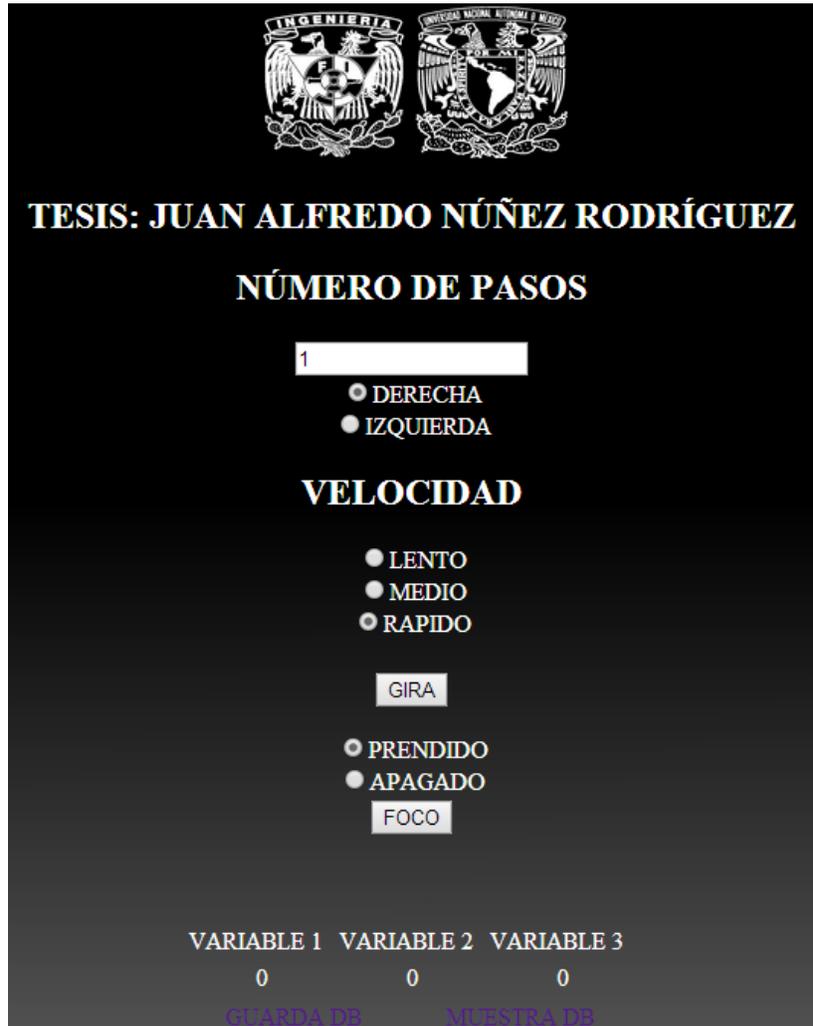


Figura 3.35: Muestra de la aplicación vista desde el navegador.

3.4.2 Conexión

Ya con el servidor funcionando se procedió a conectar la tarjeta de adquisición de datos desarrollada para probar el sistema, con sensores de temperatura, luz y salinidad; incluido también en la tarjeta los circuitos para el encendido de un foco y el movimiento de un motor a pasos.

En la figura 3.37 se muestra el esquema de conexiones mientras que a continuación se muestra la simbología utilizada en el esquema de conexiones [Figura 3.36]

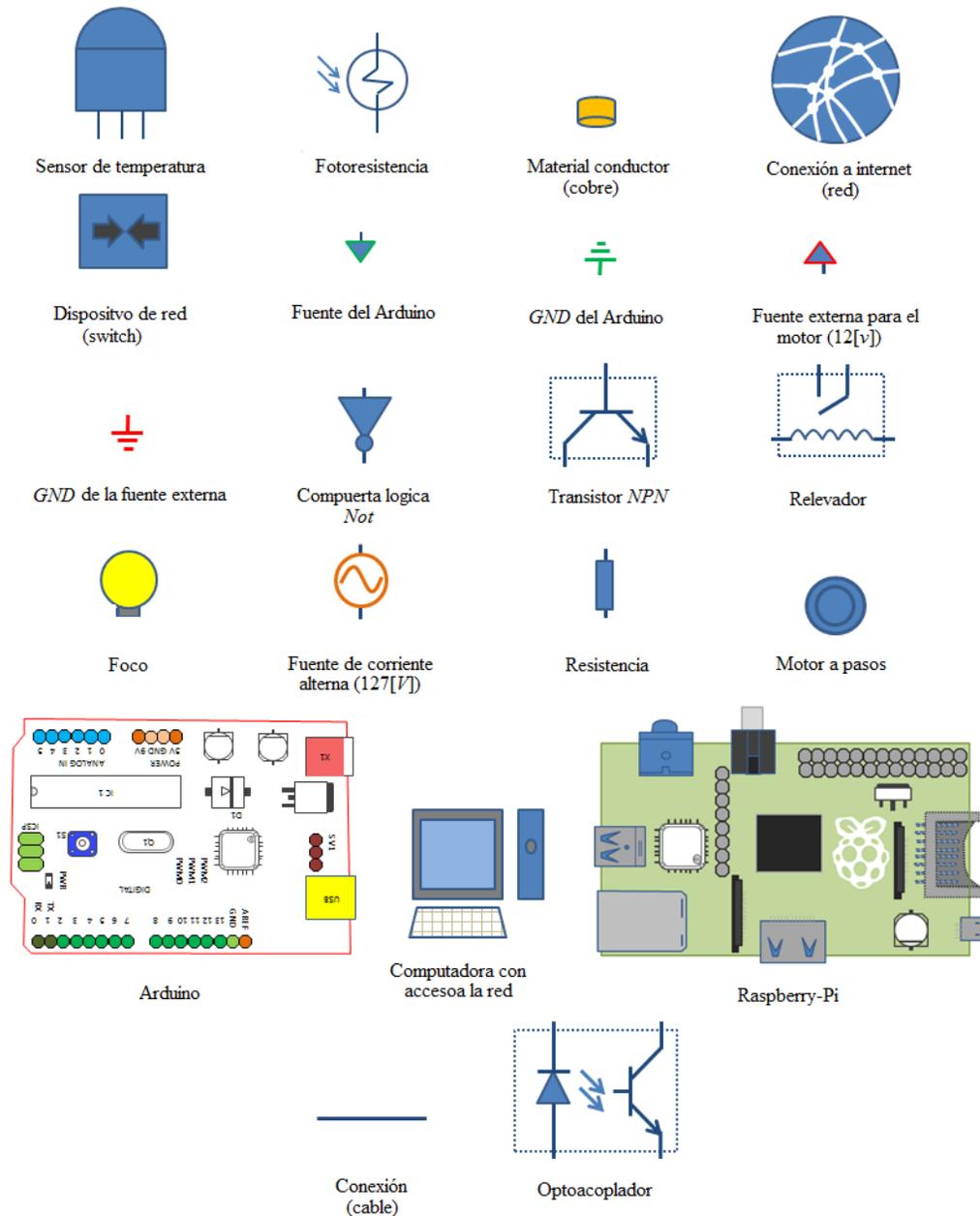


Figura 3.36: Simbología del esquema de conexiones.

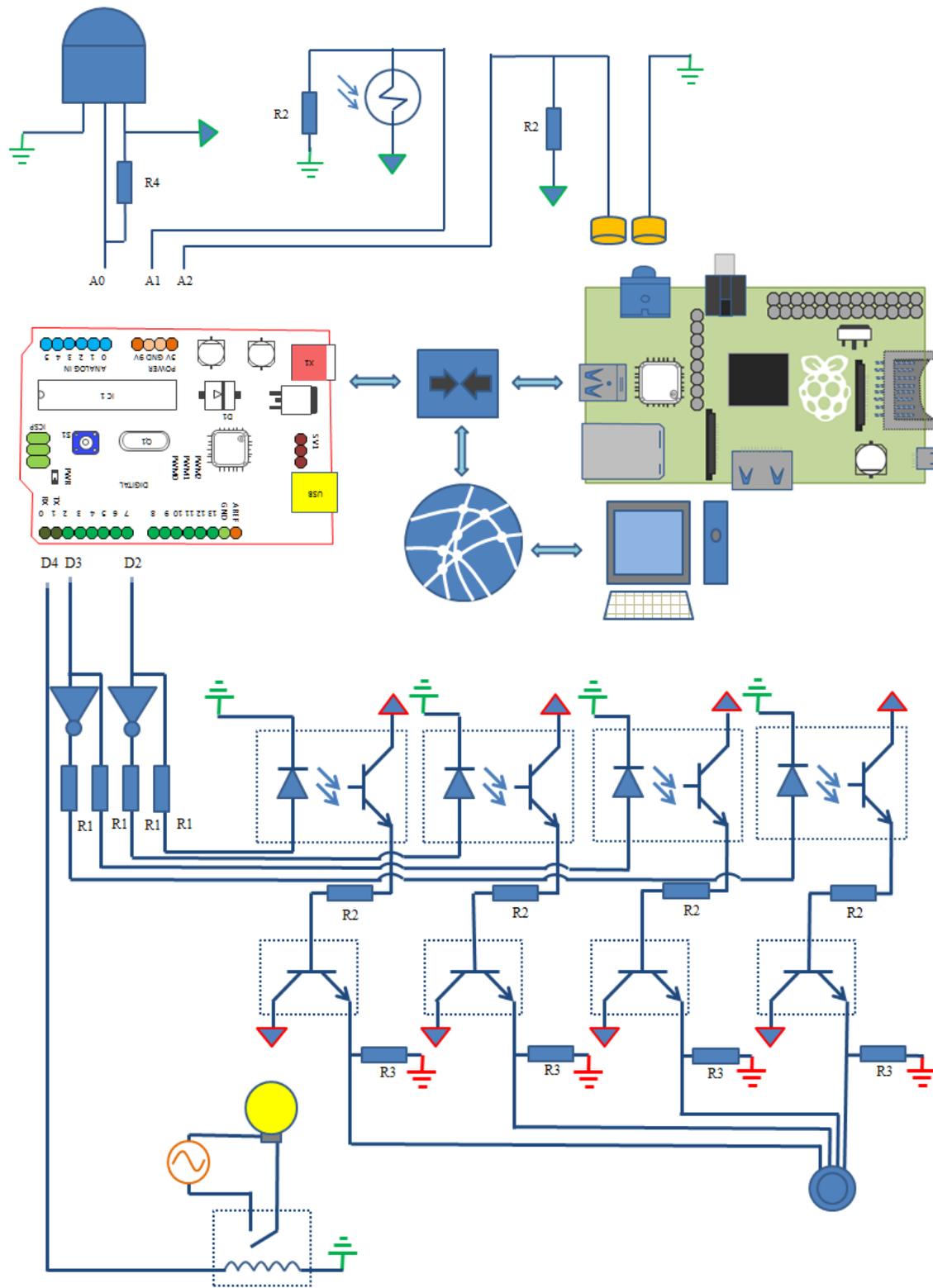


Figura 3.37: Esquema de conexiones.

Las letras incluidas en el esquema de conexiones indican:

- D2, D3: Conexión con el puerto digital de Arduino (Control del motor).
- D4: Conexión con el puerto digital de Arduino (Control del foco).
- A0: Conexión con el puerto analógico de Arduino (Sensor de temperatura).
- A1: Conexión con el puerto analógico de Arduino (Sensor de luz).
- A2: Conexión con el puerto analógico de Arduino (Sensor salinidad).
- R1: resistencia de $100\ \Omega$ a $\frac{1}{4}$ de Watt.
- R2: resistencia de $100\ \text{K}\Omega$ a $\frac{1}{4}$ de Watt.
- R3: resistencia de $100\ \Omega$ a $1\ \text{Watt}$.
- R4: resistencia de $1.7\ \text{K}\Omega$ a $\frac{1}{4}$ de Watt.

En la siguiente figura se muestran algunas imágenes del sistema conectado [Figura 3.38]:

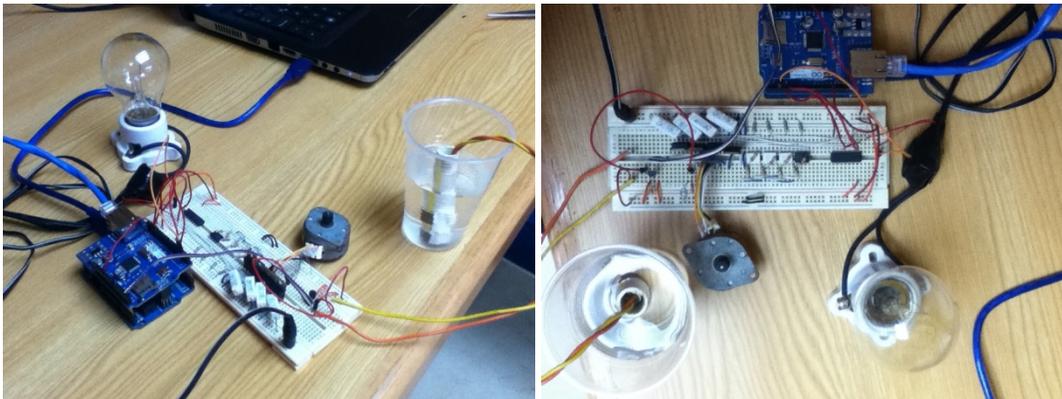


Figura 3.38: Sistema conectado.

CAPÍTULO 4: ANÁLISIS DE RESULTADOS

4.1 APLICACIÓN DEL SISTEMA EN UN ENTORNO REAL

Para este capítulo se utilizó el sistema de adquisición de datos desarrollado en el Capítulo 3 “*Desarrollo del proyecto*” en un entorno real, mostrando los resultados obtenidos en el mismo.

En el laboratorio de microcomputadoras de DGTIC se ha estado trabajando y desarrollando diferentes tipos de sistemas para las necesidades que se presentan, entre ellos esta: el SAD adecuado a una boya marina para recolectar datos de Ph, humedad, luminosidad, temperatura del aire y agua, salinidad del agua; el SAD para recolectar datos de bailarines tales como posición del bailarín, presión en los pies del bailarín y velocidad de movimiento; un SAD para captura de datos de voltajes, corrientes, temperatura, luminosidad y existencia de gas.

Con la gama de posibilidades de desarrollo y aplicación del SAD en un entorno real, se decidió orientar al SAD para la recolección de datos provenientes de voltajes, corrientes, temperaturas, luminosidad y gas.

Esta necesidad surgió a raíz de un incendio suscitado en un mercado a causa de un posible problema en la instalación eléctrica y presencia de gas inflamable. Por lo cual se decidió diseñar un sistema con un conjunto de cajas, que fuera capaz de monitorear voltajes, corrientes, temperaturas, luminosidad y gas.

El sistema se desarrolló con base con el siguiente esquema [Figura 4.1]:

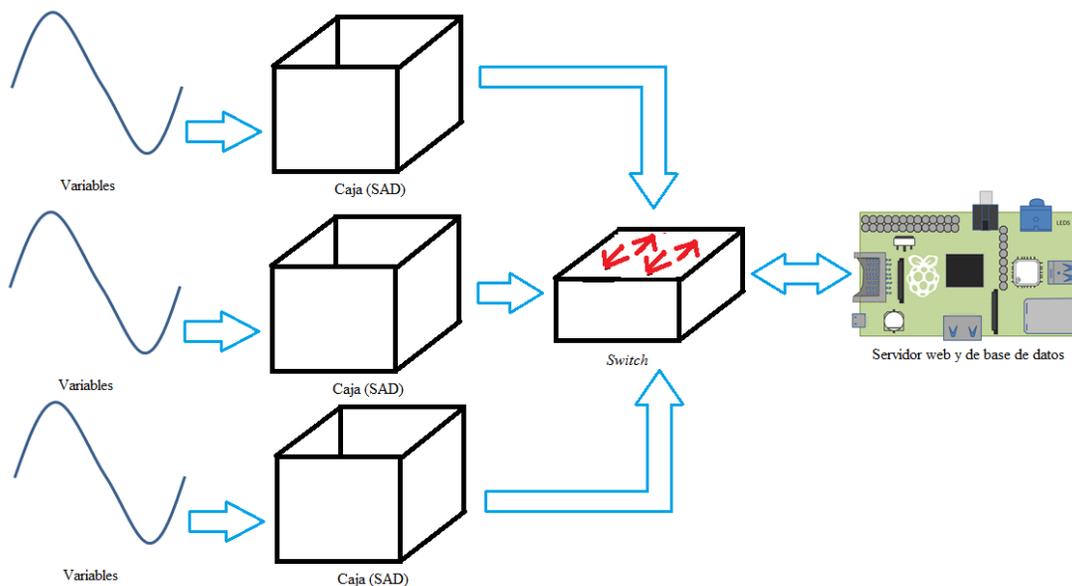


Figura 4.1: Esquema del sistema.

