



INSTITUTO POLITÉCNICO NACIONAL



**ESCUELA SUPERIOR DE INGENIERÍA
MECÁNICA Y ELÉCTRICA**

**“MONITOREO VÍA INTERNET DE UN
LABORATORIO BIOLÓGICO”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA**

P R E S E N T A N :

ORDAZ JIMENEZ JOSÉ RAÚL

Y

PIZANO TORIBIO EDUARDO

**ASESOR TITULAR:
ASESOR METODOLOGICO:**

**ARMANDO MANCILLA LEÓN
ROBERTO GALICIA GALICIA**

INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERIA MECANICA Y ELECTRICA
UNIDAD PROFESIONAL “ ADOLFO LOPEZ MATEOS”

T E M A D E T E S I S

QUE PARA OBTENER EL TITULO DE
POR LA OPCION DE TITULACION
DEBERA(N) DESARROLLAR

INGENIERO EN COMUNICACIONES Y ELECTRONICA
TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL
C. JOSÉ RAÚL ORDAZ JIMÉNEZ
C. EDUARDO PIZANO TORIBIO

“MONITOREO VÍA INTERNET DE UN LABORATORIO BIOLÓGICO”

MONITOREAR MEDIANTE UN SISTEMA VÍA INTERNET LAS CONDICIONES ÓPTIMAS PARA LA PRESERVACIÓN DE MICROORGANISMOS, TANTO PARA LOS PROCESOS BIOLÓGICOS COMO PARA LOS MICOORGANISMOS QUE SE ANALIZAN EN UN LABORATORIO.

- ❖ LOS MICROORGANISMOS COMO BASE DE TODO PROCESO BIOLÓGICO.
- ❖ ELEMENTOS PARA LA CAPTURA DE VARIABLES ANALÓGICAS.
- ❖ INTERFASE PARA EL PROCESAMIENTO Y TRANSMISIÓN DE CAPTURAS REALIZADAS.
- ❖ INTERFASE RECEPTORA Y DE ALMACENAMIENTO DE LAS CAPTURAS RALIZADAS.
- ❖ ACCESO VISUAL A LA INFORMACIÓN PREVIAMENTE ALMACENADA.
- ❖ RESULTADOS OBTENIDOS.

MÉXICO D. F., A 25 DE SEPTIEMBRE DE 2008.

A S E S O R E S


ING. ARMANDO MANCILLA LEÓN


M. EN C. ROBERTO GALICIA GALICIA


M. EN C. SALVADOR RICARDO MENESES GONZÁLEZ
JEFE DEL DEPARTAMENTO ACADÉMICO DE
INGENIERÍA EN COMUNICACIONES Y ELECTRONICA


JEFATURA
DE I.C.E.

Agradecimientos

Definitivamente el elaborar una Tesis nos forja como investigadores y desarrolladores profesionales. Desde el momento en que se decide el tema, hasta la forma en que se va desarrollando cada modulo, capítulo o parte de dicha Tesis, así mismo conforme el proyecto va tomando forma, se documenta y empieza a entregar sus primeros resultados, colma de satisfacción a los realizadores.

Más que una entrega total al proyecto de Tesis, es un reflejo documentado, desarrollado y aplicado, de todos aquellos conocimientos que fueron adquiridos durante toda nuestra formación profesional. Es por ello que esta parte, toma forma de un Caluroso y Entrañable Agradecimiento.

Primero, agradezco a la vida misma por permitirme el privilegio del esfuerzo, lucha, dedicación y entrega a mi formación profesional, que a lo largo de todo este tiempo me ha tendido innumerables retos, conflictos, conocimientos, fortalezas, sabiduría, tenacidad, perseverancia pero sobre todo humildad. Colmándome de satisfacción personal en cada instante. Dejando un sabor indescriptible de Satisfacción y Orgullo.

Agradezco a mi Padre “J. Raúl Ordaz Martínez” y a mi “Madre Crispina Jiménez Gonzáles” por el incondicional apoyo brindado día a día, a lo largo de todo este tiempo, del cual agradezco infinita y entrañablemente. Puesto que sin su presencia, apoyo y confianza no hubiese sido posible llevar a cabo este Trabajo.

A mi Hermana “Cinthya Ordaz Jiménez”, por su constante y determinante ayuda, así como su singular apoyo indiscutible, el cual alentaba a continuar con cualesquiera que fuese el reto o dificultad presentada en el camino. Además de claro, ser una guía y luz en el camino, puesto que el camino que se ha trazado en la vida, me ha servido como principal guía y meta.

A todos aquellos amores y pasiones que durante todo este tiempo, han dado ese respiro de alivio, ese apoyo indescriptible que solo puede darse en pareja. A todos esos amores que me fueron una cura a este mundo, que en ocasiones se mostraba pies de plomo, aburrido y fanfarrón, perfumado y socarrón. A todos aquellos amores que nunca me dejaron dar por pérdida la batalla, que nunca limitaron el crecimiento personal, que incluso lo apoyaron hasta su éxito.

A todos aquellos amigos que apoyaron durante los tiempos difíciles, durante esos momentos de tristeza, de sosiego, de impotencia e imprudencia. Aquellos amigos de los cuales, sin su presencia, apoyo, ayuda, consejos y sin sus palabras y abrazos, hubiese sido imposible terminar esta etapa de nuestras vidas. A todos aquellos que estuvieron con nosotros en nuestras caídas, en nuestros éxitos, en nuestras tragedias y sobre todo estuvieron alojados en nuestros corazones y lo permanecerán durante nuestras vidas.

A todos los profesores, que no solo nos formaron como ingenieros, sino como personas, por su dedicación, entrega, ética, pero sobre todo por sus conocimientos y técnicas, de las cuales nunca nos olvidaremos.

Raúl Ordaz

A la vida

Por darme la oportunidad de estar viviendo y compartiendo las mejores experiencias y vivencias que un ser humano pueda tener, a lado de la gente que más quiero y sobre todo, por tener el privilegio de pertenecer a una gran institución como lo es el **INSTITUTO POLITÉCNICO NACIONAL**.

A mis padres

Por traerme a este mundo y dejarme como legado el interés de superarme día a día y conquistar aquellos retos que parecen inalcanzables, por el apoyo y consejos que me han llevado hasta donde estoy y sobre todo por inspirarme, ya que ustedes son las figuras ejemplares a quienes siempre admiro.

A mi hermana

Con quien crecí y compartí momentos de alegría a lo largo de nuestra vida, compartimos consejos y nos brindamos apoyo en los momentos difíciles.

A mis amigos y compañeros de Carrera

Por los momentos que recorrimos a lo largo de toda la carrera, por la comprensión y apoyo que nos dimos y que los que pertenecemos al **INSTITUTO POLITÉCNICO NACIONAL**, sólo lo podemos encontrar en ese grupo selecto de gente que se conoce en esos años de estudio.

Eduardo Pizano Toribio

Índice de contenidos

1 Los microorganismos como base de todo proceso biológico	1
1.1 Los microorganismos	2
1.2 Factores que afectan las condiciones ambientales en un laboratorio	2
1.2.1 Factor temperatura	3
1.2.1.2 Refrigeración	3
1.2.1.3 Congelación	4
1.2.1.4 Altas temperaturas	5
1.2.2 Factor luminosidad	5
1.2.3 Factor humedad relativa	6
2 Elementos para la captura de variables analógicas	8
2.1 Tipos de sensores y características de operación	9
2.1.1 Sensor de temperatura	14
2.1.2 Sensor de luminosidad	15
2.1.3 Sensor de humedad relativa	17
3 Interfase para el procesamiento y transmisión de capturas realizadas	19
3.1 Adquisición de las señales provenientes de los sensores	21
3.2 Etapa previa de amplificación	22
3.3 Tarjeta de desarrollo del microcontrolador HSC08 de Motorola	24
3.3.1 Características de componentes integrados a la tarjeta HSC08	26
3.3.2 Manejo previo de las señales adquiridas	26
3.3.3 Conversión analógica a digital mediante la tarjeta de desarrollo	30
3.3.4 Procesamiento de señales de acuerdo al sensor	32
4 Interfase receptora y de almacenamiento de las capturas realizadas	35
4.1 Interfaz microcontrolador – computadora servidora	36
4.1.1 Comunicación directa a la computadora servidora	39
4.2 Procesamiento de los datos recibidos mediante C#.NET	40
4.2.1 Correspondencia de los datos de acuerdo al sensor	44
4.2.2 Almacenamiento de la información	45
5 Acceso visual a la información previamente almacenada	50
5.1 Lenguajes para la programación para Internet	51
5.1.1 PHP	52
5.1.2 Tratamiento de los datos en PHP	54
5.1.3 Graficación en PHP	57
5.2 Configuración del Servidor WAMP	64
5.2.1 Diseño y creación de la base de datos	74
5.3 Consulta de datos almacenados	78
5.4 Sitio Web	83
5.4 .1 Estructura del Sitio Web	85

6 Resultados	88
6.1 Alcances logrados	89
6.2 Posibles mejoras	93
6.3 Costos	94
6.4 Conclusiones y recomendaciones	96

Resumen

Indudablemente estos tiempos han sido definidos por los avances tecnológicos, que a cada momento nos asombra con sus descubrimientos, aplicaciones y sobre todo sus infinitos alcances. Es por ello que en esta época han surgido nuevas técnicas que involucran al extenso mundo de la tecnología, del cual además de ser complejo y avanzado, es indiscutiblemente competido. Siendo este el motivo que ha inspirado a los autores de esta Tesis a aportar un pequeño grano de arena, que puede llegar a ser de gran inspiración para que de este proyecto se desarrolle un proyecto robusto e innovador.

Por esta razón esta Tesis además de tener carácter de investigación y desarrollo universitario cuenta con toda la formalidad, seriedad y responsabilidad que todo proyecto innovador debe contar.

El proyecto presentado, no sólo da una propuesta de monitorear a distancia, sino que el objeto que monitorea es un objeto real, de naturaleza crítica, que demanda una solución como la que se plantea. En este proyecto, se pretende monitorear un laboratorio biológico vía Internet, en el cual se puedan monitorear de manera remota las condiciones ambientales reales del objeto en cuestión. El proyecto fundamentalmente busca monitorear las condiciones de luz, humedad y temperatura, condiciones de las cuales se exige que el monitoreo se de en tiempo real, condiciones que son vitales para la naturaleza de la biología, condiciones bajo las cuales puede comprometerse un producto, un cultivo, una vacuna, bancos de bacterias, un análisis químico-biológico, condiciones críticas que deben cuidarse en un laboratorio de esta naturaleza.

Es por ello que el proyecto que se plantea viene a resolver algunas de las cuestiones más importantes, como es el seguimiento y registro detallado de las condiciones ambientales bajo las cuales se encuentra dicho laboratorio. Esto se pretende resolver implementando una base de datos, de la cual pueda obtenerse un resumen detallado de meses, semanas, del día, así como una gráfica que represente dichas condiciones ambientales, dicha gráfica podrá reflejar las condiciones del día, o de la semana. Todo esto podrá consultarse en una página Web, a través de la cual podrá acceder cualquier persona que sea designada a gestionar, vigilar y revisar el monitoreo del laboratorio en cuestión.

De igual forma se pretende, emitir alertas tempranas para así dar aviso a los especialistas o encargados en cuestión sobre las condiciones, bajo la cuales se están presentando las mediciones monitoreadas.

Introducción

En el campo de la ingeniería el manejo e interpretación de los datos es algo de suma importancia, ya sea para un control de procesos, relaciones estadísticas, mediciones, comparación de fenómenos, pruebas, ensayos, etc. A lo largo de esta tesis se tratará de plantear el desarrollo de un proyecto para la medición de ciertos fenómenos naturales, que estarán aplicados al área de la biología.

Aunado a que en la actualidad todos los procesos y eventos e incluso aplicaciones de cualquier tipo, o bien actividades como el comercio, el intercambio de información o cualquier tipo de interacción que suele llevarse a través de contactos de manera directa, con un individuo o un grupo relacionado a éstos, paulatinamente tienden a llevarse a través de Internet, esto es debido a que las facilidades con las que cuenta, ya que permite una rápida y eficaz comunicación entre dos lugares, sean cuales sean o bien que se encuentren a miles de kilómetros de distancia, ahora es posible acortar distancias gracias al Internet. Ésta es la tendencia que están presentado la mayoría de las actividades realizadas de manera cotidiana.

Es por ello que se presenta en esta tesis la forma de implementar un sistema de monitoreo que a través del Internet, se pueda dar seguimiento al estado actual bajo el cual se encuentra algún lugar, en este caso se planteará la implementación de un sistema de monitoreo para un laboratorio biológico, donde en este tipo de laboratorios es de suma importancia mantener y verificar esas condiciones ambientales, bajo las cuales se encuentran los posibles cultivos, bacterias, microorganismos, muestras, etc. que es necesario se encuentren bajo ciertos rangos de temperatura, humedad, presión para que puedan preservarse de manera correcta.

Objetivos

Monitorear mediante un sistema vía Internet las condiciones ideales u óptimas para la preservación de microorganismos, ya sea para procesos biológicos ó cultivos de bacterias de un laboratorio biológico.

Debido a la gran necesidad de monitorear el mundo laboral, industrial, empresarial, o simple vida cotidiana, resulta necesario implementar sistemas de monitoreo computacional vía Internet que permitan obtener una optimización en las herramientas empleadas para el sistema de monitoreo, con el fin de simplificar tareas, reducir costos y mejorar el rendimiento y desempeño.

Antecedentes

Durante el proceso de investigación realizado para conocer más acerca de sistemas similares al propuesto, se arrojaron como resultados algunos sistemas que trabajan de manera remota (a través de una red o bien por Internet), donde en la mayoría de los casos hacen uso de un PIC (Programmable Interrupt Controller), para implementar el sistema de monitoreo aplicable a diferentes áreas, pero en general se mantiene la esencia de lo que es el monitoreo de algún dispositivo o medio.

Mencionando algunos de los casos se tiene el de un simulador de procesos remotos a que través de Internet y con ayuda del software MATLAB se generan resultados de manera virtual, fenómenos relacionados con la física, que como se sabe, gracias al estudio de este campo dichos fenómenos están estrechamente relacionados con modelos matemáticos, siendo posible reproducirlos en un entorno gráfico gracias a MATLAB, que permite trabajar de manera práctica dichos modelos matemáticos. Otro sistema similar es el que se implementa con LABNET, que permite monitorear procesos de manera remota y realizar simulaciones. Dicho monitoreo se realiza vía Internet, generando una conexión con un servidor que permite realizar el enlace con las maquetas que

Introducción

son una especie de montaje similar al que puede encontrarse en la industria. Una vez establecido el enlace, se comienza con los procesos de monitoreo sobre esas maquetas para hacer prácticas de simulación sobre el tipo de maqueta empleada.

Otros sistemas implementados son muy similares al que se plantea para esta tesis, ya que en ellos se trabaja con sistemas de medición, para el caso de telecontrol de una planta pasteurizadora, donde el control y manejo de todos los procesos es de suma importancia, ya que se trata de un producto alimenticio perecedero, en el cual los costos por pérdidas pueden ser bastante altos, por lo que debe existir un alto grado de monitoreo y control sobre dicho proceso.

El otro sistema similar es el de monitoreo de temperatura de un estanque de agua, el cual se ubica entre la ciudad de Chile y España, usando redes de alta velocidad como lo es el acceso a Internet por banda ancha. La propuesta que se presenta en esta tesis es poder monitorear desde cualquier parte del mundo el laboratorio en donde se implementará dicho sistema.

Aportaciones

Con este proyecto se pretende impulsar el desarrollo de lugares y centros especializados en la conservación de productos orgánicos o de carácter biológico, que específicamente están orientados a la conservación de muestras, cultivos, procesos biológicos, etc., en donde el objetivo principal es el de monitorear las condiciones bajo las cuales se encuentran los contenedores, bancos de cultivos e incubadoras, que preservan dichas muestras, así como los procesos biológicos que se desarrollen en las mismas. Además los temas desarrollados para la investigación abrirán la posibilidad de utilizar este tipo de monitoreo en otras áreas, ya que hoy en día las aplicaciones y operaciones de varios tipos tienden a desarrollarse a través de Internet, por lo que para cualquier área que requiera un sistema de monitoreo

vía Internet, es posible implementarlo, desde el diseño de los sistemas hasta la creación del sitio Web para acceder a los servicios deseados.

Problema específico a resolver

Se pretende resolver la situación presentada en casos donde resulta de vital importancia estar al tanto de las condiciones de algún lugar y no se cuenta con la posibilidad de conocer dichas condiciones, con este proyecto se dará esta facilidad, ya que también permitirá a algún especialista, que por causas de diversa índole no se encuentra cerca del lugar donde se genera la información, él podrá acceder a ella en cualquier parte del mundo a través de la página Web, con lo que estará siempre al tanto e informado de cómo se va desarrollando esa información, que en este caso, se refiere específicamente a las condiciones ambientales bajo las cuales se desarrolla el proyecto.

Es importante recalcar que el objetivo específico del proyecto es monitorear las condiciones ambientales propicias para la supervivencia y preservación de microorganismos, así como procesos biológicos basados en microorganismos, todo ello dentro de un laboratorio. Es por ello que para este proyecto únicamente se propone el monitoreo de dos variables, dichas variables son la temperatura y el porcentaje de luminosidad. Esto con la posibilidad de un alcance hacia 4 variables que definen las condiciones ambientales de un laboratorio, las cuales son temperatura, porcentaje de luminosidad, humedad relativa y presión atmosférica.

Capítulo 1

*Los microorganismos
como base de todo proceso biológico*

1.1 Los microorganismos

Para entender lo que es un proceso biológico resulta necesario entender antes lo que es un microorganismo.

Un microorganismo, también llamado microbio u organismo microscópico son formas de vida muy pequeñas, que sólo algunos pueden ser observados a través del microscopio. En este grupo están incluidas las bacterias, algunos virus, los mohos y las levaduras. Algunos microorganismos pueden causar el deterioro de algunos procesos biológicos, entre los cuales se encuentran los microorganismos patógenos, que a su vez pueden ocasionar enfermedades debido al consumo de alimentos contaminados por dichos microorganismos. Adicionalmente, existen ciertos microorganismos patógenos que no causan un deterioro visible en el proceso biológico. Sin embargo, existen también algunos microorganismos que son benéficos y que pueden ser usados en el procesamiento de los alimentos, con la finalidad de prolongar su tiempo de vida o de cambiar las propiedades de los mismos, así como en la elaboración de vacunas o desarrollo de nuevos medicamentos.

Otra característica es que a diferencia de las plantas y los animales, son una organización biológica elemental, en su mayoría son unicelulares, aunque en algunos casos se trate de organismos compuestos por células multinucleadas, o incluso multicelulares.

1.2 Factores que afectan las condiciones ambientales en un laboratorio.

Los factores que afectan a la supervivencia de los microorganismos tanto en procesos biológicos, como en su conservación son:

- Temperatura
- Refrigeración
- Congelación
- Altas temperaturas

- Luminosidad y
- Humedad Relativa.

1.2.1 Factor temperatura

De acuerdo a su comportamiento frente a la temperatura, los organismos pueden ser termófilos, mesófilos y psicrotrofos.

Los organismos termófilos, son organismos vivos que pueden soportar condiciones extremas de temperatura relativamente altas (superior a los 45 °C) y relativamente bajas (por debajo de los 0 °C).

Por su parte, los organismos mesófilos, son organismos que necesitan una temperatura óptima de crecimiento comprendida entre 30°C y 45°C. La temperatura mínima se encuentra en el rango de 15°C a 20°C y la temperatura máxima en torno a 45°C. La gran mayoría de los microorganismos son mesófilos, incluidos los patógenos. Así mismo los organismos psicrotrofos, son aquellos organismos que son resistentes y pueden crecer a temperaturas muy bajas. La tabla 1.1 muestra la relación de temperaturas para cada tipo de microorganismo.

Tipo de microorganismo	Temp. mínima °C	Temp. óptima °C	Temp. máxima °C
Psicrófilo	-5 +5	12 – 15	15 – 20
Psicrótrofo	-5 +5	25 – 30	30 – 35
Mesófilo	5 – 15	30 – 45	35 – 47
Termófilo	40 – 45	55 – 75	60 – 90

Tabla 1.1 Relación de Temperaturas

1.2.1.1 Refrigeración

A temperaturas inferiores a la óptima, la velocidad de crecimiento de los microorganismos disminuye y los periodos de retardo de crecimiento se alargan mucho.

A una temperatura de refrigeración (0 °C - 5° C) los organismos psicrófilos crecen más rápidamente que los mesófilos. Cuando se enfría rápidamente el objeto de control de un proceso biológico, muchas de las bacterias mesófilas que normalmente resistirían la temperatura de refrigeración, mueren como consecuencia del “choque de frío”.

A baja temperatura las rutas metabólicas de los microorganismos se ven alteradas, como consecuencia de su adaptación al frío. Estos cambios metabólicos pueden dar lugar a que se produzcan deterioros diferentes, causados por los mismos microorganismos a diferentes temperaturas.

El deterioro en procesos biológicos refrigerados se produce por microorganismos psicrófilos, ya que sus velocidades de crecimiento son lentas, los periodos de almacenamiento son muy prolongados.

Los microorganismos patógenos son en su mayoría mesófilos y no muestran crecimiento apreciable ni formación de toxinas a temperaturas de refrigeración correctas. Ahora bien, si la temperatura no es monitoreada rigurosamente puede producirse un desarrollo muy peligroso rápidamente.

1.2.1.2 Congelación

La congelación detiene el crecimiento de todos los microorganismos. Los organismos superiores como los hongos, levaduras y helmintos son más sensibles que las bacterias y mueren. A temperaturas más bajas (-30° C) la supervivencia de las bacterias es mayor que en temperaturas de congelación

más altas (-2 a -10° C), sin embargo, estas temperaturas también deterioran el proceso biológico más que las más bajas.

La congelación puede producir lesiones importantes en los microorganismos contaminantes de un proceso biológico. Este aspecto hay que considerarlo al hacer control microbiológico. Durante la congelación, la carga microbiana continúa disminuyendo. Sin embargo, las actividades enzimáticas de las bacterias pueden continuar dando lugar a más deterioro.

Tras la congelación, los microorganismos supervivientes pueden desarrollarse en un ambiente en el que la rotura de la integridad estructural del proceso biológico, como consecuencia de la congelación puede producir un ambiente favorable para el deterioro microbiano.

1.2.1.3 Altas temperaturas

Las temperaturas superiores a las de crecimiento óptimo producen inevitablemente la muerte del microorganismo o le producen lesiones importantes. Las células lesionadas pueden permanecer viables, pero son incapaces de multiplicarse hasta que la lesión haya sido reparada.

Aunque se han observado excepciones, está perfectamente establecido que la cinética de termodestrucción bacteriana es logarítmica.

La velocidad de termodestrucción se ve afectada por factores intrínsecos (diferencia de resistencia entre esporas y células vegetativas), factores ambientales que influyen el crecimiento de los microorganismos (edad, temperatura, medio de cultivo) y factores ambientales que actúan durante el tratamiento térmico (pH, tipo microorganismo, sales, etc.).

1.2.2 Factor luminosidad

La luminosidad (intensidad de luz) produce una disminución exponencial en el número de células vegetativas o de esporas vivas con el tiempo de irradiación lumínica. Por tanto se pueden calcular los valores para la irradiación lumínica.

El limitante o inconveniente de este factor, es el modo de cuantificar la intensidad lumínica, esto debido a la ubicación o distribución de la infraestructura del laboratorio biológico, puesto las variaciones de luminosidad variarán de acuerdo a su ubicación, es por ello que como norma general en los laboratorios biológicos, se controla la intensidad lumínica en porcentaje. Dicho porcentaje contemplará su 100% a la máxima intensidad lumínica que pueda generarse o a la que pueda someterse el microorganismo en el laboratorio biológico, así mismo se contempla un porcentaje nulo a la total ausencia de luminosidad, es decir, al aislamiento lumínico aplicado a microorganismos en el laboratorio biológico.