En la figura 4.2 se pueden observar las cajas en donde se colocaron los SAD.



Figura 4.2: Cajas en donde se colocaran los SAD.

En la figura 4.3 se puede notar el material para la construcción de las tarjetas que se encargan de traducir el valor de la variable física medida a un valor que lo pueda interpretar el Arduino y un *switch* utilizado para la conexión de las cajas vía Ethernet, así como el sensor de gas especializado para la medición de gas [Figura 4.4].

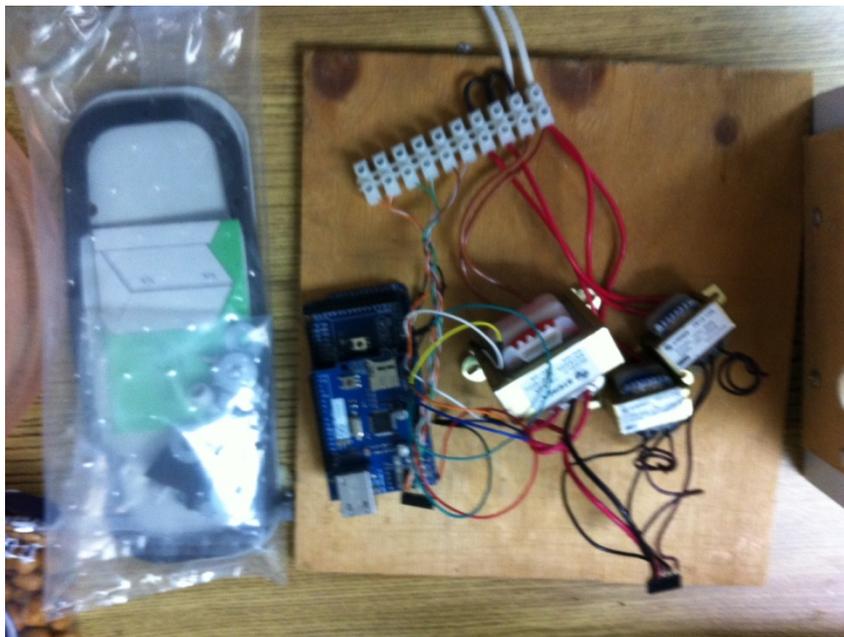


Figura 4.3: Material para el diseño de la tarjeta.



Figura 4.4: Sensor de gas.

En la figura 4.5 se puede notar como se fueron desarrollando las tarjetas de adquisición de datos.

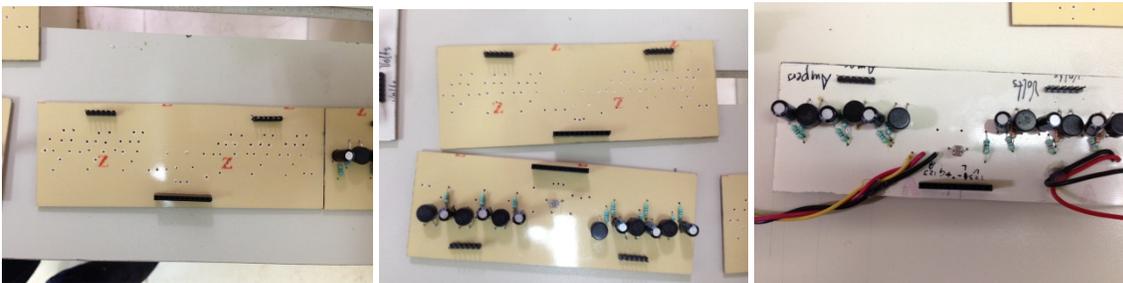


Figura 4.5: Tarjetas diseñadas para la captura de datos.

Ya terminadas las pruebas de todas las tarjetas y el monitoreo en un entorno local se procedió a la instalación de cada una de las cajas [Figura 4.6].

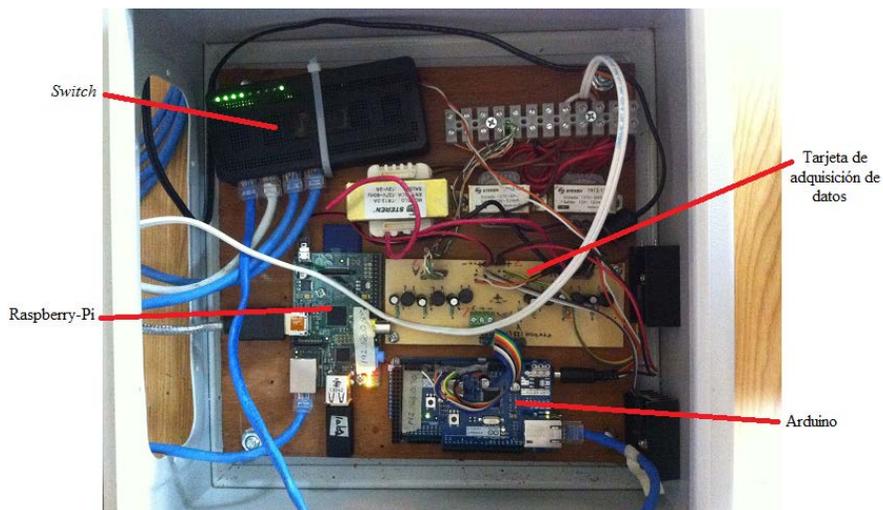


Figura 4.6: Caja con Arduino, Raspberry-Pi, tarjeta de adquisición y switch.

Luego se procedió a copiar el diseño de la primera caja en las demás (estas cajas sólo contendrán un Arduino y la tarjeta de adquisición). [Figura 4.7]

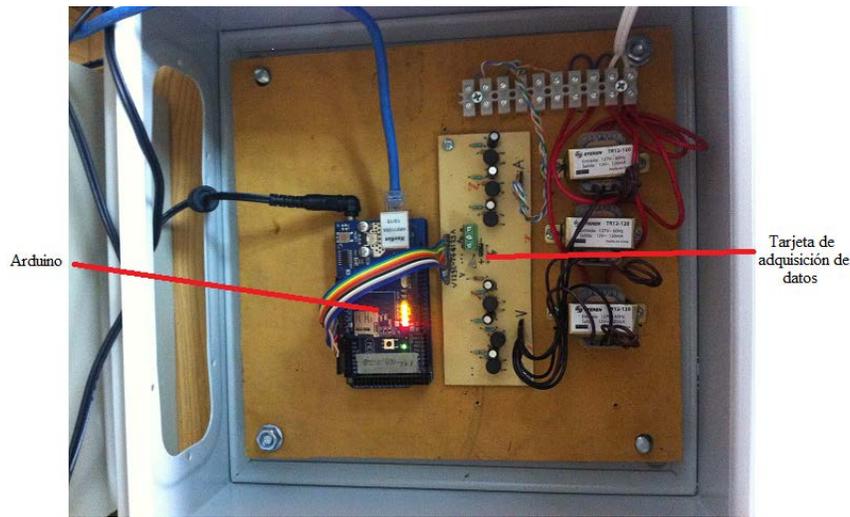


Figura 4.7: Caja con Arduino y tarjeta.

Las cajas están especializadas para el análisis de variables eléctricas tales como transientes, corrientes y voltajes, incluso se medirán variables como luminosidad (para verificar que la caja se encuentra cerrada) y gas (para verificar que no exista alguna fuga y que pueda provocar una accidente ocasionada por algún corto circuito).

Debido a que el sistema desarrollado en el Capítulo 3 “*Desarrollo del proyecto*” fue un sistema de propósito general se realizaron las siguientes modificaciones.

Primero se consideró que el número de SAD y variables físicas en los mismos aumentó, por lo que se pudo haber modificado la base de datos o cómo mejor opción se creó otra.

Para esto se tuvo que establecer la conexión a MySQL con el siguiente comando y enseguida se escribió el password asignado al usuario [Figura 4.8]:

```
root@raspberrypi:~# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 5.5.38-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Figura 4.8: Creación de base de datos para el proyecto.

Se verificó las bases de datos existentes con el siguiente comando [Figura 4.9]:

```
mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| logs              |
| mysql             |
| performance_schema |
| phpmyadmin        |
+-----+
5 rows in set (0.00 sec)
```

Figura 4.9: Muestra de las bases de datos.

Se creó una nueva base de datos con un nombre nuevo (en este caso se le llamó “*cajas*”) [Figura 4.10]:

```
mysql> CREATE DATABASE cajas;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database          |
+-----+
| information_schema |
| cajas             |
| logs              |
| mysql             |
| performance_schema |
| phpmyadmin        |
+-----+
6 rows in set (0.01 sec)
```

Figura 4.10: Creación de la nueva base de datos.

Se cambió a la base de datos “*cajas*” y se creó la tabla llamada “*lectura*” [Figura 4.11]:

```
mysql> use cajas
Database changed
mysql> CREATE TABLE IF NOT EXISTS `lectura` (
  -> `lectura_id` bigint(20) NOT NULL,
  -> `lectura_fecha` varchar(10) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL,
  -> `lectura_hora` varchar(10) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL,
  -> `lectura_amp1` int(11) NOT NULL,
  -> `lectura_amp2` int(11) NOT NULL,
  -> `lectura_amp3` int(11) NOT NULL,
  -> `lectura_volt1` int(11) NOT NULL,
  -> `lectura_volt2` int(11) NOT NULL,
  -> `lectura_volt3` int(11) NOT NULL,
  -> `lectura_luz` int(11) NOT NULL,
  -> `lectura_gas` int(11) NOT NULL,
  -> `lectura_transiente` int(11) NOT NULL,
  -> `lectura_caja` varchar(20) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL,
  -> PRIMARY KEY (`lectura_id`)
  -> ) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='tabla para datos del servidor de las cajas';
Query OK, 0 rows affected (0.07 sec)
```

Figura 4.11: Creación de la tabla.

Se pudo verificar que la tabla se ha creó correctamente con las columnas especificadas, comprobado esta acción con el siguiente comando [Figura 4.12]:

```
mysql> desc lectura;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| lectura_id     | bigint(20)    | NO   | PRI | NULL    |      |
| lectura_fecha  | varchar(10)   | NO   |     | NULL    |      |
| lectura_hora   | varchar(10)   | NO   |     | NULL    |      |
| lectura_amp1   | int(11)       | NO   |     | NULL    |      |
| lectura_amp2   | int(11)       | NO   |     | NULL    |      |
| lectura_amp3   | int(11)       | NO   |     | NULL    |      |
| lectura_volt1  | int(11)       | NO   |     | NULL    |      |
| lectura_volt2  | int(11)       | NO   |     | NULL    |      |
| lectura_volt3  | int(11)       | NO   |     | NULL    |      |
| lectura_luz    | int(11)       | NO   |     | NULL    |      |
| lectura_gas    | int(11)       | NO   |     | NULL    |      |
| lectura_transiente | int(11)      | NO   |     | NULL    |      |
| lectura_caja   | varchar(20)   | NO   |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.01 sec)
```

Figura 4.12: Descripción de la tabla.

Se modificaron los archivos necesarios para el funcionamiento correcto y se agregó un nuevo archivo en donde se incluyeron las direcciones IP de las cajas llamado “dirCajas.php” [Figura 4.13].

```
dirCajas.php
1 <?php
2 /*
3 En este archivo se agrgan las direcciones ip de los arduinos responsables de
4 obtener los datos (SAD)
5 */
6 $cajas = array(
7 "192.168.0.26",
8 "192.168.0.27",
9 "192.168.0.28",
10 "192.168.0.29",
11 "192.168.0.30",
12 "192.168.0.31"
13 );
14 ?>
```

Figura 4.13: Archivo que contiene las direcciones IP de las cajas.

Se modificó el archivo “conectaDB.php” cambiando el nombre de la base a donde se hará la conexión [Figura 4.14].

```
9 // $bd="logs";
10 $bd="cajas";
```

Figura 4.14: Selección de base de datos en archivo conectaDB.php.

Se modificó el archivo “*getValoresArduino.php*” para que el código sirva como una función, sea llamado cuando sea necesario, y que sólo se haga responsable de la captura de los datos provenientes del Arduino [Figura 4.15].

```

1  <?php
2  /*
3  En este archivo se obtienen los valores de los SAD (Arduino
4  mediante peticiones al SAD que funciona como servidor
5  */
6  /******
7  function getValoresArduino($dir){
8      $fp = fsockopen($dir, 80, $errno, $errstr, 0.5);
9      if (!$fp) { //header ("Location: ./index2.php");
10     }
11     else
12     {
13         $out = "\nGET / HTTP/1.1\r\n";
14         $out .= "Host: ".$dir."\r\n";
15         $out .= "Connection: Close\r\n\r\n";
16
17         //echo $out."<br>";
18         fwrite($fp, $out);
19         $dataString="";
20         while (!feof($fp)) {
21             try{
22                 $dataString = $dataString.fgets($fp, 1000)."\r\n";
23             }catch (Exception $e){
24                 echo "exception";
25             }
26         }
27         fclose($fp);
28     }
29     return $dataString;
30 }
31 /******
32
33 ?>
    
```

Figura 4.15: Archivo *getValoresAduino.php* modificado.

Se creó un nuevo archivo encargado de utilizar la función “*getValoresArduino (\$dir);*” para obtener todos los valores que arrojan los Arduino (identificados en el archivo “*dirCajas.php*” por la dirección ip), mediante un “*foreach*” se recorre las direcciones IP y se guardan los valores en la base de datos. [Figura 4.16].

```

1  <?php
2  include "getValoresArduino.php";
3  include "dirCajas.php";
4  include "conectaDB.php";
5
6  foreach($cajas as $dir){
    
```

Figura 4.16: Archivo *recupera.php*.

Se modificó el “*index.php*” de tal manera que primero se haga la consulta a los Arduino y después se consulte los registros que se guardaron en la base de datos. Consultando la hora del servidor, mediante la función *date()* de php se guarda un reporte del día anterior, de este modo se tiene el control del momento en que se generará el reporte. [Figura 4.17].

```

index.php
13 <?php
14 /*
15 Este archivo muestra lo que está almacenado en la base de datos
16 */
17
18 $dia=0; //variable para geerar reporte del dia actual
19 $hora_actual=floatval(date('H'));
20 $min_actual=floatval(date('i'));
21 if ($hora_actual==0 && ($min_actual==0)){
22     $dia = 1; // variable para generar reporte del dia anterior
23     include "guardaArc.php";
24 }
25 include "recupera.php";

```

Figura 4.17: Modificación al archivo "index.php".

Se modificó el archivo "guardaArc.php" para que pudiera generar el archivo general (un reporte de toda la base de datos) y el reporte del día anterior, al momento de realizar la consulta se agrega la restricción correspondiente [Figura 4.18].

```

guardaArc.php
1 <?php
2 /*
3 En este archivo se generan reportes con extensión .csv que pueden ser interpretados por las hojas de cálculo
4 */
5 include "conectaDB.php";
6 // $ruta_archivo = ".\Archivos\\"; //para windows
7 $ruta_archivo = "./Archivos/"; //para linux
8 $var_fech=date('m-d')." ";
9 $var_fech_hoy=explode("-", $var_fech);
10 $var_fech_ayer="(floatval($var_fech_hoy[1])-$dia)."-".date('m-Y');
11 $var_hor=date("H.i.s")." ";
12 $archivo = fopen($ruta_archivo."REPORTE_".$var_fech."_".$var_hor.".csv", 'w');
13
14 if($sql_todo){
15     $extra="";
16 }else{
17     $extra="where lectura fecha = '".$var_fech_ayer.'";
18     $sql_todo=false;
19 }

```

```
lectura_id FROM lectura ".$extra." ORDER BY
```

Figura 4.18: Modificación del archivo "guardaArc.php".

Se creó un archivo que responda al realizar el reporte del día anterior o el reporte total [Figura 4.19].