Existe una falta de información precisa sobre la susceptibilidad de las diferentes especies microbianas a la radiación lumínica, diferentes cepas de una misma especie pueden tener una resistencia distinta.

El mayor valor del tratamiento con radiación lumínica se encuentra en el saneamiento del aire, aunque también pueden aplicarse para esterilizar superficies de algunos compuestos o sustancias utilizadas en procesos biológicos o para el equipo manipulador de los procesos biológicos.

1.2.3 Factor humedad relativa

Es importante entender que los valores de humedad relativa se miden en porcentaje de presencia de agua, es decir en porcentajes relacionados donde el 1.0 representa el 100% y así respectivamente según el porcentaje que represente cada valor de humedad.

Los microorganismos requieren la presencia de agua en una forma disponible, para que puedan crecer y llevar a cabo sus funciones metabólicas. La mejor forma de medir la disponibilidad de agua, es mediante la actividad de agua (humedad). La humedad en un proceso físico que puede reducirse aumentando la concentración de solutos, mediante la extracción del agua o mediante la adición de solutos.

La deshidratación es un método de conservación de algunas bacterias, basado en la reducción de la humedad, durante el proceso se necesitan catalizadores y solutos, que al ser añadidos, descienden la humedad.

Un pequeño descenso de la humedad es a menudo suficiente para evitar la alteración del proceso, siempre que esta reducción vaya acompañada por otros factores antimicrobianos. La mayoría de las bacterias y hongos crecen bien a valores de humedad entre 0.98 % y 0.995 %, a valores de humedad más bajos la velocidad de crecimiento y la masa celular disminuyen, a la vez que la duración de la fase de retardo de crecimiento aumenta hasta llegar al infinito (cesa el crecimiento).

Algunos tipos de microorganismos son capaces de crecer en condiciones de alto contenido de sal (baja humedad relativa). De acuerdo a la capacidad que estos microorganismos poseen para sobrevivir a un bajo nivel bajo de humedad se denominan osmófilos, xerófilos y halófilos (según va aumentando su requerimiento de sal).

Un bajo valor de humedad relativa, reduce también la tasa de mortalidad de las bacterias, es decir, un bajo valor de humedad relativa también protege los microorganismos durante los tratamientos térmicos que requiera el proceso biológico.

Capítulo 2

*Elementos para la captura
de variables analógicas*

Introducción

Debido a la finalidad de este proyecto, se requiere el uso de sensores para así obtener una medición de los fenómenos que ocurran en los procesos biológicos dentro del laboratorio biológico a monitorear, a modo de poder percibir los sucesos lo más parecido al fenómeno tangible, así como una aproximación que sea lo más cercana al tiempo real.

Es importante recalcar que al mencionar tiempo real, se refiere al punto de vista de “ingeniería del software”, desde dicho punto de vista, un sistema funciona en tiempo real cuando procesa la información en una frontera de tiempo definida por el proceso que se monitorea. El objetivo principal es que el sistema no presente fallas y se ajuste a los tiempos que define el diseñador, lo cual no implica necesariamente que sea a máxima velocidad.

Para el caso de este proyecto, al manejar señales de naturaleza discreta, resulta importante aclarar que la frontera de tiempo mencionada, está basada tanto en la cantidad de información que se desea almacenar, como en el intervalo posible de tiempo, en el cual, las variaciones del ambiente del laboratorio biológico puedan variar drásticamente, esto es, en un intervalo definido por los mismos laboratorios biológicos, dicho intervalo se encuentra entre los 3 y 120 segundos según el proceso biológico, para lo cual se ha decidido que el intervalo se realice cada minuto, esto por fines de almacenamiento, los cuales se describirán en capítulos posteriores. Esto no significa que para ciertos procesos biológicos se modifique dicho intervalo.

2.1 Tipos de sensores y características de operación.

Los Sensores

Los sensores son dispositivos capaces de medir manifestaciones o cualidades de fenómenos físicos. Estos dispositivos aprovechan sus propiedades para adaptar esas variaciones que mide de dicho fenómeno.

Un sensor puede ser considerado como un transductor, ya que es capaz de convertir un tipo de energía a otro como puede ser convertir energía calorífica en energía eléctrica, o bien energía lumínica en energía eléctrica.

Por ejemplo, un termómetro clínico es un transductor que convierte la temperatura corporal en una medida de longitud (la progresión del mercurio a través de una regla graduada). Actualmente, la mayoría de transductores son eléctricos y por lo general se aplica la palabra "sensores" a todo tipo de transductor eléctrico.

Los transductores eléctricos (sensores) pueden ser: analógicos y digitales, pasivos y activos. Dentro de la clasificación de los pasivos, éstos pueden ser resistivos, capacitivos, inductivos y especiales.

Características de los sensores

Los sensores no son perfectos, ya que presentan algunas características que afectan a la calidad de la medición. Para la mayoría de medidas, se tienen 2 grupos principales de características: estáticas y dinámicas. Las características estáticas son aquellas que son consecuencia directa de la medida y no de su variabilidad. Las características dinámicas son perturbaciones de la medida, como consecuencia de que ésta varía continuamente de valor, creando artefactos adicionales en los sensores.

Características Estáticas de un sensor:

Se dice que un sistema funciona en régimen estático cuando la señal no varía o varía con una frecuencia inferior a 1 Hz o 2 Hz. En estos casos, las características dinámicas del sensor pueden obviarse y se toman en cuenta las características estáticas. Las características estáticas del sensor consisten básicamente en las transformaciones que realiza el sensor para pasar de la magnitud física al voltaje eléctrico.

El sensor convierte una magnitud física m en un voltaje q de la forma:

$$q = f(m)$$

Si se representa la ecuación anterior se obtiene la curva de calibración estática de la figura 2.1:

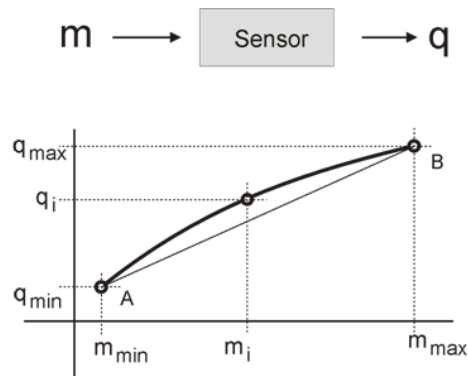


Figura 2 .1 Curva de calibración estática

Esta gráfica es importantísima para entender lo que viene a continuación, ya que para determinar la forma de la gráfica se definen los siguientes parámetros de la curva de calibración.

Parámetros de la curva de calibración estática

Rango de medida: Intervalo en el que puede emplearse un sensor. Rango de Entrada si se indica en términos de " m " y Rango de Salida si lo indicamos en términos de " q ".

Escala total o Span: Es el ancho del intervalo donde puede operar el sensor ($m_{\max}-m_{\min}$ o $q_{\max}-q_{\min}$).

Sensibilidad Estática o Factor de Escala o Ganancia: está definido por la pendiente entre la salida y la entrada.

Si el sensor es lineal, entonces $S = \text{constante}$. Su inversa es el factor de deflexión.

Factores de Calidad Estática

Los factores de calidad determinarán en mayor o menor medida la exactitud de las mediciones, y por tanto, la calidad de las mismas. Ciertos factores de

calidad pueden ser corregidos a posteriori, pero otros no. A veces, la corrección de un factor de calidad va en perjuicio de otros factores.

Existen muchísimos factores de calidad, siendo los más sencillos.

Factor Linealidad: Recordando que S (Factor de Escala) está definido por la fórmula 2.1.

Factor de Escala (S): Representa la primera derivada de la salida con respecto a la entrada. Dicho valor debería ser constante lo que produciría una línea recta

$$S = \frac{\Delta q}{\Delta m} \quad \dots \text{Fórmula 2.1}$$

Donde Z = deriva del cero (zero offset).

La mayoría de sensores pueden anular Z, pero corregir la linealidad no ha sido históricamente fácil.

Se llama linealidad a la máxima desviación entre la curva de calibración y la recta ideal.

Resolución: La resolución es el incremento mínimo necesario para obtener un cambio en la salida.

- Umbral: Es un caso particular de resolución cuando se parte de cero.
- La resolución y el umbral suelen estar comprometidos en el conversor analógico a digital, no propiamente en el sensor.

Ambos factores de calidad suelen ser los primeros que se pasan por alto al realizar medidas, ya que no suelen tenerse en cuenta. En una balanza electrónica, por ejemplo, el umbral es el valor de peso mínimo que el sistema admite. La resolución (figura 2.2) es la mínima cifra decimal que el sistema dice proporcionar, aunque esto no implica que sea con seguridad el dato real y exacto, pues además de la resolución puede haber más incertidumbres en dicha medida, por ejemplo, ruido (figura 2.3).

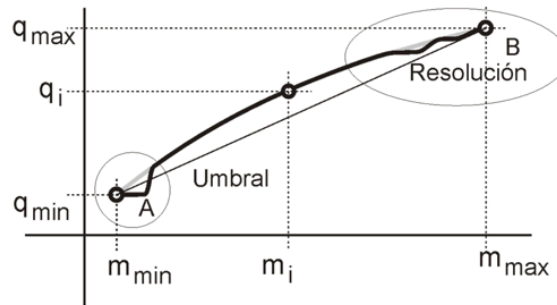


Figura 2.2 Gráfica resolución-umbral

La resolución y el umbral suelen mejorar con la sensibilidad del sensor. Cuando el sensor tiene problemas de resolución y umbral aparece un fenómeno denominado "juego". El "juego" representa una inercia o una holgura que evita cambios pequeños en el sensor, haciendo que los cambios se presenten "a saltos".

El juego suele presentarse en el umbral sin mayores consecuencias. Cuando el juego se presenta en toda la medición, suele indicar un problema grave de medida llamado "histéresis", ya que la holgura del sensor tergiversa la medida en función del sentido de variación de la misma.

Factor Ruido: El ruido es la inconsistencia continua de la medida a lo largo del tiempo. El ruido se presenta como una variación sistemática de un mismo valor medido. La amplitud del ruido implica una indeterminación de la magnitud que estamos midiendo. Dicha indeterminación no suele ser corregible a posteriori y debe ser conocida y tomada en consideración.

El aumento del muestreo, o incremento de intervalo de muestreo de la señal permite determinar el porcentaje de ruido presente y su perturbación sobre la medida.

La ausencia de sobre muestreo incorporará todo el ruido a la medida, ya que no será posible conseguir una estimación del valor medido en base a la media y varianza de las muestras repetitivas.

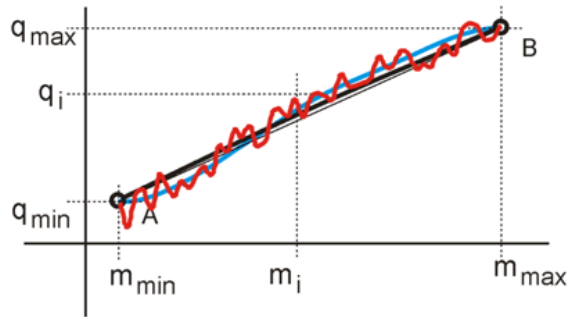


Figura 2.3 Relación del ruido en el sensor

2.1.1 Sensor de temperatura

Se ha establecido en base a los rangos de temperatura posibles para el monitoreo del laboratorio biológico que las temperaturas a medir fluctuarán entre 5 °C y 50 °C, debido a que el ambiente del microorganismo a monitorear que soporta temperaturas más bajas, tiene como valor de temperatura más bajo de 5 °C, es por ello que se ha delimitado el proyecto a la operación con valores positivos de temperatura y a su vez a la correspondencia de valores positivos de voltaje.

Lo anterior implica que se usará el sensor de precisión de temperatura lm35 (figura 2.4), dicho sensor es un sensor de propósito general, económico, resistente y confiable. La única desventaja que presenta es la del defasamiento en tiempo de la señal de respuesta del sensor, es decir, existe un retardo de alrededor de 3 a 5 segundos en dar el valor de temperatura sensada al instante al que existió una variación de la temperatura, esto es, que el valor de voltaje que entrega a la salida, es el valor de temperatura que capturó el sensor en un instante anterior a 3 segundos aproximadamente.