```

guardaArcTotal.php
1 <?php
2 $sql_todo=true;
3 $dia=0;
4 include "guardaArc.php";
5 $sql_todo=false;
6 header ("Location: ./index.php");
7
8 ?>

```

Figura 4.19: Archivo "guardaArcTotal.php".

4.2 RESULTADOS

En la figura 4.20 se puede observar todas las cajas funcionando adecuadamente, capturando datos y enviándolos a través del cable Ethernet.



Figura 4.20: Cajas (SAD).

Con el diseño dinámico del sistema basta con modificar el archivo “*dirCajas.php*” para agregar o eliminar los SAD al sistema, y este agregará automáticamente los resultados a la página web, gracias a que la página se actualiza cada cierto tiempo se pueden monitorear las variables en tiempo real [Figura 4.21].

Cabe destacar que en el momento que se hacen las consultas, se hacen peticiones “*http*” a los Arduino (que funcionan como servidores “*http*” para enviar los datos que tienen almacenados), se debe resaltar que depende de las capacidades del servidor (Raspberry-pi), para abrir y cerrarlos *sockets* que establecen la conexión con Arduino, ya que estos tardan alrededor de 20 segundos en cerrarse en la Raspberry-Pi. Por lo que la consulta a los Arduino será aproximadamente cada minuto, sin embargo en el programa de los Arduino se almacenara la información de cada segundo aproximadamente.

CAJA	IP	FECHA	HORA	AMP_1	AMP_2	AMP_3	VOLT_1	VOLT_2	VOLT_3	LUZ	GAS	TEMPERATURA
192.168.0.26	05-10-2014	16:41:21	0	0	0	508	511	509	22	3	0	
192.168.0.27	05-10-2014	16:41:24	0	0	0	511	512	511	5	4	0	
192.168.0.28	05-10-2014	16:41:26	0	0	0	513	514	521	0	2	0	
192.168.0.29	05-10-2014	16:41:29	0	0	0	512	509	509	2	0	0	
192.168.0.30	05-10-2014	16:41:31	0	0	0	790	800	919	318	1	0	
192.168.0.31	05-10-2014	16:41:39	0	0	0	511	519	514	18	3	0	

GENERAR REPORTE TOTAL

Figura 4.21: Sistema en funcionamiento.

Se pueden generar archivos con extensión “.csv” [Figura 4.22] con datos provenientes de la base de datos que son interpretados por hojas de cálculo para un posible reporte accediendo a la ruta de los archivos [Figura 4.23]:

	A	B	C	D	E	F	G	H	I	J	K	L
1	CAJA_IP	FECHA	HORA	AMPERAJE 1	AMPERAJE 2	AMPERAJE 3	VOLTAJE 1	VOLTAJE 2	VOLTAJE 3	LUZ	GAS	TRANSIENTES
2	192.168.254.133	31/08/2014	02:43:08	234	214	122	244	182	65	75	198	239
3	192.168.254.131	31/08/2014	02:43:08	65	219	68	123	245	176	43	190	100
4	192.168.254.133	31/08/2014	02:42:55	234	214	122	244	182	65	75	198	239
5	192.168.254.131	31/08/2014	02:42:55	65	219	68	123	245	176	43	190	100
6	192.168.254.133	31/08/2014	02:36:32	11	22	33	44	55	66	77	88	99

Figura 4.22: Muestra de hoja de cálculo generada a partir de la consulta a la base de datos.

The screenshot shows a web browser window with the address bar containing the URL 192.168.0.45/SERV_CAJAS/Archivos/. The main content area displays the title "Index of /SERV_CAJAS/Archivos" and a table of files with columns for Name, Last modified, Size, and Description. The files listed are:

Name	Last modified	Size	Description
Parent Directory	-	-	-
REPORTE_08-31_02.07.34_.csv	12-Sep-2014 22:35	410	
REPORTE_08-31_04.47.38_.csv	12-Sep-2014 22:35	769	
REPORTE_09-29_19.30.59_.csv	29-Sep-2014 18:31	96K	
REPORTE_09-30_16.25.20_.csv	30-Sep-2014 15:25	103K	
REPORTE_10-02_17.30.04_.csv	02-Oct-2014 16:30	118K	
REPORTE_10-05_16.38.36_.csv	05-Oct-2014 15:38	129K	

At the bottom of the page, it says "Apache/2.2.22 (Debian) Server at 192.168.0.45 Port 80".

Figura 4.23: Ubicación de los archivos generados.

CAPÍTULO 5: CONCLUSIONES Y TRABAJO A FUTURO

Utilizando tecnologías que han surgido en los últimos años se logró desarrollar un sistema de adquisición de datos que captura información proveniente de fenómenos físicos y es capaz de almacenarla en una base de datos; se desarrolló una aplicación web para monitorear los datos almacenados en la base y mediante consultas a esta también se pueden crear archivos “.csv”, este tipo de archivos se pueden interpretar como hojas de cálculo, que pueden servir para la elaboración de reportes para un estudio posterior.

El funcionamiento del sistema consiste en un conjunto de SAD recolectando información del medio, cada SAD es independiente de los otros y es identificado mediante una dirección *IP* (para este trabajo cada SAD es representado por Arduino, el *shield* de Ethernet y la tarjeta de adquisición de datos), de esta manera la Raspberry-Pi funciona como servidor web y de base de datos para todo el conjunto de SAD.

En este trabajo se describió el proceso de desarrollo e implementación de un SAD, con el objetivo de que personas interesadas en el tema puedan desarrollar un sistema semejante al aquí diseñado, dándole un uso específico a estas tecnologías (Arduino y Raspberry-Pi). También se incluyó un desarrollo semejante en el sistema operativo Windows, con esto se resalta que el sistema es adaptable a cualquier entorno que funcione como servidor web y de base de datos.

Debido a que la Raspberry-Pi y Arduino son dispositivos relativamente nuevos en el mercado existe información limitada en cuestiones de estructura, por lo cual en este trabajo se explicaron algunas características de los mismos, incluyendo algunas aplicaciones que se les pueden dar, para que se tenga una idea del alcance que pueden tener, mostrando que la adquisición de cualquiera de estos dispositivos resulta útil y rentable para diferentes usos y aplicaciones.

En la actualidad los SAD se utilizan para la investigación de variables físicas, el principal distribuidor de este tipo de sistemas es National Instruments con su producto LABVIEW, desafortunadamente este producto tiene un costo elevado, con este trabajo de esta tesis se pretende que las personas que no dispongan de los recursos necesarios para adquirir un SAD puedan desarrollar el suyo a un bajo costo y con un funcionamiento práctico.

Es importante destacar que las tecnologías Arduino y Raspberry-Pi tienen mucho futuro ya que son tecnologías relativamente nuevas que no buscan competir con las existentes (como dispositivos móviles, laptops y tabletas), sino que buscan dar una aplicación específica y a un bajo costo, sobre todo un uso para a educación y como en este caso para la investigación.

La integración y uso de las diferentes tecnologías existentes y las nuevas tecnologías que se están desarrollando es una muestra de que las ingenierías (sobre todo en área de la computación), son campos en donde se requiere el conocimiento amplio de diferentes ciencias, técnicas y estudios, que deben de trabajar en conjunto para un objetivo concreto, en este caso la unión resultante de la Raspberry-Pi, Arduino, controladores de *hardware* (para el correcto funcionamiento del motor a pasos), bases de datos, servidores web, y programación fue un sistema capaz de analizar variables capturadas mediante Arduino y almacenarlas en una base de datos alojada en la Raspberry-Pi, teniendo acceso a la aplicación mediante el protocolo internet y teniendo el control de los puertos digitales del Arduino mediante el acceso a la

red, esto da la ventaja de poder tener el conocimiento y control al sistema desde cualquier parte del mundo con acceso a internet.

El proyecto que se diseñó es un sistema de adquisición y registro de datos de propósito general, estos sistemas son muy útiles en la industria y en algunos casos son utilizados por investigadores para la recaudación y estudio de datos de cualquier tipo de variable analógica provenientes del mundo exterior, tales como la corriente, voltaje, presión atmosférica, temperatura, etc., este sistema es planeado para ser desarrollado con elementos económicos con un bajo consumo de recursos, tal como el consumo de corriente eléctrica. Por lo que es una herramienta útil en la industria y en universidades.

Constantemente en estas tecnologías están surgiendo avances y características que las hacen únicas, por ejemplo los nuevos *shields* que surgen para un uso específico, que incluso por su fabricación estandarizada se pueden crear *shields* desarrollados por nosotros y probar aplicaciones directamente. En lo particular se cree que el desarrollo de este tipo de sistemas impulsa mucho el desarrollo de la tecnología en México ya que no precisamente se tiene que comprar un sistema a la medida, sino se puede utilizar este sistema para el registro de datos de cualquier tipo, claro que sólo con los sensores y transductores adecuados.

De acuerdo al avance en estas tecnologías y específicamente hablando del SAD no sólo puede responder a las peticiones por el entorno web, ya que existen infinidad de complementos que nos brindan los desarrolladores de estos dispositivos, sobre todo Arduino para un trabajo posterior, como por ejemplo, se puede utilizar en el Arduino un complemento que envíe los datos utilizando la red telefónica, debido a la existencia de un *shield* que manipula una SIM e interactúa con el usuario vía celular, entonces en lugar de tener que conectarse a internet para interactuar con el sistema, se puede hacer enviando un mensaje o incluso una llamada al SAD.

El uso de estas tecnologías no sólo apoya a la investigación sino también al desarrollo, ya que abre la puerta a las empresas que se encargan del control de entornos, para ser más específico en el campo de la domótica, por ejemplo se pueden utilizar estos dispositivos para el desarrollo de un sistema casero encargado de monitorear si existe alguna fuga de gas en el hogar, medir la temperatura de algún horno en la industria, etc., para poder tomar medidas sobre el entorno en caso de que sobrepase los rangos permitidos, como cerrar la válvula de gas, encender o apagar las luces de casa, o directamente cortar la corriente eléctrica para apagar algo crítico.

Un aspecto importante a resaltar es el objetivo principal de la Raspberry-Pi, ya que de acuerdo a sus desarrolladores ésta fue creada para la educación. El modo de aprendizaje para el ambiente Linux es en general la búsqueda de documentación adecuada para instalar aplicaciones, servicios, etc., esto hace que el alumno interesado aprenda el uso de las computadoras cuando no existía el ambiente gráfico, es decir cuando se hacía todo desde la línea de comandos, regresando al método de enseñanza-aprendizaje del funcionamiento de las primeras computadoras, ya que hoy en día con la expansión del mercado tecnológico se nos venden aplicaciones finales, es decir, ya no es necesario preocuparse por la configuración de la aplicación en los dispositivos ya que esto se hace automáticamente, por una parte se le facilita la vida al ser humano pero por otra parte se crea una dependencia total del experto en la aplicación (la compañía que vende o desarrolla la aplicación), la ventaja que tiene Linux es que es de código abierto, es decir que se puede tener acceso al código de la aplicación para modificarla, mejorarla o para hacer diferentes pruebas sin depender de terceros que cobren por un servicio de mantenimiento. Este tipo de microcomputadoras se pueden aprovechar en la carrera para una mejor preparación al ambiente laboral.

APÉNDICE A: ARCHIVOS DEL SAD GENERAL

A.1 TesisSAD.ino

```

#include <SPI.h> //Importamos librería comunicación SPI
#include <Ethernet.h> //Importamos librería Ethernet
#define DERECHA true
#define IZQUIERDA false

//Ponemos la dirección MAC de la Ethernet Shield que está con una etiqueta debajo la placa
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 100); //Asingamos la IP al Arduino
//Creamos un servidor Web con el puerto 80 que es el puerto HTTP por defecto
EthernetServer server(80);
boolean PASO[4][2]={{ true,true},
                    { false,true},
                    { false,false},
                    { true,false}};

boolean visual=false;
int numPasos=0;
int VELOCIDAD=10;
int NUM_PASO=0;

int foco=5; //Pin del led
int pinMotor1 = 3;
int pinMotor2 = 2;

String estado="OFF"; //Estado del Led inicialmente "OFF"

void setup()
{
  Serial.begin(9600);

  pinMode(pinMotor1, OUTPUT);
  pinMode(pinMotor2, OUTPUT);
  pinMode(foco, OUTPUT);
  // Inicializamos la comunicación Ethernet y el servidor
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Direccion ip");
  Serial.println(Ethernet.localIP());
  Serial.print(".....");
}

void loop()
{
  EthernetClient client = server.available(); //Creamos un cliente Web
  //Cuando detecte un cliente a través de una petición HTTP
  if (client) {
    Serial.println("Nuevo cliente");
    boolean currentLineIsBlank = true; //Una petición HTTP acaba con una línea en blanco
    String cadena=""; //Creamos una cadena de caracteres vacía
    String strPasos="";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();//Leemos la petición HTTP carácter por carácter
        Serial.write(c);//Visualizamos la petición HTTP por el Monitor Serial
        cadena.concat(c);//Unimos el String 'cadena' con la petición HTTP (c). De esta manera convertimos la petición HTTP
        a un String

        //Ya que hemos convertido la petición HTTP a una cadena de caracteres, ahora podremos buscar partes del texto.

```

```

int posicion=cadena.indexOf("MOTOR="); //Guardamos la posición de la instancia "LED=" a la variable 'posicion'
int pasos_inti=cadena.indexOf("PASOS=");
int foco_int=cadena.indexOf("FOCO=");
strPasos=cadena.substring(pasos_inti+6,posicion);

int psVelo=cadena.indexOf("VELOCIDAD=");
int num_de_pasos=strPasos.toInt();

if(cadena.substring(psVelo)=="VELOCIDAD=LENTO"){
    VELOCIDAD=80;
}else if(cadena.substring(psVelo)=="VELOCIDAD=MEDIO"){
    VELOCIDAD=40;
}else if(cadena.substring(psVelo)=="VELOCIDAD=RAPIDO"){
    VELOCIDAD=10;
}

if(cadena.substring(posicion)=="MOTOR=DERECHA")
{
    giraMotor(num_de_pasos, DERECHA);
    visual=true;
}
if(cadena.substring(posicion)=="MOTOR=IZQUIERDA")
{
    giraMotor(num_de_pasos, IZQUIERDA);
    visual=true;
}
if(cadena.substring(foco_int)=="FOCO=PRENDIDO")
{
    digitalWrite(foco, true);
    visual=true;
}
if(cadena.substring(foco_int)=="FOCO=APAGADO")
{
    digitalWrite(foco, false);
    visual=true;
}

//Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha acabado y el servidor Web está listo para
enviar una respuesta
if (c == '\n' && currentLineIsBlank) {

    // Enviamos al cliente una respuesta HTTP
    /*client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close"); // the connection will be closed after completion of the response
    client.println("Refresh: 5"); // refresh the page automatically every 5 sec
    client.println();
    */
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    client.println("<head>");
    if(visual){
        client.println("<META HTTP-EQUIV=\\"Refresh\\" CONTENT=\\"0; URL=http://192.168.1.72/Tesis\>");//esta el
la ip del servidor Web PHP
    }
    client.println("</head>");
    for (int analogChannel = 0; analogChannel < 3; analogChannel++) {
        int sensorReading = analogRead(analogChannel);
        client.print(",");
    }
}

```

```

    client.print(sensorReading);

    }
    client.println("<br />");
    client.println("</html>");
    break;
    }
    if (c == '\n') {
        currentLineIsBlank = true;
    }
    else if (c != '\r') {
        currentLineIsBlank = false;
    }
    }
    }
    //Dar tiempo al navegador para recibir los datos
    delay(1);
    client.stop();// Cierra la conexión
    }
}

void giraMotorOnce (boolean pin1,boolean pin2){
    digitalWrite(pinMotor1, pin1);
    delay(VELOCIDAD);
    digitalWrite(pinMotor2, pin2);
    delay(VELOCIDAD);
}

void giraMotor(int paso, boolean direccion){
    for(int i = paso; i--; i>=0){
        actualizaPaso(NUM_PASO, direccion);
        giraMotorOnce(PASO[NUM_PASO][0],PASO[NUM_PASO][1]);
    }
}

void actualizaPaso(short pas, boolean dir){
    if (dir){
        NUM_PASO++;
        if(NUM_PASO>3)
            NUM_PASO=0;
    }else if(!dir){
        NUM_PASO--;
        if(NUM_PASO<0)
            NUM_PASO=3;
    }
}

```

A.2 conectaBD.php

```

<?php
/*Datos de conexion a la base de datos*/
$usuario="root";
$servidor="localhost";
$password="";
$dbd="logs";
/*Inicia conexion a bd mysql */
if ($conexion=mysql_connect($servidor,$usuario,$password))
{
//print "Conexi&oacute;n exitosa: ".$conexion." <br> ";
/*Selecciona la base de datos sobre la que trabajar */

```

```

if (mysql_select_db($bd,$conexion))
{
//print "Conexion a la base de datos exitosa <br>".$bd;
}
else
{
//print "No se logro; realizar la conexion a la base de datos, sin privilegios para acceder a ".$bd;
}
}
else
{
//print "No se logro; realizar la conexion a la base de datos".mysql_get_error($conexion);
}
?>

```

A.3 index.php

```

<html lang = "es">
<head>
    <meta charset="utf-8"/>
    <meta name="autor" content="JANR"/>
    <title>TESIS</title>
    <script>
        function enviaMotorArduino()
        {
            if (document.datosMotor.PASOS.value.length==0){
                alert("Tiene que escribir un numero valido")
                document.datosMotor.PASOS.focus();
            }else{
                document.datosMotor.submit();
            }
        }

        function enviaFocoArduino()
        {
            document.datosFoco.submit();
        }
    </script>
</head>

<body background="/img/fondoTesis.jpg">
<?php
include "getValoresArduino.php"
?>
<center>
<a href="http://www.ingenieria.unam.mx/" target="_self"></a>
<a href="http://www.unam.mx/" target="_self"></a>
<form method="get" name="datosMotor" action="http://192.168.1.100">
    <h2><font color="#FFFFFF">TESIS: JUAN ALFREDO N&Uacute;&Ntilde;EZ RODR&Iacute;GUEZ</font></h2>
    <h2><font color="#FFFFFF">N&Uacute;MERO DE PASOS</font></h2> <input type="number" name="PASOS"
id="PASOS" min="1" value="1"><br>
    <input type="radio" name="MOTOR" value="DERECHA" checked><font color="#FFFFFF">DERECHA</font><br>
    <input type="radio" name="MOTOR" value="IZQUIERDA"><font color="#FFFFFF">IZQUIERDA</font><br>
    <h2><font color="#FFFFFF">VELOCIDAD</font></h2>
    <input type="radio" name="VELOCIDAD" value="LENTO" checked><font color="#FFFFFF">LENTO</font><br>
    <input type="radio" name="VELOCIDAD" value="MEDIO"><font color="#FFFFFF">MEDIO</font><br>
    <input type="radio" name="VELOCIDAD" value="RAPIDO" checked><font
color="#FFFFFF">RAPIDO</font><br><br>
    <input type="button" value="GIRA" onclick="enviaMotorArduino()">
</form>
<form method="get" name="datosFoco" action="http://192.168.1.100">

```

```

<input type="radio" name="FOCO" value="PRENDIDO" checked><font
color="#FFFFFF">PRENDIDO</font><br>
<input type="radio" name="FOCO" value="APAGADO"><font color="#FFFFFF">APAGADO</font><br>
<input type="button" value="FOCO" onclick="enviaFocoArduino()">
</form>
<br/><br/>
<table style="width:300px" >
<tr>
<td><font color="#FFFFFF">VARIABLE 1</font></td>
<td><font color="#FFFFFF">VARIABLE 2</font></td>
<td><font color="#FFFFFF">VARIABLE 3</font></td>
</tr>
<tr>
<td><center><font color="#FFFFFF"><?php echo $val1; ?></font></center></td>
<td><center><font color="#FFFFFF"><?php echo $val2; ?></font></center></td>
<td><center><font color="#FFFFFF"><?php echo $val3; ?></font></center></td>
</tr>
</table>
<table style="width:300px">
<tr>
<td>
<form>
<a href="/guardaDB.php" target="_self"><center>GUARDA
DB</center></a>
</td>
<td>
<a href="/muestraDB.php" target="_self"><center>MUESTRA
DB</center></a>
</td>
</tr>
</table>
</center>
</body>
</html>

```

A.4 muestraDB.php

```

<html lang = "es">
<head>
<meta charset="utf-8"/>
<meta name="autor" content="JANR"/>
</head>
<body background="/img/fondoTesis.jpg">
<?php
include "conectaDB.php";
$var=0;
$querySelectAll="SELECT lectura_fecha, lectura_hora, lectura_f1,
" lectura_f2, lectura_contador, lector_id FROM lectura ORDER BY lector_id DESC;";
$result = mysql_query($querySelectAll, $conexion);
echo "<center>";
echo "<table border = '1' > \n";
echo "<tr><td><font color='\#FFFFFF'\>FECHA</font></td>
<td><font color='\#FFFFFF'\>HORA</font></td>
<td><font color='\#FFFFFF'\>LECTURA 1</font></td>
<td><font color='\#FFFFFF'\>LECTURA 2</font></td>
<td><font color='\#FFFFFF'\>LECTURA 3</font></td>
</tr> \n";

while ($row = mysql_fetch_row($result)){
echo "<tr><td><font color='\#FFFFFF'\>$row[0]</font></td>
<td><font color='\#FFFFFF'\>$row[1]</font></td>

```

```

                <td><font color="#FFFFFF">$row[2]</font></td>
                <td><font color="#FFFFFF">$row[3]</font></td>
                <td><font color="#FFFFFF">$row[4]</font></td>
            </tr> \n";
            $var++;
            if($var==10)
                break;
        }
        echo "</table> \n";

        echo "<a href='./guardaArc.php' target='_self'>GENERAR
REPORTE</a>";
        echo "</center>";
        include "cerrarDB.php";
?>
</body>
</html>

```

A.5 pruebaQuery.php

```

<?php
include "conectaDB.php";
$value="";
$selectura_fecha=date('d-m-Y');
$selectura_hora =date('H:i:s');
$lector_id =1;
$selectura_f1=1.0;
$selectura_f2 =1.0;
$selectura_contador=1;
$query="INSERT INTO lectura(
`lector_id`,
`lectura_contador`,
`lectura_f1`,
`lectura_f2`,
`lectura_fecha`,
`lectura_hora`
)VALUES(
"`.`lector_id`.",
"`.`lectura_contador`.",
"`.`lectura_f1`.",
"`.`lectura_f2`.",
"`.`lectura_fecha`.",
"`.`lectura_hora`.");";
//print $query;
if ($value = mysql_query($query, $conexion))
{
//print "Se inserto la medicion: ". $query."<br>";
}else
{
//print "NO se logro insertar la medicion:". $query."<br>";
}
include "cerrarDB.php";
?>

```

A.6 probarDB.php

```

<?php
//mysql_connect('IP del servidor de la base de datos','usuario','contraseña');
$link = mysql_connect('localhost','root','');
if (!$link) {
    die('No se puede conectar a la base de datos ' . mysql_error());
}
echo 'Conexion exitosa'; mysql_close($link);
?>

```

A.7 ardaDB.php

```

<?php
include "conectaDB.php";
//recuperamos el último id para aumentar e insertar a la DB
$querySelect="SELECT MAX(lector_id) FROM lectura;";
if ($result = mysql_query($querySelect, $conexion))
{
    //print "query exitoso ".$querySelect."<br>";
}else
{
    //print "query no exitoso ".$querySelect."<br>";
}

if (!$result) {
    /*
        echo "DB Error, no se puede conectar\n";
        echo 'MySQL Error: ' . mysql_error();
        exit;
    */
}

$fila = mysql_fetch_row($result);
//si no hay elementos en la tabla
if($fila[0]=="")
    $fila[0]=0;
////////////////////////////////////
//recuperamos las demás variables
include "getValoresArduino.php";

$value="";
$lectura_fecha=date('d-m-Y');
$lectura_hora =date('H:i:s');
$lector_id =++$fila[0];
$lectura_f1=$val1;
$lectura_f2=$val2;
$lectura_contador=$val3;

$query="INSERT INTO lectura(
`lector_id`,
`lectura_contador`,
`lectura_f1`,
`lectura_f2`,
`lectura_fecha`,
`lectura_hora`
)VALUES(
"'.$lector_id.'",
"'.$lectura_contador.'",
"'.$lectura_f1.'",
"'.$lectura_f2.'",
"'.$lectura_fecha.'",
"'.$lectura_hora.'";

//print $query;

if ($value = mysql_query($query, $conexion))
{
    //print "Se inserto la medici&oacute;n: ".$query."<br>";
}else
{
    //print "NO se logro insertar la medicion: ".$query."<br>";
}

include "cerrarDB.php";
header ("Location: ../#");
?>

```

A.8 guardaArc.php

```
<?php
include "conectaDB.php";
// $ruta_archivo = ".\Archivos\\";//para windows
$ruta_archivo = "./Archivos/";//para linux
$var_fech=date('m-d')."_" ;
$var_hor=date("H.i.s")."_";
$archivo = fopen($ruta_archivo."REPORTE_".$var_fech."_".$var_hor.".csv",'w');
if($archivo == false){
    die("No se ha podido crear el archivo.");
}else{
    $var=0;
    $querySelectAll="SELECT lectura_fecha, lectura_hora, lectura_f1, lectura_f2, lectura_contador, lector_id FROM
lectura ORDER BY lector_id DESC;";
    $result = mysql_query($querySelectAll, $conexion);
    $variables="FECHA, HORA, LECTURA 1, LECTURA 2, LECTURA 3\r";
    fwrite($archivo, $variables);
    while ($row = mysql_fetch_row($result)){
        fwrite($archivo, $row[0].",".$row[1].",".$row[2].",".$row[3].",".$row[4]."\r");
        $var++;
        if($var==10)
            break;
    }
    fclose($archivo);
}
include "cerrarDB.php";
header ("Location: ./muestraDB.php");
?>
```

A.9 cerrarDB.php

```
<?php
mysql_close($conexion);
?>
```

A.10 getValoresArduino.php

```
<?php
$dia = date('d-m-Y');
$segundos = date('H:i:s');
$dataString="";
$dir = "192.168.1.100";
$fp = fsockopen($dir, 80, $errno, $errstr, 60);
if (!$fp) { echo "Serrstr ($errno)<br />\n";}
else
{
    $out = "GET / HTTP/1.1\r\n";
    $out .= "Host: ".$dir."\r\n";
    $out .= "Connection: Close\r\n\r\n";
    //echo $out."<br>";
    fwrite($fp, $out);
    while (!feof($fp)) {
        $dataString = $dataString.fgets($fp, 100)."\r\n";
    }
}
```

APÉNDICE B: ARCHIVOS DEL SAD DEL ENTORNO REAL

B.1 servidorParaCaja.ino

```

/*
  Servidor_trifasico_para_HTML_Caja5

  Ethernet shield attached to pins 10, 11, 12, 13

  Connect the photoresistor one leg to pin 0, and pin to +5V
  Connect a resistor (around 10k is a good value, higher
  values gives higher readings) from pin 0 to GND.
      PhotoR  10K
+5  o---^V^---^V^---o GND
      |
      Pin 0
-----
  Sensor de Gas MQ2 Conectar S a A0
*/

#include <SPI.h>
#include <Ethernet.h>
#include <NewPing.h>
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 RTC;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// Enter MAC address & IP address for your controller below.
IPAddress ip(192,168,0,27); //<<<< ENTER YOUR IP ADDRESS HERE!!!
// Initialize the Ethernet server library
EthernetServer server(80); // (port 80 is default for HTTP):
int hora = 0, minuto = 0, segundo = 0;
int iteracion = 1;
int num_cad = 1;
String cadena=String("");
String cadena_1=String("");
String cadena_2=String("");
String cadena_3=String("");
String cadena_aux=String("");

// para sensor de Gas
// -----

void setup()
{
  pinMode(7, OUTPUT);
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  //Si quitamos el comentario de la linea siguiente, se ajusta la hora y la fecha con la del ordenador
  //RTC.adjust(DateTime(__DATE__, __TIME__));
  Serial.println("Servidor_web_para_manitorear_circuitos_trifasicos: ....\nip: 192.168.0.27");

  Ethernet.begin(mac, ip); // start the Ethernet connection and the server:
  server.begin();
  //delay(1000);

```

```

}
void loop() {

// Real Time Clock
double amps1=0, amps2=0, amps3=0, volts1=0, volts2=0, volts3=0, dato, gas, luz;
for (int i=0; i<400; i++) {
  dato = analogRead(0)-1;
  if (dato > amps1) amps1 = dato;
}
//Serial.print(", amps: "); Serial.print(amps1*0.28);

for (int i=0; i<400; i++) {
  dato = analogRead(1)-1;
  if (dato > amps2) amps2 = dato;
}
//Serial.print(", "); Serial.print(amps2*.28);

for (int i=0; i<400; i++) {
  dato = analogRead(2)-1;
  if (dato > amps3) amps3 = dato;
}
//Serial.print(", "); Serial.print(amps3*.28);

for (int i=0; i<400; i++) {
  dato = analogRead(3);
  if (dato > volts1) volts1 = dato;
} // Serial.print(", volts,"); Serial.print(volts1*1.96);

for (int i=0; i<400; i++) {
  dato = analogRead(4);
  if (dato > volts2) volts2 = dato;
} // Serial.print(", "); Serial.print(volts2*1.75);

for (int i=0; i<400; i++) {
  dato = analogRead(5);
  if (dato > volts3) volts3 = dato;
}
//Serial.print(", "); Serial.print(volts3*1.75);

// -----
// Ahora intensidad luminosa
delay(80);
luz=analogRead(9); //Serial.print(" Luz: "); Serial.print(luz);

// -----
// Lectura sensor de gas combustible
gas = analogRead (7) * (10.00/1023);
// gas= analogRead(7);
// cadena=cadena+", "+amps1+", "+amps2+", "+amps3+", "+volts1+", "+volts2+", "+volts3+", "+luz+", "+gas+", \n";
//dtostrf(floatVar, minStringWidthIncDecimalPoint, numVarsAfterDecimal, charBuf);

char cadena_temp[7];

dtostrf(amps1,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena="> "+cadena_aux;

dtostrf(amps2,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+", "+cadena_aux;

```

```
dtostrf(amps3,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux;
```

```
dtostrf(volts1,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux;
```

```
dtostrf(volts2,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux;
```

```
dtostrf(volts3,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux;
```

```
dtostrf(luz,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux;
```

```
dtostrf(gas,7,5,cadena_temp);
cadena_aux=String(cadena_temp).substring(0,6);
cadena=cadena+" "+cadena_aux+" \n";
iteracion=iteracion+1;
```

```
if (num_cad==1){
  cadena_1=cadena_1+""+cadena;
  delay(50);
} else if (num_cad==2){
  cadena_2=cadena_2+""+cadena;
  delay(50);
} else if (num_cad==3){
  cadena_3=cadena_3+""+cadena;
  delay(50);
}
```

```
//-----
/* Impresión de resultados
  Serial.print(" ");
  Serial.print(amps1);
  Serial.print(" ");
  Serial.print(amps2);
  Serial.print(" ");
  Serial.print(amps3);
  Serial.print(" ");
  Serial.print(volts1);
  Serial.print(" ");
  Serial.print(volts2);
  Serial.print(" ");
  Serial.print(volts3);
  Serial.print(" ");
  Serial.print(luz);
  Serial.print(" ");
  Serial.print(gas);
  Serial.println(" ");
*/
delay(1000);
```

```
//-----
EthernetClient client = server.available();
if (client) {
  iteracion=1;
  num_cad=1;
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      if (c == '\n' && currentLineIsBlank) {

        //client.print(cadena);
        delay(30);
        client.print(cadena_1);
        delay(30);
        client.print(cadena_2);
        delay(30);
        client.print(cadena_3);
        delay(30);
        client.print(">");
        cadena="";
        cadena_1="";
        cadena_2="";
        cadena_3="";
        cadena="";
        cadena_1="";
        cadena_2="";
        cadena_3="";
        delay(60);
        num_cad=1;
        iteracion=1;
        break;
      }
      if (c == '\n') {
        currentLineIsBlank = true; // you're starting a new line
      }
      else if (c != '\r') {
        // you've gotten a character on the current line
        currentLineIsBlank = false;
      }
    }
  }
  // give the web browser time to receive the data
  delay(255);
  // close the connection:
  client.stop();
}
if(iteracion==20|| iteracion==40||iteracion==60){
  num_cad++;
}
}
```

B.2 dirCajas.php

```
<?php
```

```
/*
```

```
En este archivo se agrgan las direcciones ip de los arduinos responsables de
obtener los datos (SAD)
```

```
*/
```

```

$cajas = array(
"192.168.0.26",
"192.168.0.27",
"192.168.0.28",
"192.168.0.29",
"192.168.0.30",
"192.168.0.31"
);
?>

```

B.3 conectaDB.php