Las características fundamentales de dicho sensor son las siguientes:

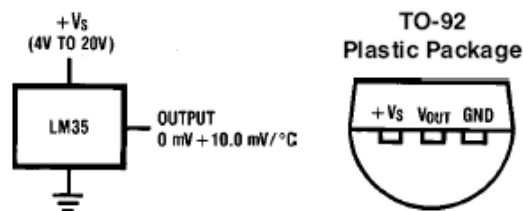


Figura 2.4 Sensor de temperatura

- Calibrado a grados Celsius.
- Escala lineal de 10 mV/°C.
- Rango completo de operación de -55 a 150 oC
- Recomendado para aplicaciones remotas
- Operación de 4 a 30 volts
- Corriente de drenaje menor a 60 μ Amper
- Baja impedancia de salida, 0.1 Ohms con 1 mA de carga

2.1.2 Sensor de luminosidad

Se decidió usar, como sensor de luz una foto celda, esto por la factibilidad de detección de intensidad de luz, aunque, lo correcto debería ser el uso de un foto resistor, el cual varía su resistividad de acuerdo a la intensidad de luz, la inconveniencia que implica el uso de un foto resistor, es la necesidad de implementar, un circuito el cual entregue variaciones de voltaje, es decir, se tendría que acoplar algún arreglo de divisores de voltaje, a modo de que nos entregase un voltaje, el cual oscile entre 0 y 5 volts.

Es por ello, que por motivos prácticos se ha optado por el uso de una foto celda, de modo que se ha de entender, la disponibilidad de habilitar en un futuro la existencia de un sensor de luz dedicado a la tarea en particular, o de algún otro foto sensor de precisión. A continuación se describe como funciona y qué es una fotocelda.

Las Fotoceldas

Las fotoceldas son pequeños dispositivos que producen una variación eléctrica en respuesta a un cambio en la intensidad de la luz. Las fotoceldas pueden clasificarse como fotorválticas o fotoconductoras.

Una celda fotorváltica es una fuente de energía cuyo voltaje de salida varía en relación con la intensidad de la luz que incide en su superficie. Una celda

fotoconduktiva ya es un dispositivo pasivo, incapaz de producir energía. Su resistencia varía en relación con la intensidad de la luz en su superficie.

Industrialmente, las aplicaciones de las fotoceldas caen en dos categorías generales:

1.-Detección de la presencia de un objeto opaco.

a) La detección puede hacerse en una base de todo o nada, en la que el circuito de la fotocelda, tiene sólo dos estados de salida que representan la presencia o la ausencia de un objeto. Este es el tipo de detección suele ser usada para contar las partes que viajan por una banda transportadora, o para evitar la operación de un mecanismo si las manos del operador no están fuera de la luz de trabajo.

b) La detección puede hacerse en una base continua, teniendo en el circuito de la fotocelda una salida continuamente variable, que representa la posición variable del objeto. Éste es el tipo de detección usada para observar la orilla de una tira de material en movimiento, para evitar que se desvíe demasiado de su posición adecuada.

La ventaja principal de las fotoceldas sobre otros dispositivos de detección, es que no se requiere contacto físico con el objeto de detección.

2.- Detección del grado de luminosidad

Detección del grado de luminosidad (capacidad de captar intensidad de luz) o el grado de luminiscencia (capacidad de generar luz) de un líquido o un sólido. En estas aplicaciones, el proceso ha sido dispuesto de manera que la translucidez o luminiscencia representen una variable de proceso importante. Algunos ejemplos de variables que pueden ser medidas de esta manera son densidad, temperatura y concentración de algún compuesto químico específico.

La fotocelda que se muestra en la figura 2.5 es la que se usará para este proyecto, se describe a continuación.

Módulo de celda solar 1V 400 mA, de Steren numero de parte PS-815

- Módulo de celdas solares 1 volt
- Corriente de 400 mA CD
- Medidas 9.5cm x 6.5cm
- Comportamiento lineal como respuesta a la salida.



Figura 2.5 Foto celda usada como sensor de luz

2.1.3 Sensor de humedad relativa

En este proyecto, se propone un sensor de humedad relativa, siendo importante recalcar que no se implementará dicho sensor debido a su costo, ya que a diferencia de los sensores anteriores, este sensor representa un alto costo, dichos costos pueden observarse en el Capítulo 6.3.

El sensor que se propone es el sensor de humedad con número de parte HC5V5 de la compañía "SCMSTORE" (figura 2.6) conocido también como transductor de humedad relativa, por entregar en respuesta a su salida una diferencia de potencial, es decir, un voltaje, a diferencia de otros tipos de sensores de humedad que son resistivos o capacitivos. El sensor que se propone es usado en sistemas de control, en sistemas de aire acondicionado, en humectantes, secadores, automotores y algunos electrodomésticos, por lo cual, se entiende que es un sensor de propósito general. Como características

principales de este sensor están su salida de tensión proporcional, puesto que tiene una respuesta lineal, cuenta con alta precisión y estabilidad.

Características

- rango de operación de 0 a 100 % HR.
- voltaje de alimentación de 5 volts.
- salida de 0.8 a 3.9 volts a 25 °C.
- corriente típica de 0.38 mA y corriente máxima de 0.50 mA aprox.
- temperatura de trabajo de -40°C a +80°C.
- tiempo de respuesta de 15 seg.
- estabilidad de 1\% RH por año.

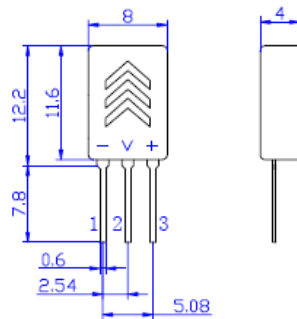


Figura 2.6 Sensor de humedad relativa.

Capítulo 3

*Interfase para el procesamiento y
transmisión de las capturas realizadas*

Introducción

Debido a la finalidad del proyecto ya antes mencionada, no sólo se necesitará obtener señales de los sensores, convertirlas de analógico a digital, preprocesarlas y enviarlas a la computadora para su procesamiento y almacenamiento final, sino que también resulta importante habilitar la existencia de una mejora a futuro, es decir, las opciones de expansibilidad de este proyecto no sólo son óptimas a simple vista, sino que además, son suficientes para futuras expansiones.

Es por ello que se ha decidido usar un microcontrolador potencialmente estable, con una capacidad de procesamiento para tareas de propósito general, como señales analógicas simples, es decir, simples conversiones analógicas digitales, donde la velocidad de muestreo y/o captura, no sea un factor de alto riesgo o extrema importancia para el proyecto, ya que con intervalos de muestreo del orden de milisegundos, es posible tener una aproximación lo más real posible al fenómeno que acontece en el laboratorio biológico.

Así mismo, resulta basto y suficiente contar con una resolución máxima de 10 bits, esto significa tener 1024 valores posibles en un rango de voltaje de operación de 0 a 3.3 volts, es más que suficiente, pues esto representa variaciones de hasta 3.22 mV. Como se ha explicado, esto resulta sobrado, debido a que con una resolución de 8 bits se obtienen 255 valores posibles y suficientes para representar valores de temperatura y luminosidad, dicha resolución de 8 bits será la que se implemente en este proyecto. Debido a que con dicha resolución se pueden detectar variaciones de 12.94 mV.

Se pretende que a futuro, exista la posibilidad de monitorear no sólo luz y temperatura, sino que también, presión, humedad, movimiento y señal de video de lo que acontece en el laboratorio biológico (donde la señal de video se manejará de manera independiente del microcontrolador).

3.1 Adquisición de las señales provenientes de los sensores.

La adquisición de señales previamente amplificadas se refiere a la necesidad de una etapa previa, que aumente la ganancia del valor de voltaje que entrega la salida del sensor mismo, los sensores que se proponen en este proyecto tienen por respuesta en la salida, variaciones de voltaje en el orden de los milivolts. Esto significa que el sistema que capture muestras, debe tener una resolución de muestreo del orden de milivolts.

La tarjeta de desarrollo HCS08 tiene la opción de muestrear con resolución de 8 y 10 bits, pero antes de entrar en detalles de resoluciones, conoceremos la etapa previa de preamplificación. Con esto, se entenderá fácilmente la importancia de la resolución. Ésto implica entender la respuesta a la salida de los sensores que se usan descritos a detalle en el capítulo anterior.

Retomando la cuestión de la etapa de preamplificación, conlleva a lo siguiente. Como ya se ha explicado, el sensor de temperatura lm35, entrega por respuesta variaciones de 10 mV por grado centígrado. Es por ello que se decidió usar una etapa previa de amplificación que a continuación se detalla.

Para el caso del sensor de temperatura supongamos una temperatura ambiente es de 20 °C, el sensor entregará a la salida 20 mV (idealmente), por lo cual las variaciones que se necesitan en este proyecto, son de 5 °C a 50 °C, esto significa que si se amplifica mediante amplificadores operacionales de propósito general con una ganancia de 5, entonces, ahora el circuito nos entregará 1 volt aproximadamente para representar 20 °C, esto significará que si la temperatura ambiente se incrementa en un volt entonces, como respuesta del circuito preamplificador tendremos, 1050 mV aprox, esto ya es suficiente para que el ADC (convertidor analógico a digital) de la tarjeta de desarrollo pueda distinguir cada variación de un grado. Es importante recalcar que se optó por un factor de ganancia de 5, debido a que los rangos mínimo y máximo

para el ADC del microcontrolador son de 0 volts y 3.3 volts respectivamente, por lo que para 22 °C se tiene a la salida del amplificador 1.1 volts aproximadamente. Con esto los rangos propuestos para este proyecto que oscilan de 5 °C a 55 °C son suficientes, al contar con 3.3 volts máximo de operación para el ADC.

Esto significa que este proyecto, está delimitado por diseño, a sensar temperaturas que van desde 0 °C hasta 65 °C aproximadamente. Con esto se obtendrá el voltaje deseado y así la facilidad de usar un nivel de resolución de 8 bits en el ADC de la tarjeta de desarrollo HCS08, con ello, si los valores oscilan entre 0 volts y 3.3 volts, entonces se encuentra dentro de los valores de operación del ADC de la tarjeta de desarrollo.

3.2 Etapa previa de amplificación

Antes de explicar la etapa previa de amplificación, es importante entender cómo debe estar estructurado un sistema de captura de datos para ser un sistema de tiempo real óptimo y estable.

Sistema de captura de datos

Para que un sistema de captura de datos sea óptimo, debe pasar por una serie de etapas, algunas de ellas pueden omitirse debido a que no son estrictamente necesarias, pero sí deben de implementarse algunas, como se muestra en la figura 3.1.



Figura 3.1 Sistema de captura de datos

Etapa de Amplificación

Es importante recalcar que para este proyecto, sólo se usarán dos etapas, la de amplificación de bajo ruido junto con la de recortador y sujetador de nivel, debido a que los amplificadores operacionales que se emplearon, son de

propósito general y presentan un comportamiento con poco ruido, al menos para la aplicación que se les da, es por ello que la etapa de filtrado se omitió debido a que se cuenta con una señal pura de CD, la cual, se sujetará de nivel (figura 3.2) mediante un seguidor de voltaje, esto para evitar caídas de voltaje al conectarse a los pines del ADC del micro HCS08.

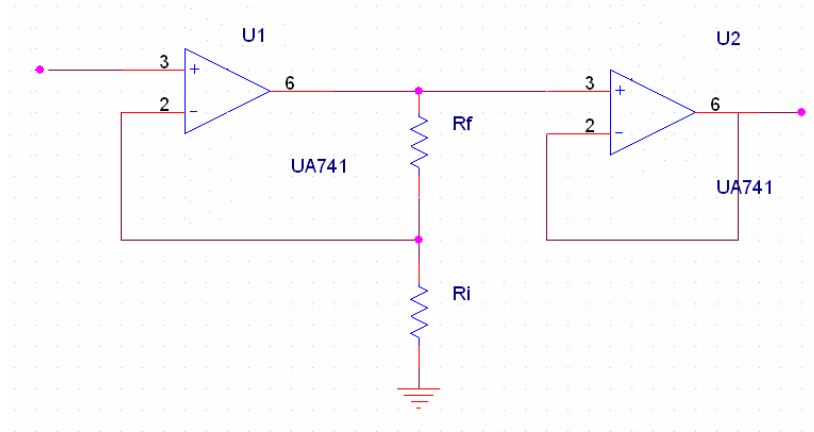


Figura 3.2 Etapa de amplificación

A continuación detallamos dicha etapa previa de amplificación.

La configuración del amplificador operacional es de amplificador inversor, por lo cual la ganancia está en función de (ver fórmula 3.1):

$$G_v = -\frac{R_2}{R_1} = \frac{V_o}{V_i} \quad \dots \text{Fórmula 3.1}$$

Suponiendo que seguimos con el ejemplo de los 20 °C, entonces el voltaje de entrada es 20 mV y se desea una ganancia de 5, por lo cual de respuesta se tendrá idealmente 2 V, proponemos R1=2.5 K ohms, teniendo como resultado R2= 10 K ohms. Con esto se obtendrá a la salida el voltaje deseado y así la facilidad de usar un nivel de resolución de 8 bits en el ADC de la tarjeta de desarrollo, es decir, si los valores oscilan entre 0 volts y 5 volts, entonces se encuentra entre los valores de operación del ADC de la tarjeta de desarrollo.

3.3 Tarjeta de desarrollo del microcontrolador HCS08 de Motorola (freescale).

El sistema de desarrollo HCS08 se basa en la familia de microcontroladores MC9S08QG8 de FREESCALE, es una familia de bajo costo de propósito general, la cual tienen periféricos estándar, CPU. La tarjeta HCS08 cuenta con un procesador de 8 bits, memoria flash de 8Kbytes, 512 bytes de RAM, 1 puerto SCI, 2 canales PWM, convertidor ADC de 8 canales (convertidor analógico digital de 10 bits), característica de despertar (Wake-up) del modo de operación STOP o WAIT, regulador interno 3.3 volts que permite su operación con fuente externa de 3.3 volts a 5.5volts.

Características de la tarjeta de desarrollo:

- Programación por puerto usb (usb 2.0) o puerto serial (RS232 , DB9)
- 2 Leds para el usuario
- 2 Botones (tipo push bottom)
- 1 Potenciómetro para pruebas con el convertidor AD
- Voltaje de alimentación de 3.3 a 5 V cd con jack o terminales
- No requiere fuente especial para programación.
- Led de encendido.
- Puerto RS232 para monitor (SCI0)

Programación

- El programa monitor ocupa 1K de flash.
- Se programa con Code Warrior CW12V3.1
- En ensamblador, en C o ambos.
- Permite licencia especial hasta 4K en C

Interfase para el procesamiento y transmisión de las capturas realizadas

En la figura 3.3 se muestra la correspondencia de pines con el uso dedicado que se le ha asignado de fábrica al microcontrolador.

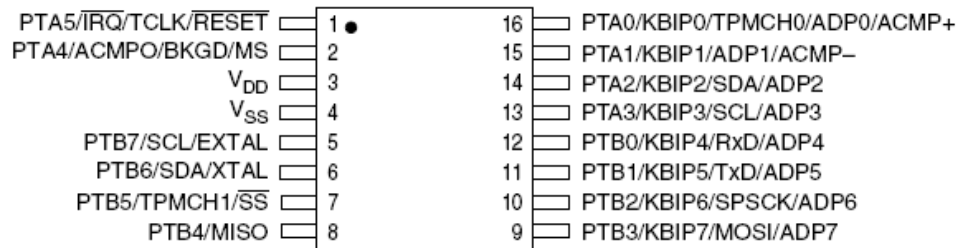


Figura 3.3 Distribución de pines en el microcontrolador

En la figura 3.4 se observa el diagrama a bloques de la estructura interna del microcontrolador.

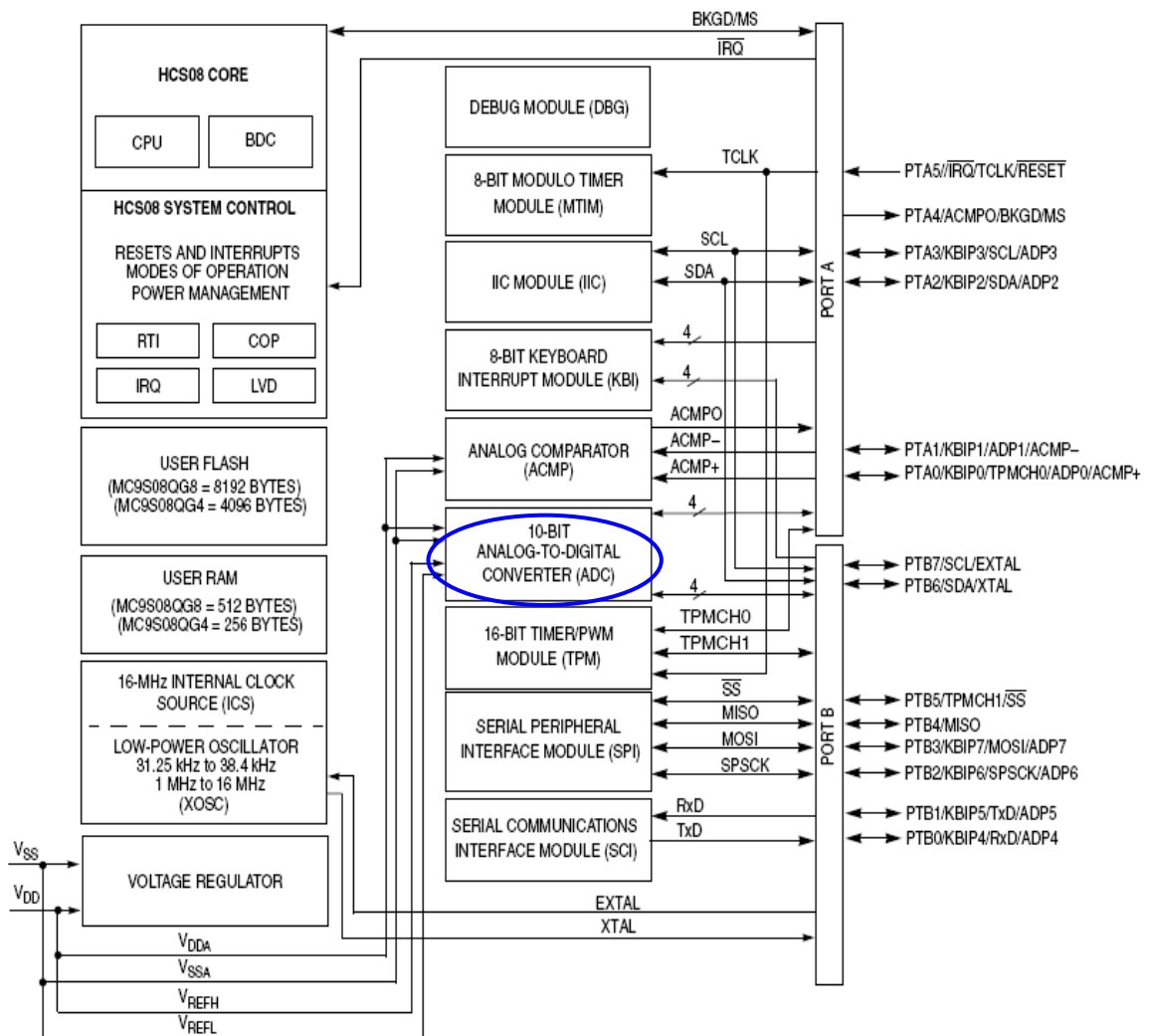


Figura 3.4 Diagrama a bloques de la estructura interna del HCS08

3.3.1 Características de componentes integrados a la tarjeta de desarrollo HSC08.

Para los fines de este proyecto, la principal funcionalidad y aplicación de la tarjeta de desarrollo, es en su totalidad el ADC (convertidor analógico a digital) La tarjeta de desarrollo tiene un módulo de conversión analógico a digital integrado, que cuenta con una resolución máxima de 10 bits, esto significa 1024 valores diferentes posibles en un rango de operación que oscila entre 0 volts y 3.3 volts. El ADC de esta tarjeta es del tipo “aproximaciones sucesivas”, es un tipo de ADC económico y lo suficientemente eficaz para este proyecto, además de ser un ADC unipolar, esto significa que no opera con voltajes negativos, así que si en un futuro se desea implementar la alternativa de monitorear valores negativos de temperatura, existe la posibilidad de cubrir tal necesidad, únicamente bajo las condiciones de modificar ligeramente algunos aspectos, tales como reducir la ganancia entre 10 y 50 y no en 100 como se pretende, además de añadir un sujetador de nivel, el cual sumará una señal de DC a la señal obtenida del sensor, esto para ajustar un centro de operación y así poder manejar voltajes negativos. Dicho ajuste de operación obliga a que también se tenga que modificar también en el software, que se describirá posteriormente.

3.3.2 Manejo previo de las señales adquiridas.

A continuación se detallará la configuración necesaria y requerida para llevar a cabo la captura de señales y preprocesarlas para su tratamiento.

Para el proyecto se hace uso del software Metrowerks Codewarrior el cual permite configurar y programar los componentes integrados de la tarjeta de desarrollo HCS08, para así lograr los fines propuestos.

CodeWarrior Development Studio para microcontroladores Freescale HCS08(x), brinda a los ingenieros una gran herramienta de creación y desarrollo. El sistema para HCS08 es rápido y sencillo. Este conjunto de herramientas provee de gran capacidad a cada ingeniero durante su ciclo de desarrollo, es por ello que este software acompaña a una gran tarjeta de desarrollo que es la misma que se usa en este proyecto.

Se explicará a continuación el modo de desempeño y uso de Codewarrior así como sus principales características.

La gran ventaja de Codewarrior es que se puede programar en ensamblador y en c/c++, como diversos lenguajes de programación de microcontroladores, ya que cuenta con “Beans”, o cápsulas que son configurables y ahorran mucho tiempo de programación, además de algo importante son portables, ya que pueden usarse en distintos proyectos, con lo que convierten a Codewarrior en un entorno IDE (Integrated Development Environment) totalmente visual, además que cuenta con “CPU Inspector” el cual es un diagrama del integrado donde se puede observar lo que se encuentra configurado en cada pin, así como otros detalles del Bean configurado. Algo relevante en Codewarrior es el llamado “Processor Expert”, que es una ayuda interactiva con el usuario y permite una configuración a fondo con extensas opciones habilitadas.

Conforme se va configurando el CPU o un Bean, el Procesor Expert facilita su manejo, de acuerdo al Bean, al evento o a las funciones propias que genera el Bean, así como también detecta posibles errores tanto de gramática como de depuración.

A continuación se explicará el entorno de desarrollo de Codewarrior de la figura 3.5.

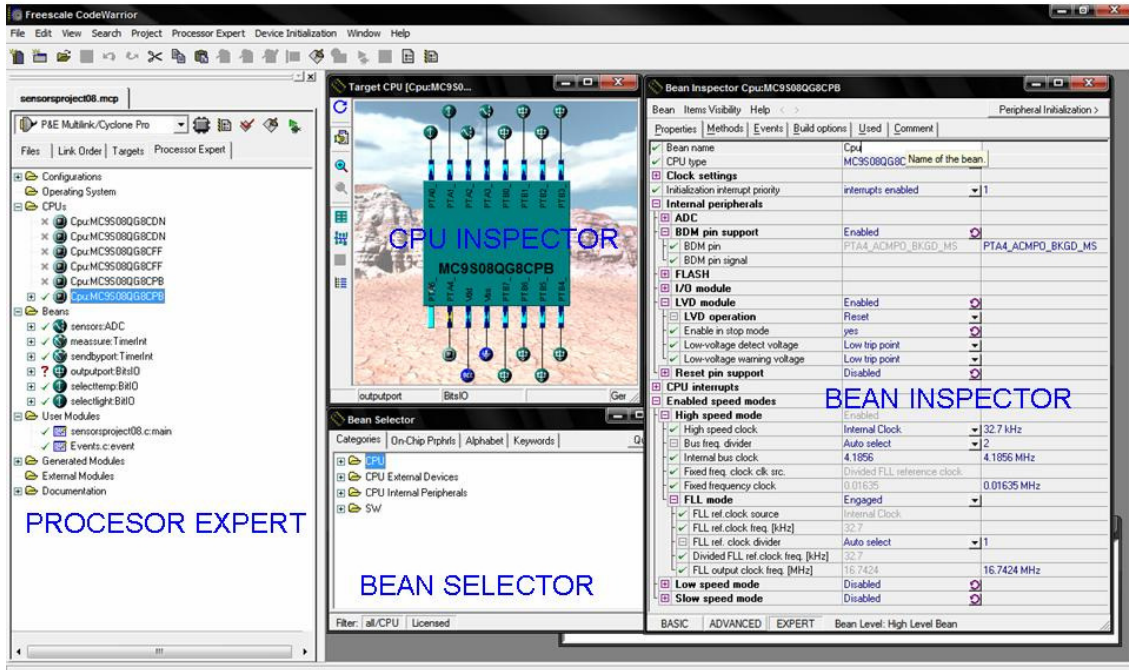


Figura 3.5 Entorno de Codewarrior

Processor Expert:

Es una ayuda interactiva y de expansibilidad de configuración (figura 3.6), permite la configuración de algún Bean, permite obtener información acerca de un Bean, o de un evento o función que genere dicho Bean, además de información como tipo de datos que maneja, opciones de retorno y dependencias. En cuanto a la expansibilidad, realmente habilita ediciones de configuración a mayor profundidad con lo que se convierte en un potente desarrollo y desempeño del microcontrolador.