```

<?php
/*
Este archivo abre una sesion para manejar la base de datos
*/
/*Datos de conexion a la base de datos*/
$usuario="root";
$servidor="localhost";
$password="";
//$bd="logs";
$bd="cajas";
/*Inicia conexion a bd mysql */
if ($conexion=mysql_connect($servidor,$usuario,$password))
{
//print "Conexi&oacute;n exitosa: ".$conexion." <br> ";
/*Selecciona la base de datos sobre la que trabajar */
if (mysql_select_db($bd,$conexion))
{
//print "Conexion a la base de datos exitosa <br>".$bd;
}
else
{
//print "No se logro; realizar la conexion a la base de datos, sin privilegios para acceder a ".$bd;
}
}
else
{
//print "No se logro; realizar la conexion a la base de datos".mysql_get_error($conexion);
}
}
?>

```

B.4 guardaArcTotal.php

```

<?php
    $sql_todo=true;
    $dia=0;
    include "guardaArc.php";
    $sql_todo=false;
    header ("Location: ./index.php");

?>

```

B.5 index.php

```
<html lang = "es">
<head>
  <meta charset="utf-8"/>
  <meta http-equiv="refresh" content="70"/><!--MODIFICAR EL TIEMPO DE REFRESH DE LA PÁGINA-->
  <meta name="autor" content="JANR"/>
</head>

<body background="/img/fondoTesis.jpg">
<center>
<a href="http://www.ingenieria.unam.mx/" target="_self"></a>
<a href="http://www.unam.mx/" target="_self"></a>
</center>
<?php
/*
```

Este archivo muestra lo que está almacenado en la base de datos
*/

```

$dia=0; //variable para geerar reporte del dia actual
$hora_actual=floatval(date('H'));
$min_actual=floatval(date('i'));
if ($hora_actual==0 && ($min_actual==0)){
    $dia = 1; // variable para generar reporte del dia anterior
    include "guardaArc.php";
}
include "recupera.php";
//$var=0;

//include "dirCajas.php";
echo "<center>";
echo "<table border = '1'> \n";
echo "<tr>
  <td><font color=\"#FFFFFF\">CAJA_IP</font></td>
  <td><font color=\"#FFFFFF\">FECHA</font></td>
  <td><font color=\"#FFFFFF\">HORA</font></td>
  <td><font color=\"#FFFFFF\">AMP_1</font></td>
  <td><font color=\"#FFFFFF\">AMP_2</font></td>
  <td><font color=\"#FFFFFF\">AMP_3</font></td>
  <td><font color=\"#FFFFFF\">VOLT_1</font></td>
  <td><font color=\"#FFFFFF\">VOLT_2</font></td>
  <td><font color=\"#FFFFFF\">VOLT_3</font></td>
  <td><font color=\"#FFFFFF\">LUZ</font></td>
  <td><font color=\"#FFFFFF\">GAS</font></td>
  <td><font color=\"#FFFFFF\">TEMPERATURA</font></td>
</tr> \n";
include "conectaDB.php";
foreach($cajas as $caja){
  $querySelectAll="SELECT lectura_fecha, lectura_hora, lectura_amp1, lectura_amp2,
lectura_amp3, lectura_volt1, lectura_volt2, lectura_volt3, lectura_luz, lectura_gas, lectura_transiente, lectura_caja".
  " FROM lectura WHERE lectura_caja='".$caja.'" ORDER BY lectura_id DESC LIMIT 1;";
  $result = mysql_query($querySelectAll, $conexion);
  while ($row = mysql_fetch_row($result)){
    echo "<tr>
      <td><font color=\"#FFFFFF\">$row[11]</font></td>
      <td><font color=\"#FFFFFF\">$row[0]</font></td>
      <td><font color=\"#FFFFFF\">$row[1]</font></td>
      <td><font color=\"#FFFFFF\">$row[2]</font></td>
      <td><font color=\"#FFFFFF\">$row[3]</font></td>
      <td><font color=\"#FFFFFF\">$row[4]</font></td>
      <td><font color=\"#FFFFFF\">$row[5]</font></td>
      <td><font color=\"#FFFFFF\">$row[6]</font></td>
      <td><font color=\"#FFFFFF\">$row[7]</font></td>
    </tr>";
  }
}

```

```

                <td><font color="#FFFFFF">$row[8]</font></td>
                <td><font color="#FFFFFF">$row[9]</font></td>
                <td><font color="#FFFFFF">$row[10]</font></td>
            </tr> \n";
            //$var++;
            //if($var==6)
                //break;
        }
    }
    echo "</table> \n";
    echo "<a href='./guardaArcTotal.php' target='_self'>GENERAR REPORTE
TOTAL</a>";
    echo "</center>";
    include "cerrarDB.php";
?>
</body>
</html>

```

B.6 getValoresArduino.php

```

<?php
/*
En este archivo se obtienen los valores de los SAD (Arduinos)
mediante peticiones al SAD que funciona como servidor
*/
/*****/
function getValoresArduino($dir){
    $fp = fsockopen($dir, 80, $errno, $errstr, 0.5);
    if (!$fp) { //header ("Location: ./index2.php");
    }
    else
    {
        $out = "\nGET / HTTP/1.1\r\n";
        $out .= "Host: ".$dir."\r\n";
        $out .= "Connection: Close\r\n\r\n";

        //echo $out."<br>";
        fwrite($fp, $out);
        $dataString="";
        while (!feof($fp)) {
            try{
                $dataString = $dataString.fgets($fp, 1000)."\r\n";
            }catch (Exception $e){
                echo "exception";
            }
        }
        fclose($fp);
    }
    return $dataString;
}
/*****/
?>

```

B.7 cerrarDB.php

```

<?php

```

```

/*
Este archivo cierra la conexion, abierta por el archivo conectaDB.php
*/
mysql_close($conexion);
?>

```

B.8 recupera.php

```

<?php
include "getValoresArduino.php";
include "dirCajas.php";
include "conectaDB.php";

foreach($cajas as $dir){
    $lectura_fecha=date('d-m-Y');
    $lectura_hora =date('H:i:s');
    $infoToStorage = getValoresArduino($dir);
    $querySelect="SELECT MAX(lectura_id) FROM lectura;";
    if ($result = mysql_query($querySelect, $conexion))
    {
        //print "query exitoso ".$querySelect."<br>";
    }else
    {
        //print "query no exitoso ".$querySelect."<br>";
    }

    if (!$result) {
        echo "DB Error, no se puede conectar\n";
        echo 'MySQL Error: ' . mysql_error();
        exit;
    }
    $fila = mysql_fetch_row($result);
    if($fila[0]=="")
        $fila[0]=0;
    try{
        $lineValues = explode(">", $infoToStorage);
    }catch (Exception $e){
    }
    foreach($lineValues as $linea){
        try{
            $pos = strpos($linea, ",");
            if($pos !== false){
                $valor = explode(",", $linea);

                $lectura_id =++$fila[0];
                //echo $linea."<br>";
                $lectura_amp1=floatval($valor[1]);
                $lectura_amp2=floatval($valor[2]);
                $lectura_amp3=floatval($valor[3]);
                $lectura_volt1=floatval($valor[4]);
                $lectura_volt2=floatval($valor[5]);
                $lectura_volt3=floatval($valor[6]);
                $lectura_luz=floatval($valor[7]);
                $lectura_gas=floatval($valor[8]);
                $lectura_transiente=0;
                $lectura_caja=$dir;
                $query="INSERT INTO lectura(
                `lectura_id`,
                `lectura_fecha`,

```

```

`lectura_hora`,
`lectura_amp1`,
`lectura_amp2`,
`lectura_amp3`,
`lectura_volt1`,
`lectura_volt2`,
`lectura_volt3`,
`lectura_luz`,
`lectura_gas`,
`lectura_transiente`,
`lectura_caja`
)VALUES(
"$lectura_id.",
"$lectura_fecha.",
"$lectura_hora.",
"$lectura_amp1.",
"$lectura_amp2.",
"$lectura_amp3.",
"$lectura_volt1.",
"$lectura_volt2.",
"$lectura_volt3.",
"$lectura_luz.",
"$lectura_gas.",
"$lectura_transiente.",
"$lectura_caja.");

if ($value = mysql_query($query, $conexion))
{
//print "Se inserto la medicaci&oacute;n: ".$query."<br>";
}else
{
print "NO se logro insertar la medicion:".$query."<br>";
}
}
}catch (Exception $e){
}
}
}
include "cerrarDB.php";
?>

```

B.9 guardaArc.php

```
<?php
```

```
/*
```

En este archivo se generan reportes con extensión .csv que pueden ser interpretados por las hojas de cálculo

```
*/
```

```

include "conectaDB.php";
// $ruta_archivo = ".\\Archivos\\";//para windows
$ruta_archivo = "./Archivos";//para linux
$var_fech=date('m-d')." ";
$var_fech_hoy=explode("-", $var_fech);
$var_fech_ayer="(floatval($var_fech_hoy[1])-$dia)."-".date('m-Y');
$var_hor=date("H.i.s")." ";
$archivo = fopen($ruta_archivo."REPORTE_".$var_fech."_".$var_hor.".csv",'w');