Bean Selector:

Este complemento da la posibilidad de elegir todos los Beans posibles de una lista, es decir, de aquí se elige el Bean a usar en el proyecto, pueden agregarse tantos Beans como se deseen.

Bean Inspector:

En este complemento se pueden editar todas las configuraciones que serán habilitadas y usadas. Este complemento cambiará cualquier estado de configuración, así como también se puede revisar la configuración que previamente se ha editado (ver figura 3.7).

CPU Inspector:

En este complemento se pueden ver a detalle los Beans que se tienen ya configurados, así como ver en que pin se tienen habilitados.

A continuación se explicará brevemente qué Beans se requieren en este proyecto y posteriormente se entrará en detalle a cada Bean.

Se necesitaran los siguientes Beans:

- 2 timers: uno para el muestreo, y uno para el envío de datos.
- 1 ADC: para la conversión, se ocuparan 2 canales.
- 1 bitsIO: para designar el puerto de salida.
- 2 bitIO: para etiquetar el dato que se envía, según el sensor al que corresponde

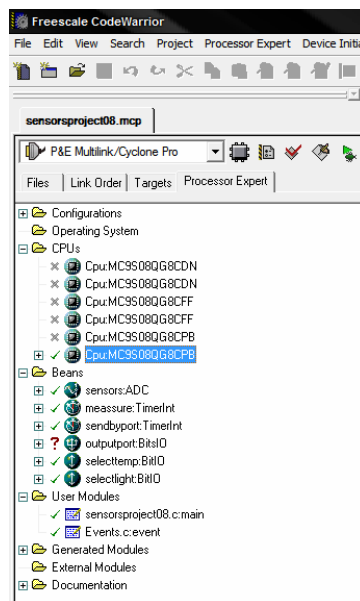


Figura 3.6 Processor Expert

Detallando la función y el sobrenombre de cada Bean se tiene:

- Sensors: es el Bean del ADC, el cual contiene los dos canales del mismo, configurados a 8 bits de resolución.
- Measure: este Bean corresponde al Timer que define los intervalos de muestreo (captura).
- Sendbyport: Bean de 8 bits, designado para los intervalos de envío de datos a puerto paralelo.
- Selecttemp: Bean de un bit, dedicado al etiquetado del sensor de temperatura.
- Selectlight: Bean de un bit, dedicado al etiquetado del sensor lumínico.

3.3.3 Conversión analógica a digital mediante la tarjeta de desarrollo.

Antes de explicar la configuración del Bean que llevará acabo la conversión de analógico a digital, es importante saber cómo funciona dicho Bean. El modo de operación del Bean es de la siguiente forma.

- Depende de un timer previamente configurado, el cual debe estar configurado con un tiempo de intervalo definido para el tiempo deseado en el que se hará muestreo (captura), con esto se generará un evento (método o función), al cual recurrirá cada cierto tiempo según se halla definido el intervalo de tiempo.
- Una vez configurado el timer, es necesario para que pueda operar correctamente el ADC, se procede a configurarlo.
 - Se selecciona el a/d a usar así como el número de canales, propiamente la tarjeta cuenta con un solo a/d el cual cuenta con 8 canales, para este proyecto sólo se seleccionan dos canales, en todo caso que se desee expandir el desempeño del proyecto, se puede agregar un sensor extra o los que se deseen y agregar un canal extra por cada sensor añadido, de igual forma se tendría

que modificar parte del software que recibe los datos en la computadora.

- Dado cada canal, se seleccionan que pines se utilizarán para cada canal del ADC, esto debe hacerse con cautela, no sólo por verificar que el pin que elijamos no se encuentre ocupado, pues en todo caso Codewarrior avisaría del error cometido, sino que de preferencia los pines que se elijan para el ADC, se encuentren ya sea cercanos o de preferencia en el mismo bloque de headers que el puerto de envío de la tarjeta de desarrollo. Con esto será más fácil el cableado.
- Así mismo se selecciona la resolución que se requerirá, para este proyecto se ocupará una resolución de 8 bits, aunque el máximo puede ser de 10 bits, así como el tiempo de conversión que tarda en convertir cada muestra de todos los canales empleados, por configuración establecida, el valor óptimo para un uso de 8 canales, es de 16 microsegundos, para este caso, se dejará con ese valor, por motivo de futuras expansiones.

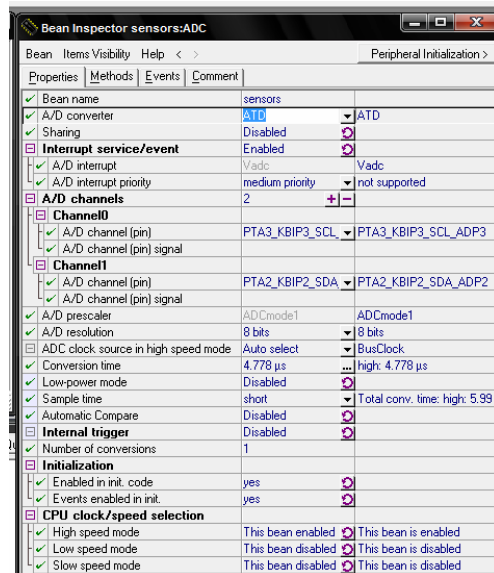


Figura 3.7 Bean Inspector

Este Bean “ADC”, generará un método en el código de programa “eventos”, en dicho método (función), se definen las acciones correspondientes que se desea se realicen, de acuerdo a los fines requeridos, análogamente, cada Bean genera un método, todos los métodos generados, se encuentran en un solo archivo de código llamado “eventos”.

3.3.4 Separación de señales para su procesamiento de acuerdo al sensor.

En este proyecto sólo se están usando dos sensores, eso implica el hecho de tener dos señales distintas, cada una correspondiente a un sensor, a continuación se explicará como se pretende distinguir cada señal a partir del valor que nos entrega el ADC. Para esto es necesario “etiquetar” los datos provenientes de cada sensor. Existen muchas maneras de etiquetar o saber a que sensor corresponde cada valor.

Se pretende etiquetar cada valor que se envía por el Puerto B de propósito general de manera simultánea, para ello se usarán dos bits más de los 8 bits que ya estamos usando para los datos. Es decir, los dos bits extras que se usarán, identificarán a qué sensor corresponde cada valor. Para esto se hará uso de 2 bits, debido a que con un solo bit, al configurar cierto pin en la tarjeta de desarrollo, se debe de configurar con un valor inicial, esto conlleva a que por default ya estaría etiquetando a un sensor, sin la necesidad de que se esté enviando el valor de dicho sensor, este error, no sería notablemente visible dentro del microcontrolador, sino al momento en que es recibido en la computadora por el programa de C#.NET el cual identificaría que el dato que está llegando sea el último estado enviado o bien el inicial que tiene por default el microcontrolador, lo tomaría cómo correspondiente a cierto sensor, es por ello que para evitar colisiones de datos, se propone el uso de dos bits, con esto las combinaciones hacen posible el etiquetado de 2 sensores.

Si a futuro se quisiera expandir el proyecto, sólo habría que modificar, en algunos aspectos el programa de C#.NET, puesto que los bits de etiquetado se

conectarán al puerto paralelo a los pines correspondientes al registro de estado, esto se detallará a fondo más adelante.

Es importante definir la cantidad de bits que se usarán para todo este proceso, en total serán 10 bits, de los cuales 8 son para datos y 2 para etiquetado, todo esto se detalla en la figura 3.8.

Cabe mencionar que el envío de los 2 y 8 bits será simultáneo, por lo que se pretende que no existan problemas de sincronización, es decir, se pretende el envío de etiquetas de manera simultánea con los datos, puesto que de esta forma no hay problemas de tiempos.

A grandes rasgos, este es el diseño del etiquetado.

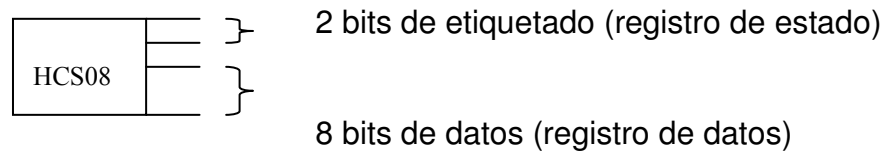


Figura 3.8 Distribución del uso de los bits en el puerto paralelo

En la figura 3.9 se observa detalladamente la distribución y conexión de los pines correspondientes detalladamente.

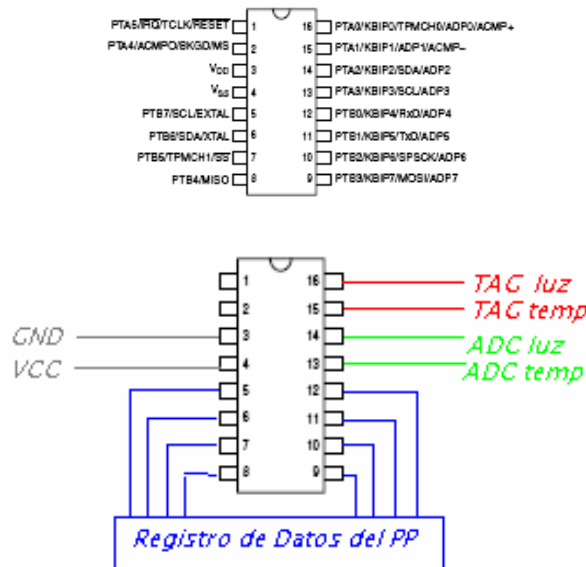


Figura 3.9 Esquema Detallado de conexiones

Capítulo 4

*Interfase receptora y de almacenamiento
de las capturas realizadas.*

Introducción

Como ya se ha explicado en capítulos anteriores, una vez capturadas las muestras de las señales de los sensores y convertidas a señales digitales, es necesaria una interfaz que pueda transmitir los datos hacia la computadora. Por cuestiones de practicidad y dominio en la interfaz, se optó por una interfaz mediante puerto paralelo de la PC, esto con motivos que a lo largo de este capítulo se explican.

4.1 Interfaz microcontrolador – computadora servidora.

Para realizar esta interfaz se decidió usar el puerto paralelo para comunicar el microcontrolador y la computadora. Como ya se explicó en el capítulo anterior, el microcontrolador, nos entregará 8 bits de datos mas 2 bits de etiquetado, esto implica, que dichos pines deben de conectarse específicamente en ciertos pines del puerto paralelo.

El puerto paralelo se apega al estándar IEEE 1284 liberado en 1994 y que define 4 modos de operación soportados, aún en la actualidad y son:

1. Puerto Paralelo Estándar (SPP)
2. Puerto Paralelo PS/2 (Bidireccional)
3. Puerto Paralelo Mejorado (EPP)
4. Puerto Paralelo con Capacidades Extendidas (ECP).

La mayoría de las computadoras personales recientes, tanto de escritorio como portátiles, presentan por omisión una configuración del puerto paralelo en dos direcciones de datos (bidireccional), para cualquier sistema operativo. Los sistemas operativos menos recientes (hablando de Windows 98 y anteriores) también son capaces de soportar este tipo de esquema, para recibir y enviar datos por el puerto de impresión, siempre y cuando se configure manualmente dicho modo de operación, preferentemente desde el *SETUP*.

Puerto Paralelo Unidireccional.

Considerando el modo de una sola dirección, comúnmente llamado Puerto Paralelo Estándar (*SPP*), existen tres direcciones consecutivas asociadas con un puerto paralelo, estas direcciones pertenecen al registro de datos (*Data Register*), el registro de estado (*Status Register*) y el registro de control (*Control Register*). Se le denomina dirección base a la que indica la propia del registro de datos, por lo general 0x378, así se tendría para el registro de estado la dirección inmediata siguiente 0x379 y para el registro de control la dirección 0x37A.

Existen alternativas diferentes para encontrar la dirección de los puertos, debido a que puede cambiar dependiendo de la arquitectura y organización interna de la PC, para ello es posible acceder directamente al *panel de control* de Windows y verificar el *sistema*, dentro de los recursos *hardware* se encuentra el *administrador de dispositivos*. El puerto paralelo se utiliza para la conexión de impresoras, por lo que aparecen etiquetados como LPT1, LPT2, LPT3 ó LPT4, según las características de la PC.

Es importante recordar que en el modo estándar, el puerto de datos sólo es de salida, de ahí que se le conozca como *unidireccional* y es de 8 bits. El puerto de estado es de sólo entrada, con 5 bits referidos en el conector y el propio de control tiene 4 bits de sólo salida. En resumen, bajo este modo se tienen 12 líneas de salida (de las cuales, 3 son de tipo activa bajo) y sólo 5 de entrada (con una sola línea de tipo activo bajo), tal y como se aprecia en la figura 4.1 puerto paralelo utiliza un conector hembra clase *D* de 25 pines (DB-25), definido como *TIPO A* por el estándar IEEE 1284 (obsérvese la *figura 1.b*).

En las figuras 4.1, se aprecia la distribución física de los pines en el conector DB-25. Para fines de análisis, se considera que los tres registros del puerto son de 8 bits, por lo que se tiene un orden significativo que es necesario respetar cuando se forma una palabra de configuración, por ejemplo, en el caso del registro de estado, se tiene disponible a partir del bit 4 y hasta el bit 8 (S7, S6, S5, S4, S3), los demás están comprometidos o reservados para otros propósitos. De acuerdo al diagrama interno aproximado mostrado en la figura

Interfase receptora y de almacenamiento de las capturas realizadas.

4.1 b, el bit más significativo del registro de estado (S7), trabaja con lógica negativa y está físicamente ubicado en el pin 11 del conector. Si se requiere leer una palabra de entrada a través de este registro, es importante considerar con qué lógica funciona cada pin.

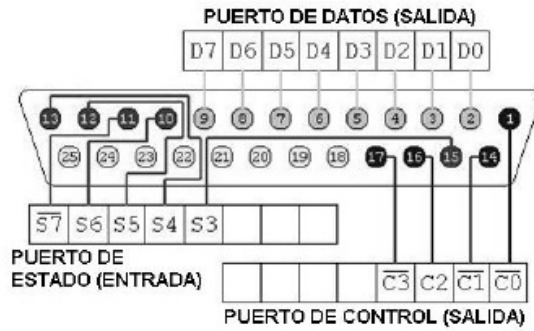


Figura 4.1 a)

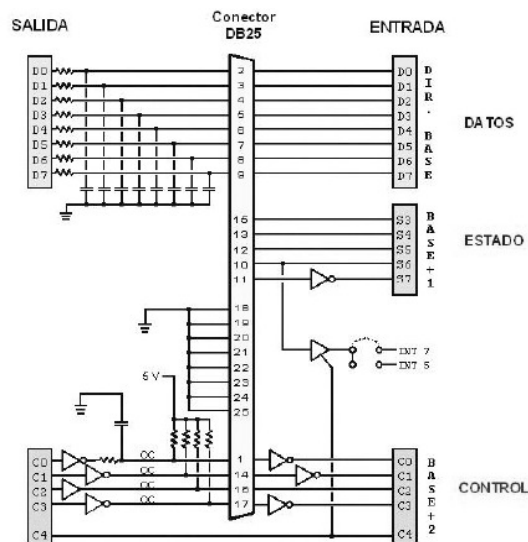


Figura 4.1 b)

4.1.1 Comunicación directa vía puerto paralelo a la computadora servidora.

Una vez definidos los aspectos de transmisión por puerto paralelo, se sobre entiende que hablamos de un puerto paralelo bidireccional, no sólo de salida, puesto que la función principal del puerto paralelo será la de ser un puerto de lectura, para así recibir los datos que se envíen por el mismo. La conexión física será tal y como se describa a continuación en las figura 4.2.a y figura 4.2.b, respectivamente.

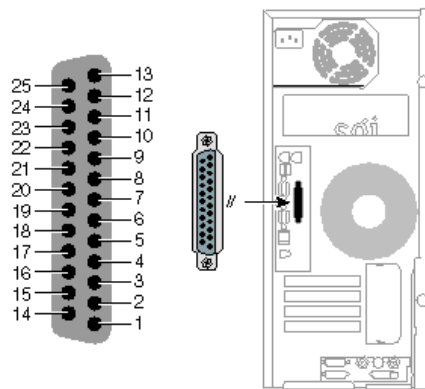


Figura 4.2.a Puerto paralelo PC



Figura 4.2.b Conector puerto paralelo

Interfase receptora y de almacenamiento de las capturas realizadas.

4.2 Procesamiento de los datos recibidos por puerto paralelo mediante C#.NET

Ya se ha hablado del funcionamiento de todas las etapas anteriores, microcontrolador, puerto paralelo, ahora se pretende, hablar de cómo se van a procesar los datos, una vez que se hayan recibido por puerto paralelo, es decir, ahora toca todo el tratamiento del monitoreo de las variables, que previamente ya se han capturado y transmitido hacia la computadora, ahora toca partida a un entorno de programación (IDE) reconocido, donde lo más importante es su gran desempeño, potencial y confiabilidad que brinda para plataforma Windows.

Dicho IDE es el visual estudio 2005, mejor conocido como “ado.net“, el cual se detalla a continuación en la figura 4.3.

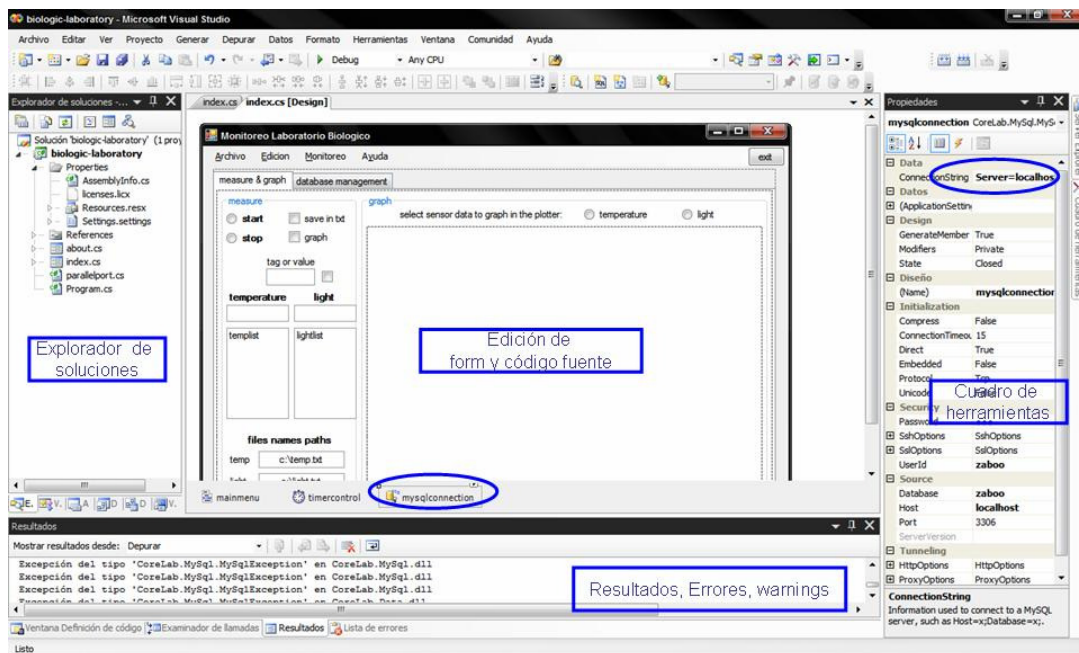


Figura 4.3 Aspecto visual detallado de visual estudio .net 2005

A continuación se explicará brevemente el programa en c#.net que se usará para este proyecto. Cabe recordar que la finalidad de este programa, es servir como interfaz entre el micro y el Web Server, donde la única finalidad es la de capturar los datos que se transmiten por puerto paralelo, separarlos de acuerdo al etiquetado previamente realizado, para así todos los valores medidos

almacenarlos en espacios diferentes, por conexión a base de datos y simultáneamente a archivos de texto, esto para resguardar la información en caso de perder la conexión a la base de datos.

En el diagrama de flujo de la figura 4.4 se muestra a grandes rasgos lo mencionado en el párrafo anterior.

A continuación se define y detalla la estructura con la que cuenta el programa desarrollado en c#.net.

Como ya se ha explicado en el capítulo anterior, el puerto paralelo sin configurar, por default está habilitado como unidireccional, esto implica la necesidad de establecer la bidireccionalidad del puerto paralelo, esto es mediante una instrucción que se manda al registro de estado, una vez que se ha establecido el puerto paralelo en modo bidireccional, se comienzan a leer datos. Donde uno de ellos es del registro de datos del puerto paralelo y el otro del registro de estado del puerto paralelo. Esta lectura debe de ser simultánea, debido a que una vez que se halla leído el dato del registro de estado, se debe hacer una comparación con el valor que se ha etiquetado por dicho registro, las comparaciones que se hagan serán con las que se hallan establecido para el etiquetado de cada sensor, como se usaron dos bits (2 pines) para el etiquetado, cada bit (pin) corresponde a un sensor, por lo que en este caso al bit más significativo se le ha asignado el sensor de temperatura, mientras que para el sensor de luz se le ha asignado el otro bit. Esta prioridad sólo ha sido para llevar un orden, puesto que no existen jerarquías entre datos correspondientes a señales, todo tienen el mismo privilegio a nivel software y hardware, por lo anterior se ha establecido el pin 5 del registro de estado para temperatura y el pin 4 del registro de estado para el sensor de luz, esto implica que se ha asignado un valor a cada etiqueta que corresponde a cada sensor, es decir, a temperatura se le ha asignado el valor 32 decimal y a luz el valor 16 decimal. Con estos valores se hacen las comparaciones en las condiciones del programa, para así saber a que sensor se está haciendo referencia.

Una vez que ya se haya detectado la correspondencia de cada valor con su sensor, se separan los datos, esto es para evitar colisiones de datos, ya que no es estrictamente necesaria la separación, basta sólo con que se alternen los datos, pero esto implica sincronizaciones, mismas que no sólo dependen de configuración y de programación, sino sobre todo de los recursos de cómputo que se estén ocupando, puesto que los retardos por recursos de hardware de las PC's, implica una gran influencia en los tiempos y en las sincronizaciones. Ya que se hayan separando los datos de acuerdo al sensor se procede a almacenarlos, base de datos usando un servidor local mysql, además de que se cuenta con el respaldo de la información en archivos de texto.

Dicho programa desarrollado en c#.net, además de almacenar a la base de datos en tiempo real los datos obtenidos, puede funcionar en modo archivos de texto, que almacenará únicamente los datos que se estén capturando en el mismo momento que se leen por puerto paralelo, es decir lo hará en tiempo real y los archivos de texto se guardarán por cada sesión de muestreo con nombres indicados por el usuario, aunque puede funcionar de forma autónoma, con una simple activación de forma local mediante un usuario. Recordemos que el término tiempo real, hace referencia a un concepto visto desde ingeniería del software, el cual dice que tiempo real en un sistema, es la diferencia de tiempo mínima para llevar a cabo determinado proceso, dicha diferencia de tiempo implicada por el uso de recursos, es decir, lo mas rápido o real posible, según permitan los recursos empleados en dicho sistema.