if($sql_todo){

```

```

        $extra="";
    }else{
        $extra="where lectura_fecha = ".$var_fech_ayer."";
        $sql_todo=false;
    }
    if($sarchivo == false){
        die("No se ha podido crear el archivo.");
    }else{
        $var=0;
        //$querySelectAll="SELECT lectura_fecha, lectura_hora, lectura_f1, lectura_f2, lectura_contador,
lector_id FROM lectura ORDER BY lector_id DESC;";
        $querySelectAll="SELECT lectura_caja, lectura_fecha, lectura_hora, lectura_amp1, lectura_amp2,
lector_id FROM lectura ORDER BY lector_id DESC;";
        $result = mysql_query($querySelectAll, $conexion);
        //$variables="FECHA, HORA, LECTURA 1, LECTURA 2, LECTURA 3\r";
        $variables="CAJA_IP, FECHA, HORA, AMPERAJE 1, AMPERAJE 2, AMPERAJE 3, VOLTAJE 1,
VOLTAJE 2, VOLTAJE 3, LUZ, GAS, TRANSIENTES\r";
        fwrite($sarchivo, $variables);
        while ($row = mysql_fetch_row($result)){
            fwrite($sarchivo,
$row[0].",".$row[1].",".$row[2].",".$row[3].",".$row[4].",".$row[5].",".$row[6].",".$row[7].",".$row[8].",".$row[9].",".$r
ow[10].",".$row[11]."\r");
            $var++;
            //if($var==10)// si se desea crear un archivo con toda la informacion de la base de
datos comentar el if y break
                //break;
        }
        fclose($sarchivo);
    }
    include "cerrarDB.php";
    //header ("Location: ./muestraDB.php");
?>

```

APÉNDICE C: OTROS CONCEPTOS

De acuerdo al trabajo de esta tesis se desarrolló un SAD con un propósito general y después se modificó para darle un uso en específico, sin embargo para profundizar más en el tema, en este apéndice se incluye una investigación acerca de sensores y actuadores, así como el tipo de interferencias que se pueden tener en el momento de realizar una medición.

C.1 Transductores

En esta parte se pretende profundizar en los transductores, sin embargo existen infinidad de transductores que sería imposible de describir todos, por consiguiente se realizó una síntesis de algunos transductores relevantes descritos en el Capítulo 5. "Transductores" del autor B.A.Gregory de la obra *Instrumentación eléctrica y sistemas de medida*, (1984). [6]

Los transductores se definen como componentes capaces de interconectar sistemas similares o distintos, con la finalidad de transmitir energía entre ellos. Pero en este caso en el campo de la Ingeniería se define como un dispositivo utilizado para convertir un fenómeno físico en una señal eléctrica. Los transductores se pueden clasificar según el método de transducción o según la variable que son capaces de medir. [Tabla C.1]

Tabla C.1 Tabla de transductores.

	Resistencia variable	Reactancia variable	Electromagnético	Semiconductor	Digital	Termoeléctrico
Aceleración	*	*	*	*	*	
Desplazamiento	*	*	*	*	*	
Caudal	*	*	*			
Fuerza	*	*		*		
Humedad	*					
Nivel	*	*			*	
Presión	*	*	*	*		
Temperatura	*			*	*	*
Espesor	*				*	
Velocidad	*	*	*	*	*	

C.1.1 Transductores resistivos

Este tipo de transductor funciona como sistema de medida de una posición, sólo hace falta conectar el contacto móvil del resistor al movimiento o desplazamiento que se desea medir. Un cambio en la variable medida producirá un cambio en la corriente. Para que este transductor funcione satisfactoriamente, la tensión deberá de ser constante independientemente de la resistencia del circuito, de modo que la corriente solamente dependa de las variaciones de la resistencia R . Otra ventaja que tiene este tipo de transductores es que se pueden observar variaciones n de desplazamientos en dos sentidos (positivo y negativo). [Figura C.1]

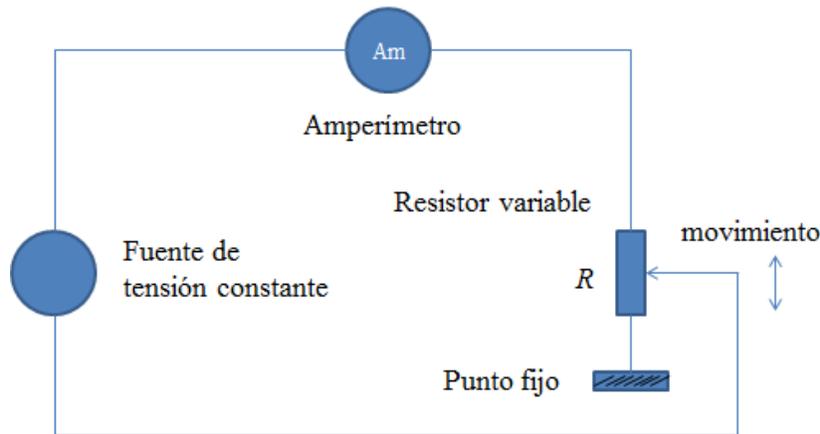


Figura C.1: Transductor de posición de resistencia variable.

Transductores potenciométricos

Un transductor de resistencia variable produce una variación de corriente, que está en función de la variable a medir. Un transductor potenciométrico se debe alimentar con una fuente de corriente constante, y midiendo la salida del transductor con un instrumento de alta impedancia, de modo que se desprecian los efectos de la carga sobre el potenciómetro. [Figura C.2]

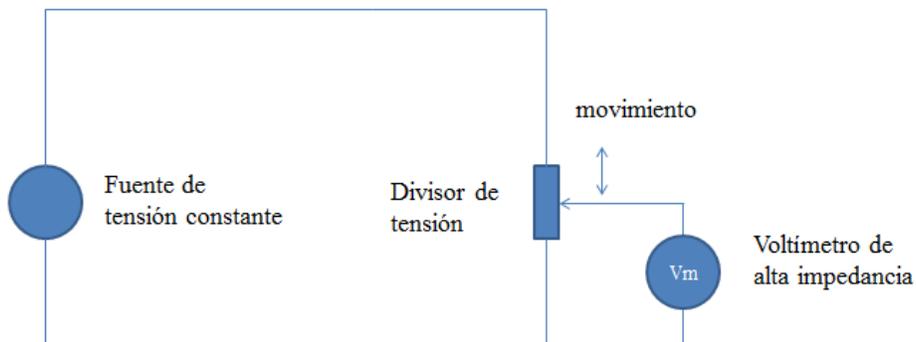


Figura C.2: Principio del transductor de posición potenciométrico.

El funcionamiento de este tipo de transductores depende del contacto del cursor y el elemento resistivo, su vida depende del desgaste de este contacto.

Galgas extenciométricas resistivas

Si a un hilo de un material conductor se le alarga produciendo un aumento en su longitud ∂L , y mientras no rebase el límite elástico del material, el cambio de la longitud es proporcional a la carga, tanto el aumento y la reducción del área, contribuyen a un cambio en la resistencia del hilo tensado. Además al cambiar las dimensiones del material también contribuyen al cambio de resistividad del material, conocido a este efecto como piezorresistividad. Así que el tamaño de una galga extenciométrica depende directamente de la aplicación en cuestión. [Figura C.3]

Un factor que puede afectar al funcionamiento de una galga extenciométrica resistiva es la temperatura, ya que tiene efectos triples: en primer lugar el coeficiente del hilo de la galga con la que fue diseñada tiene un coeficiente de temperatura el cual es afectado a cambios de esta, en segundo lugar las galgas extenciométricas están diseñadas con distintos materiales afectando de manera diferente los cambios de temperatura a cada uno de ellos. Por último el problema se puede encontrar en el objeto de medida ya que dependiendo del coeficiente de dilatación del mismo este puede expandirse o contraerse debido a cambios de temperatura.

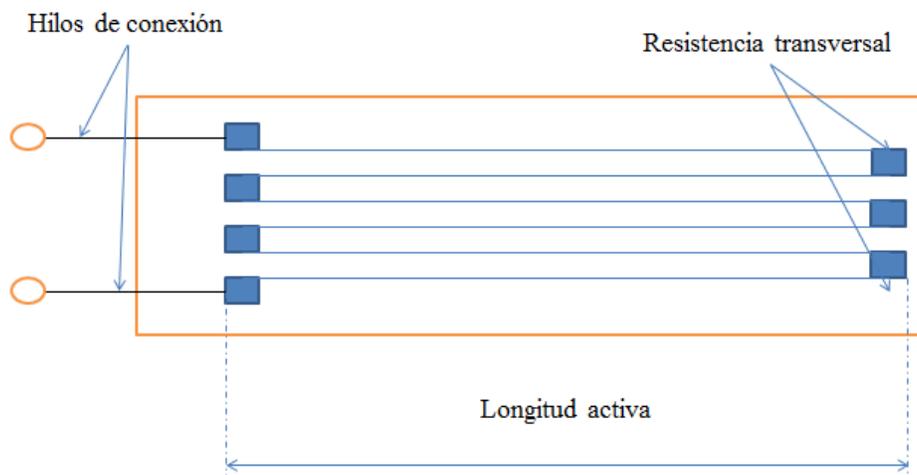


Figura C.3: Galga extenciométrica resistiva.

Anemómetros de hilo caliente

Para medir el caudal del aire se puede utilizar un transductor resistivo, consistente en un hilo de wolframio o de una aleación de platino por el que se hace circular una corriente. La magnitud de la corriente es suficiente para calentar el hilo, y cualquier movimiento del aire a su alrededor lo enfría y produce un cambio en la resistencia, y por consiguiente una caída en la tensión del hilo.

Transductores de humedad

La humedad es una medida de la cantidad de vapor de agua presente en un gas. Se puede describir de varias formas distintas, pero la expresión que más frecuentemente se usa es en forma de humedad relativa, que es la relación entre tensión del vapor de agua presente en la mezcla de gas y vapor de agua, y la tensión necesaria para saturar el gas a la temperatura de medida. Esta expresión se expresa en porcentaje (% h.r.), y depende de la temperatura. [Figura C.4]

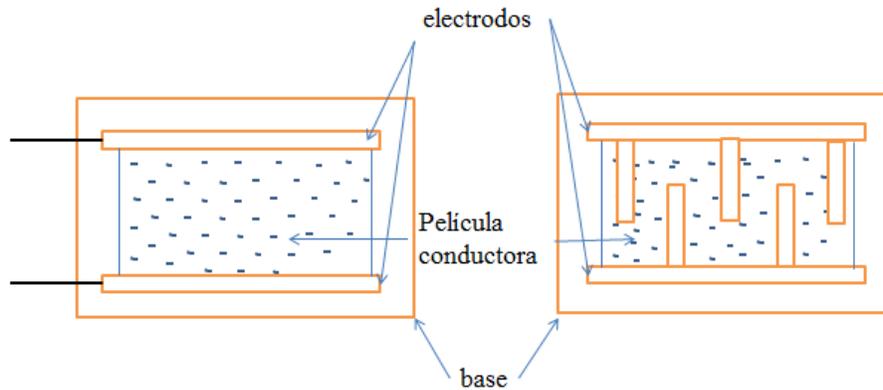


Figura C.4: Transductores de humedad con sensor resistivo.

C.1.2 Transductores de reactancia variable

En este tipo de transductores, un desplazamiento modula una resistencia capacitiva o inductiva. Dado que las variaciones de reactancia sólo pueden medirse cuando la alimentación es una corriente alterna.

Capacitancia variable

Para cualquier fenómeno físico un cambio se puede convertir en un desplazamiento, dado que la capacitancia entre dos placas conductoras es aproximadamente proporcional a $\epsilon A/d$, donde ϵ es la constante dieléctrica de las placas, A es el área de las placas, y d es la distancia que hay entre ellas.

Inductancia variable

La inductancia de una bobina depende del modo en que las líneas de flujo magnético conectan sus arrollamientos, utilizando convertidores adecuados para producir un cambio en la concatenación del flujo magnético en una bobina, se pueden obtener una serie de transductores que presenten un cambio de inductancia proporcional a la variable medida.

C.1.3 Transductores con semiconductores

Las propiedades eléctricas de los dispositivos semiconductores quedan afectadas por las variaciones de temperatura, iluminación y fuerza. La sensibilidad de estos materiales es mucho mayor que la de otros materiales.

Termistores

Se puede fabricar un transductor de temperatura muy sensible, denominado termistor, mediante la sintetización de óxidos de materiales tales como manganeso, cobalto, cobre, níquel, hierro o uranio, formando pequeñas gotas y luego recubriéndolas con resina epoxy o vidrio para protegerlas, la variación de resistencia es exponencial, contrariamente a la resistencia hecha de metal que presenta características casi lineales. En la figura C.5 se puede observar la relación resistencia/temperatura.

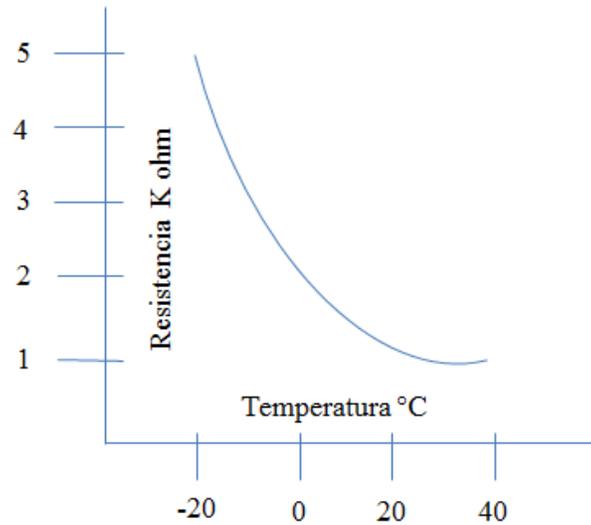


Figura C.5: Relación resistencia/temperatura para un termistor.

Galgas extensiométricas semiconductoras

Los materiales utilizados para la fabricación de estos componentes se les denominan piezorresistivos, esto significa que cuando a la galga se le somete a una fuerza, en la estructura cristalina hay una alteración y un cambio en la resistencia eléctrica. Las galgas normalmente están fabricadas de silicio mientras que los terminales de conexión pueden ser de cinta de oro, plata o níquel. Las longitudes efectivas de las galgas están entre 2 y 7 [mm]. [Figura C.6]

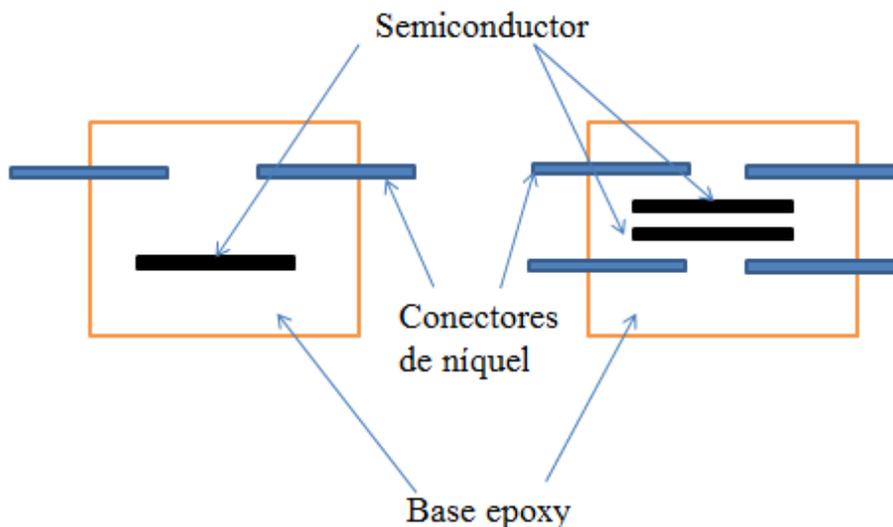


Figura C.6: Galgas extensiométricas semiconductoras.

Este tipo de transductor tiene algunos inconvenientes, son componentes no lineales, tienen un efecto con variaciones de la temperatura debido al conductor. Por lo tanto cuando se utilicen galgas extensiométricas semiconductoras hay que agregar al sistema una compensación de temperatura en el circuito de medida.

Fotodiodos y fototransistores

Las características eléctricas de la mayoría de los diodos y transistores quedan afectadas por la luz, y de ahí que sea normal evitar que la luz actúe en las uniones de los dispositivos, fabricándolos dentro de una capa opaca que no permita el paso de la luz. Los fotodiodos y los fototransistores se construyen de forma que realzan esta propiedad. La mayoría de los semiconductores sensores de luz son de silicio, pero cuando se requiere una respuesta óptima en la región de infrarrojo, pueden utilizarse dispositivos de germanio.

C.1.4 Transductores pasivos

Este tipo de transductores no requieren una fuente de alimentación externa para que funcionen.

Transductores electromecánicos

Para obtener la generación de energía en los transductores electromagnéticos, el flujo magnético generalmente lo proporciona un imán permanente. [Figura C.7]

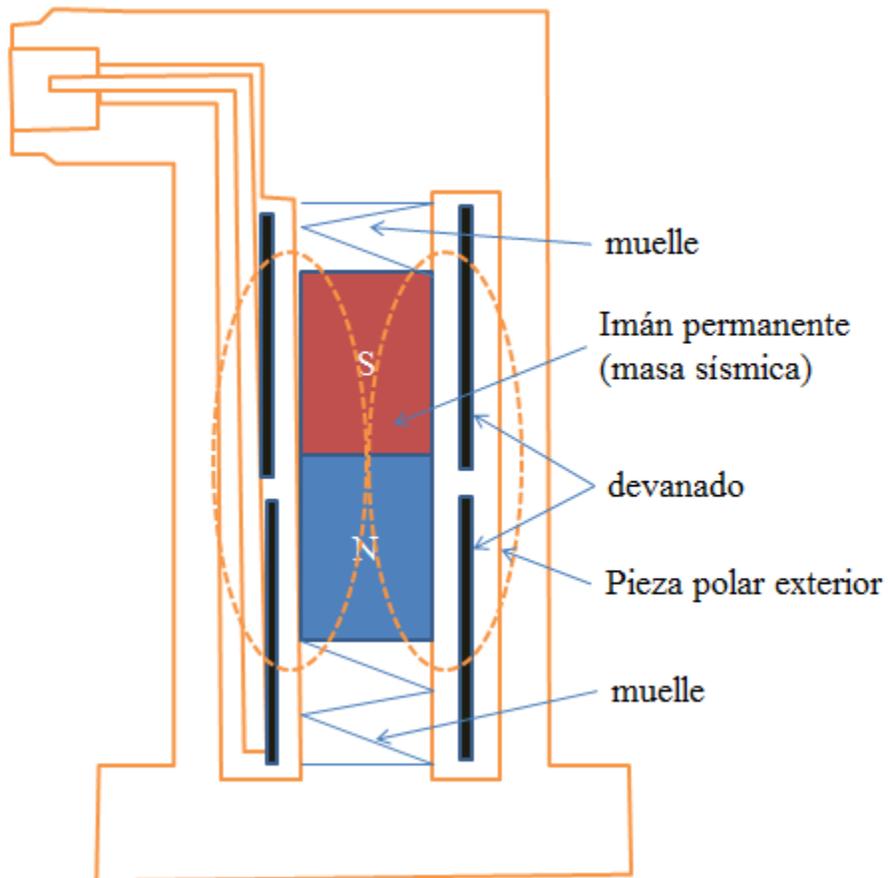


Figura C.7: Transductor de vibraciones electromagnético.

Transductor de velocidad lineal

La versión más simple del transductor electromagnético es cuando un imán permanece unido al eje móvil pero se puede mover libremente en el interior de una bobina cilíndrica, haciendo así que el movimiento del objeto/eje /imán genere una tensión en las terminales de la bobina.

Tacómetro con rotor dentado

Los tacómetros que utilizan un rotor de material ferromagnético dentado, y una bobina de transducción devanada alrededor de un imán permanente son probablemente el tipo de transductor de velocidad angular que tiene como salida una frecuencia variable. [Figura C.8]

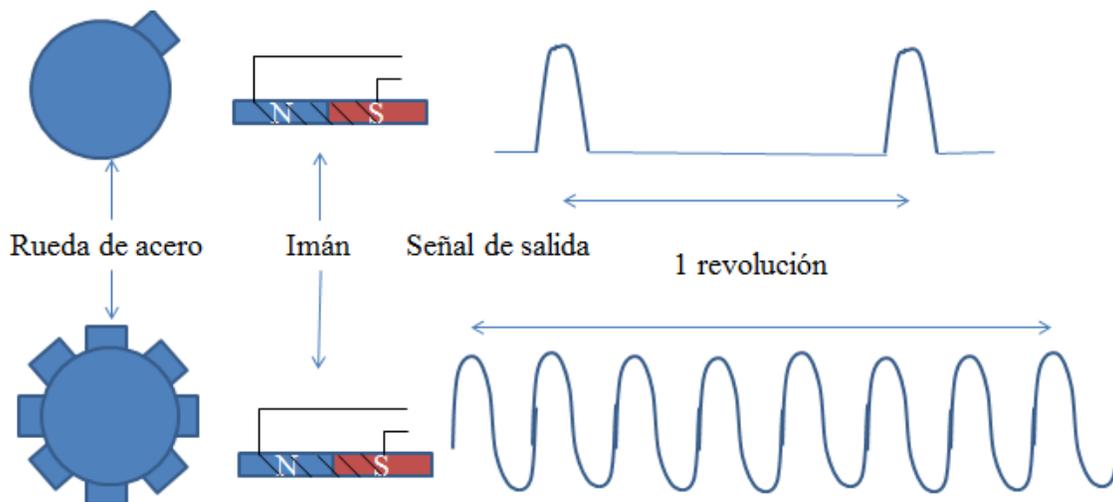


Figura C.8: Tacómetro de rueda dentada.

Transductores piezoeléctricos

Para generar una señal eléctrica, estos dispositivos utilizan las características piezoeléctricas de ciertos cristales y materiales cerámicos. En 1880 Pierre y Jacques Curie encontraron que si se colocaban pesas sobre un cristal de cuarzo se podía generar una carga eléctrica, estudios posteriores han descubierto que alrededor de cuarenta materiales cristalinos si se someten a una presión pueden generar una carga eléctrica, también descubrieron de manera similar el efecto inverso, es decir si se le aplica una carga a un cristal de cuarzo este cambia sus dimensiones.

Existen dos tipos de cristales: los naturales y los sintéticos, siendo estos últimos cerámicos sobre todo, la adición de impurezas tales como titanato de calcio, mejoraba algunas de las características de los cristales.

Todos los cristales piezoeléctricos tienen una impedancia de salida elevada, y por esta razón el instrumento al que se le conecte el transductor piezoeléctrico deberá tener una alta impedancia o se deberá conectar un amplificador de carga entre el transductor y el instrumento.

C.1.5 Transductores termoelectricos

En 1834, Jean Charles Peltier, descubrió que al circular una corriente por la unión de dos conductores distintos estos cambiaban su temperatura, veinte años después William Thomson descubrió que si un conductor por el que circulaba una corriente se calentaba en un punto concreto los puntos equidistantes de la fuente de calor a lo largo del conductor se mantenían a temperaturas distintas, este es el principio de funcionamiento del termopar.

Termopar

El efecto *Seebeck*, que es la combinación de los efectos Peltier y Thomson es la aplicación que se le da al Termopar. [Figura C.9]

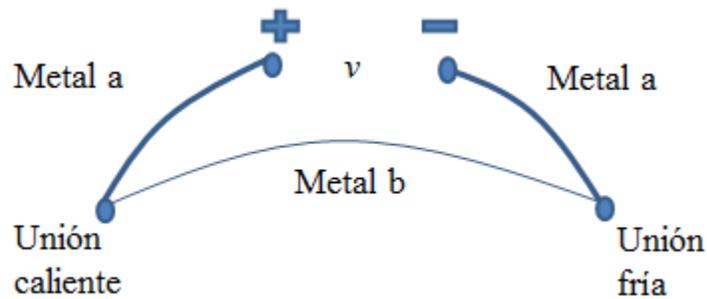


Figura C.9: Circuito de termopares básico.

Estos transductores constan con un par de barras o hilos de metales distintos por ambos extremos: un extremo se utiliza como unión caliente (sensor) mientras que el otro se utiliza como unión fría como referencia.

Cuando se conectan en serie varios termopares de los mismos materiales, con todas las uniones calientes a una temperatura y todas las uniones frías a otra, se dice que constituyen una termopila. La tensión de salida de una termopila es igual a la suma de los termopares utilizados. [Figura C.10]

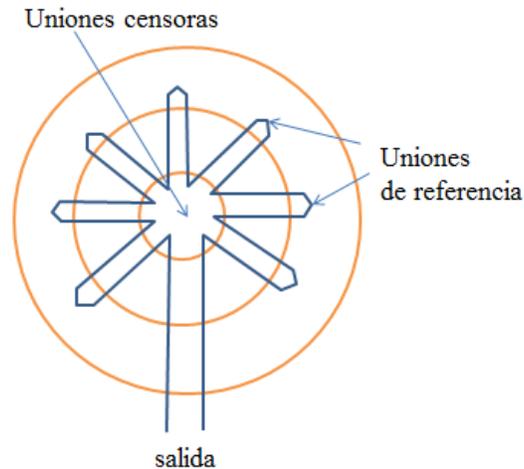


Figura C.10: Termopila.

C.1.6 Transductores ultrasónicos

Estos dispositivos se desarrollan utilizando las propiedades magnetoresistivas de los materiales ferromagnéticos, la cual se refiere a los cambios de forma de un material ferromagnético producidos por la variación de un campo magnético alterno, un buen ejemplo de este tipo de transductores es el sonar, y se aplican para la medida de flujos y espesores, la detección de imperfecciones y los sistemas de detección de intrusos.

C.1.7 Transductores digitales

Los transductores digitales son aquellos cuya salida está representada por una serie de incrementos discretos, aquellos dispositivos que tienen una salida en forma de impulsos que se pueden agregar a un contador digital, un buen ejemplo es el tacómetro, los dispositivos que generan un único tren de impulsos no pueden dar una indicación de dirección, pero esto se soluciona con el tacómetro de rotor dentado. Otro ejemplo son los codificadores digitales, estos son dispositivos con salida en forma de impulsos, capaces de detectar un cambio en la variable de medida.

C.2 Actuadores

Un actuador es un dispositivo capaz de generar una fuerza o acción a partir de energía hidráulica, energía eléctrica o energía neumática. El actuador se activa cuando recibe la orden de un controlador y tiene efecto directo en algún elemento final, como por ejemplo un motor.

De acuerdo al tipo de energía que maneja el actuador se pueden clasificar en tres tipos:

C.2.1 Actuadores hidráulicos

El funcionamiento de este tipo de actuadores se basa en la presión ejercida por algún líquido, que generalmente se utiliza algún tipo de aceite. Por lo general este tipo de actuadores se utilizan en aplicaciones donde requieran soportar una carga pesada.

Todos los actuadores de tipo hidráulico deben de estar herméticamente sellados para que no se derrame el líquido que contienen y para mantener a presión en el mismo ya que la presión en el fluido es la que hace que el actuador funcione, este tipo de actuadores requiere un mantenimiento constante y equipo que suministre una energía adecuada.

Cilindro hidráulico

Delgado, A. [4] clasifica a los actuadores hidráulicos de acuerdo con su función, se puede clasificar a los cilindros hidráulicos en 2 tipos: de efecto simple y de acción doble. En el primer tipo se utiliza fuerza hidráulica para empujar y una fuerza externa, diferente, para contraer. El segundo tipo se emplea la fuerza hidráulica para efectuar ambas acciones.

En el interior poseen un resorte que cambia su constante elástica con el paso de la corriente. Es decir, si circula corriente por el pistón eléctrico este puede ser extendido fácilmente.

Cilindro de presión dinámica

Lleva la carga en la base del cilindro. Los costos de fabricación por lo general son bajos ya que no hay partes que resbalen dentro del cilindro. [Figura C.11]

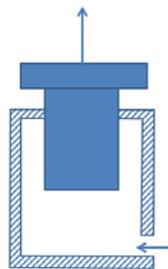


Figura C.11: Cilindro de presión dinámica.

Cilindro de efecto simple

Una barra es colocada en un extremo del pistón, cuando la presión es ejercida en la parte contraria al extremo del pistón donde está la barra, esta sube hasta donde la presión lo empuje, ejerciendo una fuerza sobre la barra de contracción, después la barra es regresada a la posición inicial por la simple acción de resortes o de la gravedad. La carga sólo puede colocarse en un extremo del cilindro. [Figura C.12]

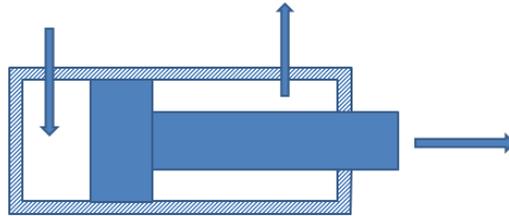


Figura C.12: Cilindro de efecto simple.

Cilindro de efecto doble

La carga puede colocarse en cualquiera de los lados del cilindro. Se genera un impulso horizontal debido a la diferencia de presión entre los extremos del pistón [4] [Figura C.13].

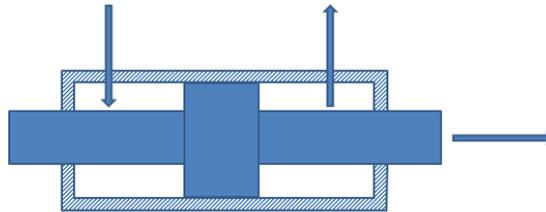


Figura C.13: Cilindro de efecto doble.

Cilindro telescópico

La barra de tipo es empujada sucesivamente conforme se va aplicando al cilindro aceite a presión. Se puede lograr una carrera relativamente en comparación con la longitud del cilindro [4] [Figura C.14]

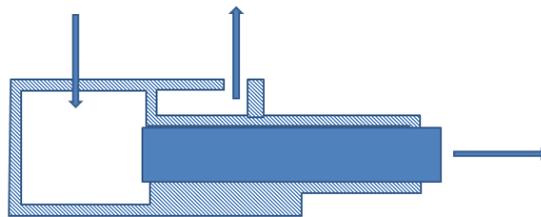
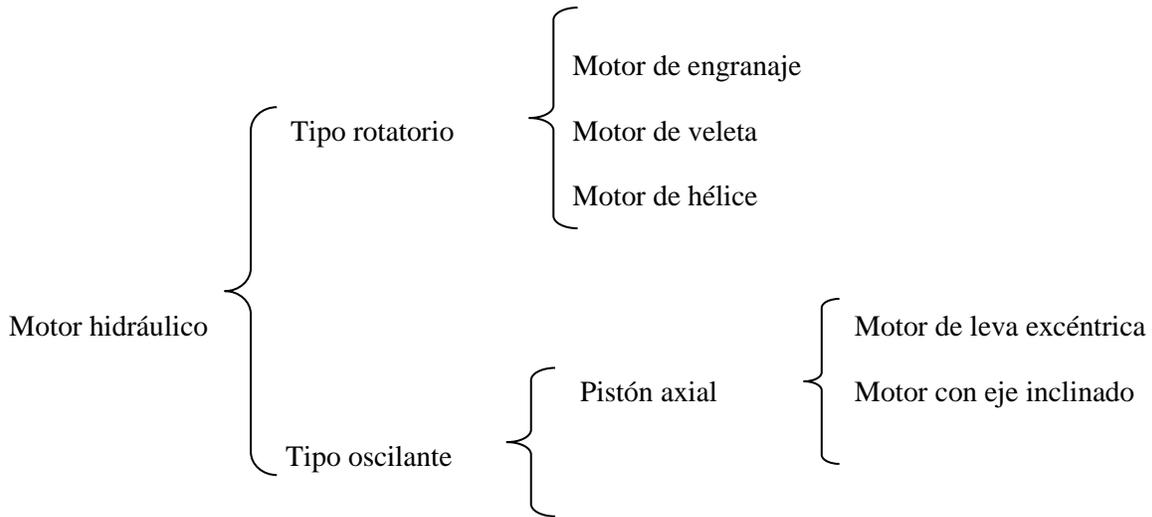


Figura C.14: Cilindro telescópico.

Motor hidráulico

En este tipo de motores el giro generado es a causa de la presión, se pueden clasificar en dos grupos, el rotatorio y el oscilante, el primero los engranes son movidos por la presión ejercida en el fluido, el otro su movimiento se debe a la acción oscilatoria de un pistón.



Motor de Engranaje.

El motor tiene una entrada de aceite, el cual mueve los pistones cuando entra a presión moviendo los engranes, se usa principalmente en operaciones de alta velocidad. [Figura C.15]

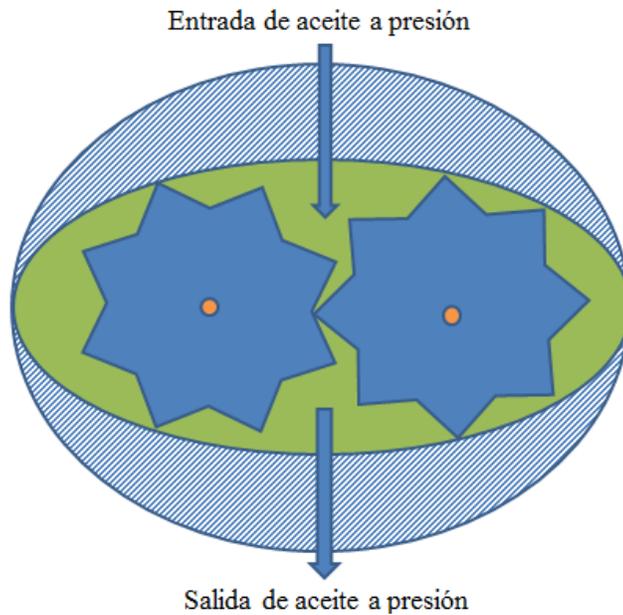


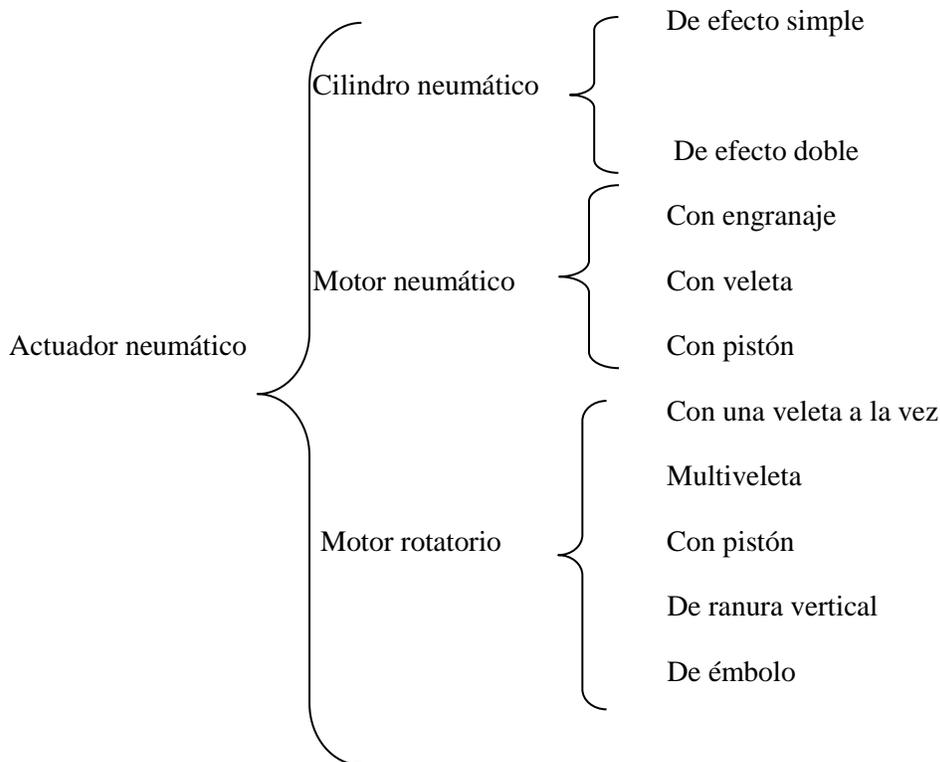
Figura C.15: Motor de engranaje.

Motor con pistón eje inclinado

El fluido entra a presión llenando los pistones que en primera instancia encuentra, generando un giro del motor, este tipo de motores al introducir el aceite a presión del lado opuesto hace que el motor gire en otro sentido debido a la inclinación del eje funciona de igual manera que el motor de engranaje pero a diferencia de que el eje varia de inclinación para poder cambiar las propiedades del giro.

C.2.2 Actuadores neumáticos

Los actuadores neumáticos también utilizan un fluido para que funcionen pero a diferencia de los anteriores estos utilizan aire comprimido pudiéndose comprimir aún más.



C.2.3 Actuadores eléctricos

Como la electricidad es el medio por el cual se activan este tipo de actuadores la velocidad de respuesta es inmediata y sin importar la distancia. Por ejemplo los motores, existen en gran cantidad de modelos y es fácil utilizarlos, están estandarizados según la aplicación. En la mayoría de los casos es necesario utilizar reductores, debido a que los motores son de operación continua.

Relevador

El actuador eléctrico más utilizado es el relevador y su función es prácticamente la de un *switch* activado por corriente eléctrica, específicamente por un campo magnético. El relevador funciona de manera que puedan manejarse dos corrientes diferentes completamente aisladas, una corriente es la que activa el relevador, produciendo en una bobina un flujo de corriente y por lo tanto también un campo magnético, que afecta directamente a un imán que funciona como interruptor. [Figura C.16]

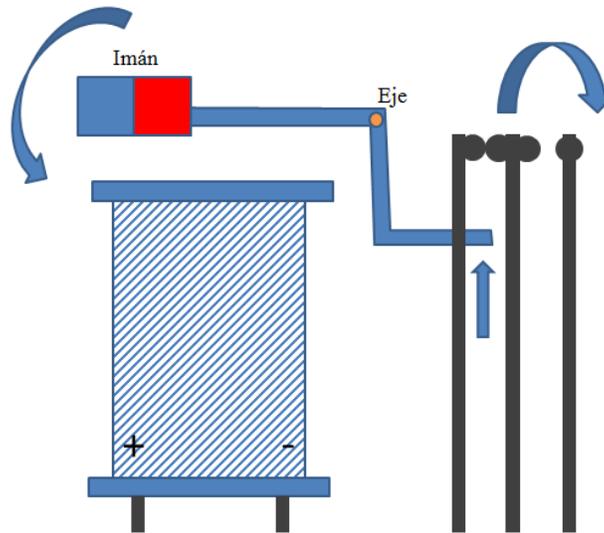


Figura C.16: Funcionamiento de un relevador.

Alambres musculares

Los Alambres Musculares son actuadores que quedan activados con corriente eléctrica estos se contraen generando fuerzas desde los 20 a los 2000 gramos, dependiendo de su diámetro.

Motores eléctricos

Los motores eléctricos son dispositivos que utilizan la corriente eléctrica y la convierten en energía mecánica, los motores pueden ser de corriente directa o alterna.

- Funcionamiento

Si un conductor por el que circula una corriente eléctrica se encuentra dentro de la acción de un campo magnético, éste tiende a desplazarse perpendicularmente a las líneas de acción del campo magnético. El conductor tiende a funcionar como un electroimán debido a la corriente eléctrica que circula por el mismo adquiriendo de esta manera propiedades magnéticas, que provocan, debido a la interacción con los polos ubicados en el estator, el movimiento circular que se observa en el rotor del motor.

- Motores AC

Motor síncrono. Los imanes del campo se montan sobre un rotor y se excitan mediante corriente continua, y las bobinas de la armadura están divididas en tres partes y alimentadas con corriente alterna trifásica.

Motor asíncrono. La variación de las tres ondas de corriente en la armadura provoca una reacción magnética variable con los polos de los imanes del campo, y hace que el campo gire a una velocidad constante, que se determina por la frecuencia de la corriente en la línea de potencia de corriente alterna.

- Motores CD

Motor serie. Los devanados se encuentran conectados en serie. [Figura C.17]

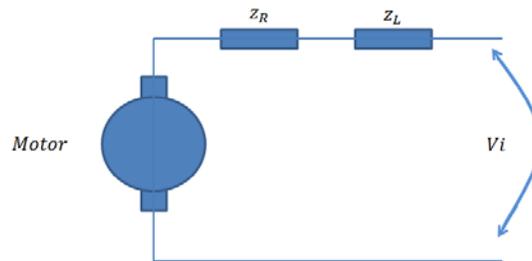


Figura C.17: Motor serie.

Motor paralelo. Los devanados del motor y el inductor se encuentran en paralelo. [Figura C.18]

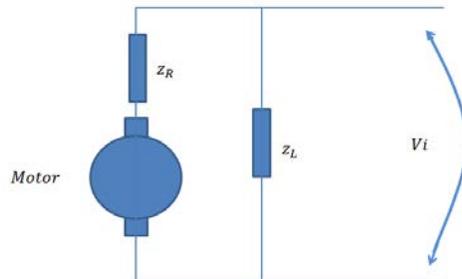


Figura C.18: Motor paralelo.

Motores a pasos. Tienen diferente número de devanados, y cada uno con un par de conexiones independiente (hilos), generalmente los motores a pasos constan de 4 hilos es decir 2 devanados utilizados en trabajos de precisión.

Diferentes pasos del motor de acuerdo a la polarización de cada hilo.

Paso/hilo	A	B	C	D
1	V+	V+	V-	V-
2	V-	V+	V+	V-
3	V-	V-	V+	V+
4	V+	V-	V-	V+

C.3 Interferencias

Para la investigación referente a interferencias se tomó como base el Capítulo 7. “*Interferencias y apantallamiento*” del autor B.A.Gregory de la obra *Instrumentación eléctrica y sistemas de medida, (1984)*. [7]

La conducción de una señal eléctrica desde el punto de medida al instrumento, puede verse afectada en la práctica por interferencias de diversas clases.

C.3.1 Efectos ambientales

El efecto final de una interferencia del ambiente se manifiesta generalmente cuando un cambio gradual o deriva, en la salida de una parte del sistema y medida cuando no hay cambio en la entrada.

Efectos térmicos

Son los más comunes de la causa de deriva y pueden ser debidos al calentamiento de componentes dentro de un instrumento o a cambios en la temperatura del ambiente.

Coefficiente de dilatación

Los cambios de temperatura producen cambios dimensionales que dependen del coeficiente de expansión térmica del material del que están contruidos los componentes, por ejemplo, cambios en las dimensiones de los inductores que a su vez producen un cambio de magnitud.

Coefficiente de resistencia

El coeficiente de temperatura resistivo depende directamente de la temperatura. Por lo tanto, en sistemas de instrumentación delicados es imprescindible utiliza componentes con pequeño coeficiente de temperatura.

Efecto termoeléctrico

En el desarrollo de un circuito son inevitables las uniones entre diferentes tipos de metales, y si estos tienen un coeficiente de temperatura distinto a los demás puede que afecten las propiedades de los componentes que estén cerca de ellos cuando estén trabajando conjuntamente.

Efectos de la humedad

Los efectos de la humedad en el aire tienden en general a ser pequeños aunque algunos materiales empleados en la fabricación de componentes hidrosféricos y estos afectan al valor del componente al cambiar la humedad. Esto puede ser debido a una variación en las propiedades conductoras o dieléctricas de los materiales. También puede ser debido a la absorción de la humedad de los materiales (por ejemplo la madera) que produce un cambio en las dimensiones y un cambio por consiguiente en el valor del componente.

Presión

Los efectos de los cambios de la presión del ambiente son más probables que ocurran en aquellos casos en que intervienen las propiedades dieléctricas del aire a presión atmosférica.

Tierras múltiples

Idealmente todas las tierras de un sistema están al mismo potencial (cero), pero esto sólo puede ser cierto si el camino de puesta a tierra tiene resistencia o inductancia nula, el acoplo capacitivo entre el camino de puesta a tierra y el conductor de la señal puede producir una interferencia electrostática que se suma a la señal, también si la corriente de retorno se suma a la corriente de tierra de otros circuitos estas aparecerán como un aumento en el nivel de la señal.

Ruido inherente

Parte del ruido o señales presentes en un sistema de medida delicado se genera en los componentes o partes del sistema, por ejemplo el ruido eléctrico surge de conexiones deficientes, que pueden ser debidas a soldaduras malas o al polvo en los contactos del interruptor.

Imperfecciones en los componentes

Es casi imposible fabricar un componente de tal modo que sólo tenga una propiedad, ya que al fabricarlos intervienen muchos factores para su correcto funcionamiento.

Imperfecciones en resistencias

El empleo de componentes resistivos, principalmente resistencias es frecuente en la instrumentación, y dado que muchas resistencias se fabrican devanando un hilo resistivo en un carrete, que posee cierta inductancia.

Así como las resistencias también los demás componente tienen características propias que afectan o pueden afectar a la medición del sistema.

C.3.2 Interferencias electrostáticas

Son debidas al acoplamiento que existe entre superficies conductoras.

C.3.3 Interferencias electromagnéticas

Dado que este tipo de interferencias resulta del acoplamiento electromagnético entre un conductor por donde circula la corriente y otros conductores, las fuentes más probables de interferencia de esta clase son las conexiones de las fuentes de alimentación y los transformadores.

C.3.4 Diafonía

Cuando se interconectan circuitos de señales distintas mediante un conductor plano múltiple o un cable multifilar, se pueden producir interferencias o diafonías entre los circuitos.

C.3.5 Rechazo de ruido

Ruido de modo normal

Este término es aplicado a todas las tensiones de interferencia continuas y alternas que aparecen ante el instrumento o circuito de medida como si fuesen parte de la señal medida.

Si la señal medida es en efecto una tensión continua, las tensiones de interferencias debidas a efectos galvánicos y termoelectricos dentro de la fuente de señal son, para el instrumento, indistinguibles de la cantidad real que se desea medir.

Las interferencias de modo normal alternas pueden ser debidas a acoplamientos de campos electrostáticos y electromagnéticos externos con los componentes y conexiones de la fuente de señal o a efectos intrínsecos tales como el ruido browniano dentro de los componentes.

Ruido de modo común

Este es el término que se utiliza para incluir todas las tensiones continuas y alternas que están presentes entre la tierra del instrumento de medida y la conexión de menor potencial de la señal que se mide. Por lo tanto, la corriente de modo común puede ser una combinación de: El nivel de tensión respecto a la tierra, a que esté la señal medida; Los efectos de corrientes debidos a bucles de tierra; Resultado de otras fuentes de interferencia, por ejemplo acoplamientos electrostáticos y electromecánicos.

Relación señal a ruido

En todos los instrumentos electrónicos hay ruido generado en su interior, y se utilizan amplificadores de alta ganancia para hacer medidas de señales pequeñas, hace falta un criterio con el que se pueda juzgar el efecto de este ruido de origen interno. La relación señal-ruido se utiliza con este fin, y se define como la relación de nivel de la señal al nivel de ruido.

$$S/N = 20 \log_{10} \left(\frac{V_s}{V_n} \right) [dB]$$

TABLA DE ILUSTRACIONES

Figura 1.1: Señales en tiempo continuo y en tiempo discreto.....	6
Figura 1.2: Ejemplo de una señal periódica.....	7
Figura 2.1: Sistema básico de medición.....	13
Figura 2.2: Estructura de un microprocesador.....	16
Figura 2.3: Estructura de un microcontrolador.....	16
Figura 2.4: Arquitectura Von Newmann.....	17
Figura 2.5: Arquitectura Harvard.....	18
Figura 2.6: Esquema de la placa de Arduino Uno.....	23
Figura 2.7: Esquema de la placa de la Raspberry-Pi.....	32
Figura 2.8: Sistemas operativos de la Raspberry-Pi.....	35
Figura 2.9: Entorno de un sistema de base de datos simplificado.....	39
Figura 3.1: Esquema simple del SAD.....	41
Figura 3.2: Ampliación y diseño general de un SAD.....	42
Figura 3.3: <i>Secure Shell</i>	43
Figura 3.4: <i>Wizard</i> del instalador de Aduino.....	44
Figura 3.5: Funcionamiento correcto del servidor Apache.....	45
Figura 3.6: Archivo de prueba php.....	45
Figura 3.7: Prueba php.....	46
Figura 3.8: Ingresar contraseña en MySQL.....	46
Figura 3.9: Instalador de XAMPP.....	47
Figura 3.10: Herramientas a instalar en XAMPP.....	48
Figura 3.11: Carpeta de instalación de XAMPP.....	48
Figura 3.12: Panel de XAMPP.....	49
Figura 3.13: Prueba de XAMPP.....	49
Figura 3.14: Archivo Makefile.....	51
Figura 3.15: Problema de carga de programa.....	51
Figura 3.16: Solución de carga de programa.....	52
Figura 3.17: Prueba de Arduino funcionando como servidor web.....	52
Figura 3.18: Acceso a MySQL.....	53
Figura 3.19: Muestra de bases de datos.....	53
Figura 3.20: Creación de tabla.....	54
Figura 3.21: Acceso a phpMyAdmin.....	54
Figura 3.22: Creación de base de datos con phpMyAdmin.....	55
Figura 3.23: Creación de tabla desde phpMyAdmin.....	55
Figura 3.24: Archivo de prueba de conexión con la base de datos.....	56
Figura 3.25: Funcionamiento correcto de archivo de prueba de base de datos.....	56
Figura 3.26: Archivo para conectar a la base de datos.....	57
Figura 3.27: Archivo para cerrar la conexión con la base de datos.....	57
Figura 3.28: Archivo de prueba para manipular la base de datos.....	58
Figura 3.29: Archivo que se cargara al Arduino.....	59
Figura 3.30: Archivo para recuperar valores del Arduino.....	59
Figura 3.31: Archivo que almacena en la base de datos.....	59

Figura 3.32: Archivo que muestra el contenido de la base de datos.	60
Figura 3.33: Visualizar contenido de la base de datos.	60
Figura 3.34: Archivo que genera el archivo ".csv".....	60
Figura 3.35: Muestra de la aplicación vista desde el navegador.	61
Figura 3.36: Simbología del esquema de conexiones.	62
Figura 3.37: Esquema de conexiones.....	63
Figura 3.38: Sistema conectado.	64
Figura 4.1: Esquema del sistema.....	65
Figura 4.2: Cajas en donde se colocaran los SAD.	66
Figura 4.3: Material para el diseño de la tarjeta.....	66
Figura 4.4: Sensor de gas.	67
Figura 4.5: Tarjetas diseñadas para la captura de datos.	67
Figura 4.6: Caja con Arduino, Raspberry-Pi, tarjeta de adquisición y <i>switch</i>	67
Figura 4.7: Caja con Arduino y tarjeta.	68
Figura 4.8: Creación de base de datos para el proyecto.	68
Figura 4.9: Muestra de las bases de datos.	69
Figura 4.10: Creación de la nueva base de datos.	69
Figura 4.11: Creación de la tabla.	69
Figura 4.12: Descripción de la tabla.....	70
Figura 4.13: Archivo que contiene las direcciones <i>IP</i> de las cajas.	70
Figura 4.14: Selección de base de datos en archivo <i>conectaDB.php</i>	70
Figura 4.15: Archivo <i>getValoresAduino.php</i> modificado.....	71
Figura 4.16: Archivo <i>recupera.php</i>	71
Figura 4.17: Modificación al archivo " <i>index.php</i> ".	72
Figura 4.18: Modificación del archivo " <i>guardaArc.php</i> ".	72
Figura 4.19: Archivo " <i>guardaArcTotal.php</i> ".	72
Figura 4.20: Cajas (SAD).....	73
Figura 4.21: Sistema en funcionamiento.....	73
Figura 4.22: Muestra de hoja de cálculo generada a partir de la consulta a la base de datos.....	74
Figura 4.23: Ubicación de los archivos generados.....	74
Figura C.1: Transductor de posición de resistencia variable.....	96
Figura C.2: Principio del transductor de posición potenciométrico.....	96
Figura C.3: Galga extenciométrica resistiva.	97
Figura C.4: Transductores de humedad con sensor resistivo.	98
Figura C.5: Relación resistencia/temperatura para un termistor.	99
Figura C.6: Galgas extenciométricas semiconductoras.....	99
Figura C.7: Transductor de vibraciones electromagnético.....	100
Figura C.8: Tacómetro de rueda dentada.	101
Figura C.9: Circuito de termopares básico.....	102
Figura C.10: Termopila.	103
Figura C.11: Cilindro de presión dinámica.	104
Figura C.12: Cilindro de efecto simple.	105
Figura C.13: Cilindro de efecto doble.....	105
Figura C.14: Cilindro telescópico.	105

Figura C.15: Motor de engranaje.	106
Figura C.16: Funcionamiento de un relevador.	108
Figura C.17: Motor serie.	109
Figura C.18: Motor paralelo.	109

BIBLIOGRAFÍA Y REFERENCIAS

1. Aguayo, P. (2004). Introducción al microcontrolador. Recuperado el 9 de junio de 2014, de <http://www.olimex.cl/tutorial/tutorial1.pdf>
2. Castro, S. (1973). Funcionamiento y propósito de la construcción de un lenguaje de máquina para una microcomputadora. Tesis de licenciatura. Universidad Nacional Autónoma de México.
3. Copper, W. & Helfrick, A. (1991). Instrumentación Electrónica Moderna y técnicas de medición. Edo. México: Pearson Educación
4. Delgado, A (2013). Actuadores hidráulicos. Recuperado el 19 de junio de 2014, de http://www.uhu.es/rafael.sanchez/ingenieriamaquinas/carpetaapuntes.htm/Trabajos%20IM%202009-10/Antonio%20Delgado%20Diez-Actuadores%20hidraulicos_2.pdf
5. Elmasri, R. & Navathe, S., (2007). *Fundamentos de sistemas de bases de datos*. Madrid: Pearson Education
6. Gregory, B. (1984).cap.5 *Transductores. Instrumentación eléctrica y sistemas de medida*. México: Gustavo gili
7. Gregory, B. (1984).cap.7 *Interferencias y apantallamiento. Instrumentación eléctrica y sistemas de medida*. México: Gustavo gili
8. Gridling, G. & Weiss Be. (2006). Introduction to Microcontrollers. Recuperado el 11 de junio de 2014, de <http://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf>
9. Haro, H., (2002). Diseño y construcción de un sistema de adquisición de datos de propósito general en base a una microcomputadora. Tesis de maestría. Universidad Nacional Autónoma de México
10. Morón, J. (2011). Señales y Sistemas. Venezuela: Fondo editorial
11. Morris, M. (1982). Lógica Digital y diseño de computadores. Edo. De México: Prentice Hall
12. Oppenheim, A. & Willsky, A. (1998). Señales y Sistemas. Edo. De México: Prentice Hall Hispanoamericana S.A.
13. Richardson, M. & Wallace, S. (2013). Getting Started with Raspberry Pi. United States of America: O'Reilly Media

14. <http://archlinuxarm.org> consultado el 16 de junio del 2014
15. <http://arduino.cc/en> consultado el 16 de junio del 2014
16. <http://betawolf.nl/electronics/projects/2-aw-water-activity-meter>, consultado el 10 de septiembre del 2014
17. <http://blog.arduino.cc/2014/07/02/freefall-camera-an-autonomous-skydiving-robot/>, consultado el 10 de septiembre del 2014.
18. <http://blog.arduino.cc/2014/07/17/a-low-cost-robotic-hand-tutorial-mirroring-your-own-fingers/#more-9816>, consultado el 10 de septiembre del 2014.
19. <http://blog.arduino.cc/2014/07/31/how-long-can-you-store-food-for/>, consultado el 10 de septiembre del 2014
20. <http://blog.arduino.cc/2014/08/18/play-the-diy-arpeggiator-with-infrared-detectors-and-arduino-mega/>, consultado el 10 de septiembre del 2014
21. <http://blog.arduino.cc/2014/09/01/music-skills-get-a-boost-with-toot/> consultado el 10 de septiembre del 2014
22. <http://lema.rae.es/drae/?val=error> consultado el 16 de junio del 2014
23. <http://pidora.ca/> consultado el 16 de junio del 2014
24. <http://raspbian.org/> consultado el 16 de junio del 2014
25. <http://www.arduino.cc/en/Guide/Introduction> consultado el 16 de junio del 2014
26. <http://www.marcopucci.it/tutorial-arduino-12-robotic-hand/>, consultado el 10 de septiembre del 2014
27. <http://www.musiktech.org/infraHarp.html>, consultado el 10 de septiembre del 2014
28. <http://www.ni.com/data-acquisition/what-is/> consultado el 16 de junio del 2014
29. <http://www.raspberrypi.org/help/what-is-a-Raspberry-Pi/> consultado el 17 junio del 2014
30. <https://www.riscosopen.org> consultado el 16 de junio del 2014