Así mismo, la operación del programa se torna a modo conexión a base de datos en servidor mysql. De esta forma se tendrá un mayor control de los datos, debido que a dicha base de datos, se le puede hacer modificaciones en tiempo real, es decir, varios usuarios pueden acceder a ella al mismo tiempo, esto implica una reestructuración del modo en que se están guardando los datos, la cual es mínima, pero resulta de gran ayuda para futuras expansiones.

Interfase receptora y de almacenamiento de las capturas realizadas.

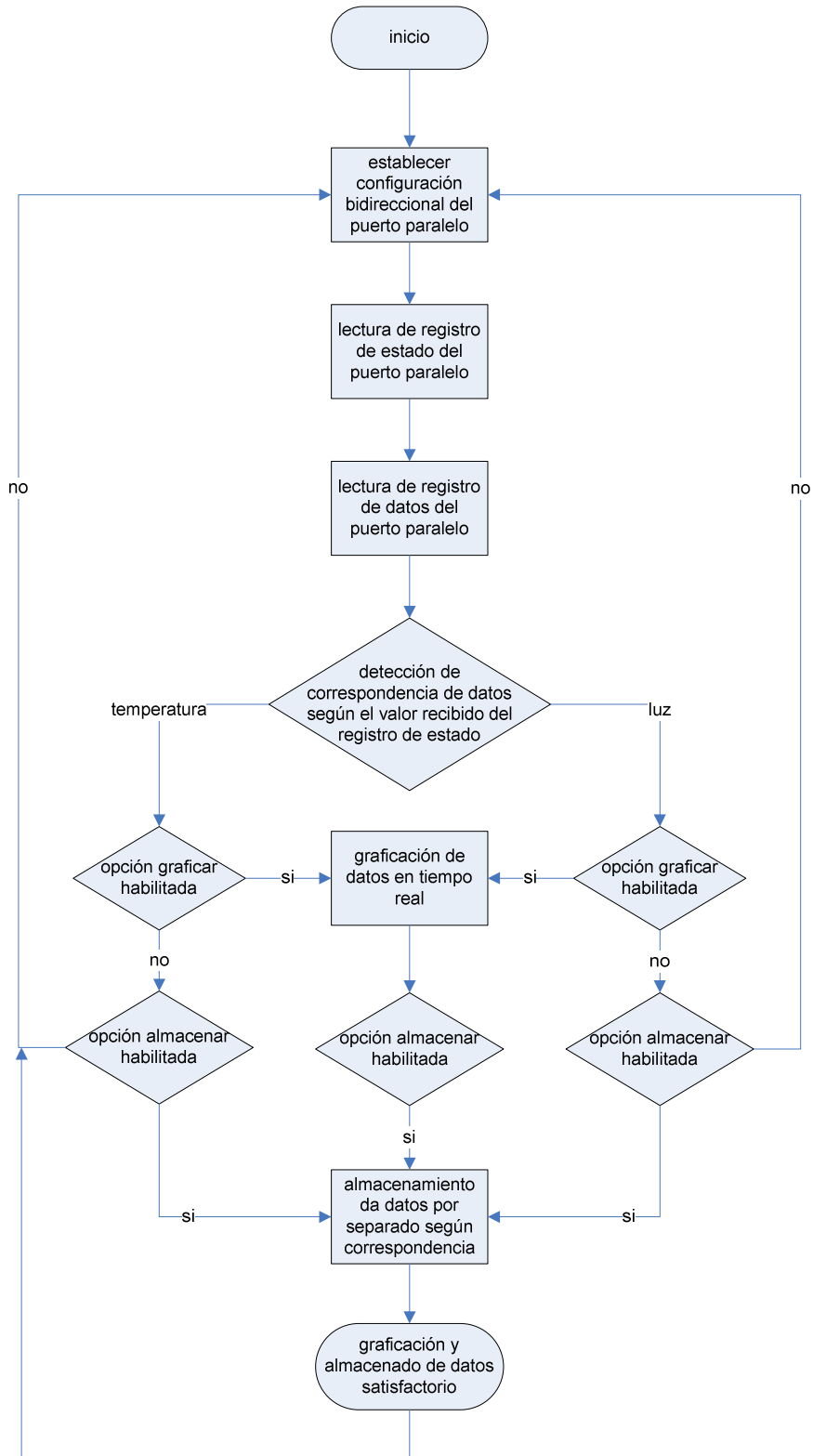


Figura 4.4 Diagrama de flujo de la estructura del programa de c#.net

4.2.1 Determinación de la correspondencia de los datos de acuerdo al sensor

Como ya se ha explicado en los puntos anteriores se usaron los pines 5 y 4 del registro de estado, debido a que el registro de estado es de sólo lectura, fue posible tomar dichos pines para la selección del sensor, es decir, para el etiquetado de la correspondencia de los datos.

Se usaron dos pines debido a que el microcontrolador tiene un estado inicial de sus puertos con valor de cero digital (estado bajo), cuando no está enviando información, esto crearía colisiones de datos o errores en destiempo de etiquetado, lo que se vería reflejado en problemas de sincronización, tal como se muestra en la figura 4.5.



Figura 4.5 Conector puerto paralelo

Debido a que se puede leer varios registros del puerto paralelo simultáneamente según lo descrito anteriormente, en el funcionamiento del puerto paralelo el registro de estado tiene la prioridad al igual que el de control sobre el registro de datos, esto ha sido así para evitar colisiones al enviar cuando aun no está escuchando el destino.

De manera simultánea se lleva el proceso de lectura en el mismo intervalo de tiempo, por ello que las instrucciones de lectura de registro, debe tener una jerarquía, donde el registro de estado tiene la prioridad ante el de datos.

Interfase receptora y de almacenamiento de las capturas realizadas.

A continuación se detallará la documentación del código para determinar, la correspondencia de los datos.

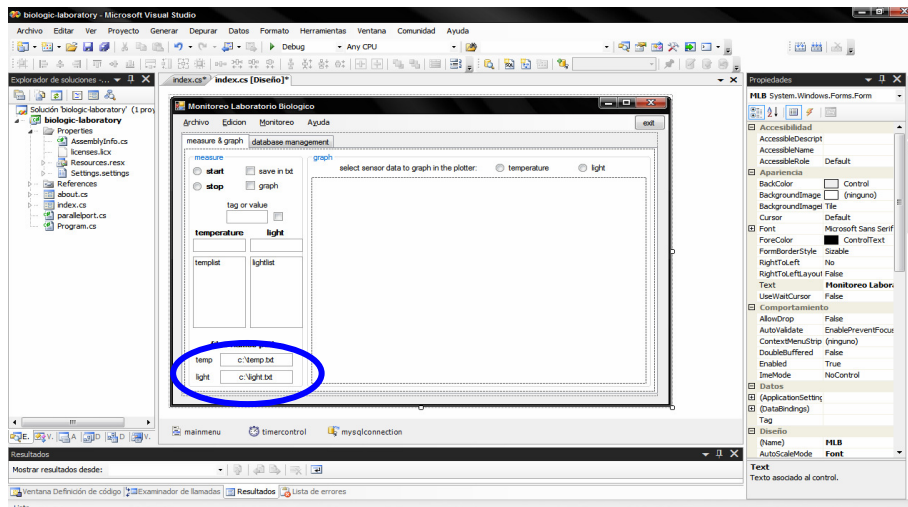
```
parallelport.Output(890,32); //establecer bidireccionalidad
etiqueta = parallelport.Input(889); //lectura registro de estado
valor = parallelport.Input(888) * (float)0.01294117647; //lee datos
```

Se puede observar claramente en las líneas de código superiores la prioridad de lectura que tiene el registro de estado. Esto para obtener de la variable “etiqueta” el valor que corresponde a cada sensor, con un 32 y 16, ambos en base decimal, correspondientes a sensor de temperatura y sensor de luminosidad, respectivamente.

Es importante aclarar que la función de lectura y escritura por puerto se encuentran en la librería “INPOUT32.DLL” la cual puede obtenerse en un banco de librerías en Internet.

4.2.2 Almacenamiento de la información.

Una vez que se ha determinado a qué sensor pertenece el valor que se esté leyendo, se separa por condiciones y se almacena la base de datos así mismo en archivos de texto. Para que posteriormente pueda ser leído por otro subproceso, en caso de ser implementado en un futuro.

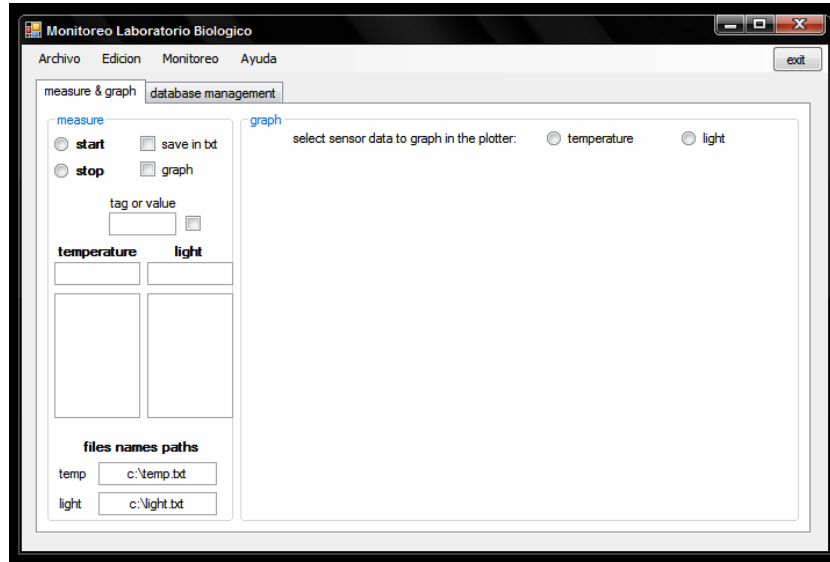


4.6 Visualización de la form del programa hecho en c#.net

Interfase receptora y de almacenamiento de las capturas realizadas.

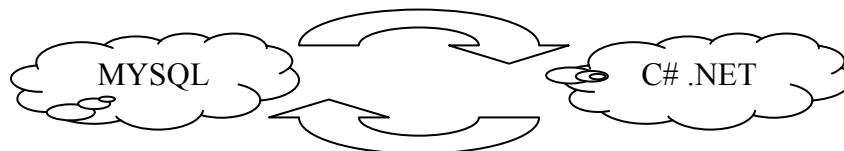
Mientras el programa comienza a capturar las muestras se puede previsualizar la información con un graficador integrado en el mismo programa. Tal como se muestra en la figura 4.7

Para la toma de las muestras se configura un timer a intervalos de 100 mili segundos.



4.7 Captura de muestras con graficados simultáneo

Existen varias formas de hacer la conexión a mysql desde un ambiente c#.net



4.8 Modos de conexión a bases de datos.

Una de las más populares o de las más usadas, es usando un conector de mysql hacia c#.net, esto descargando un driver de licencia libre para su uso en c#.net. Otra alternativa es descargar un ligero DSK (framework) de Microsoft para conectar a mysql, pero dicho conector no es gratuito, por lo cual se optó, debido a que su implementación es factible y sencilla, descargar de la página oficial de mysql un conector ODBC para ado.net, un archivo el cual es un instalador que configura automáticamente los servicios desde ODBC.

Interfase receptora y de almacenamiento de las capturas realizadas.

Dicho driver, es el “MySql ODBC Conector 3.51”, el cual mediante 3 pasos, se configura en el sistema deseado. Se optó por hacerlo mediante ODBC, por la confiabilidad que esto representa, ya que ODBC son orígenes de datos que ya están preestablecidos en el Sistema Operativo. Es decir, la estabilidad de la conexión es óptima, llevándola a cabo de esta forma. En la figura 4.9 se observa cómo se logra la conexión a la base de datos exitosamente, y así mismo se muestra en la figura 4.10 la base de datos a la cual se le ha agregado la información desde el programa hecho en c.net

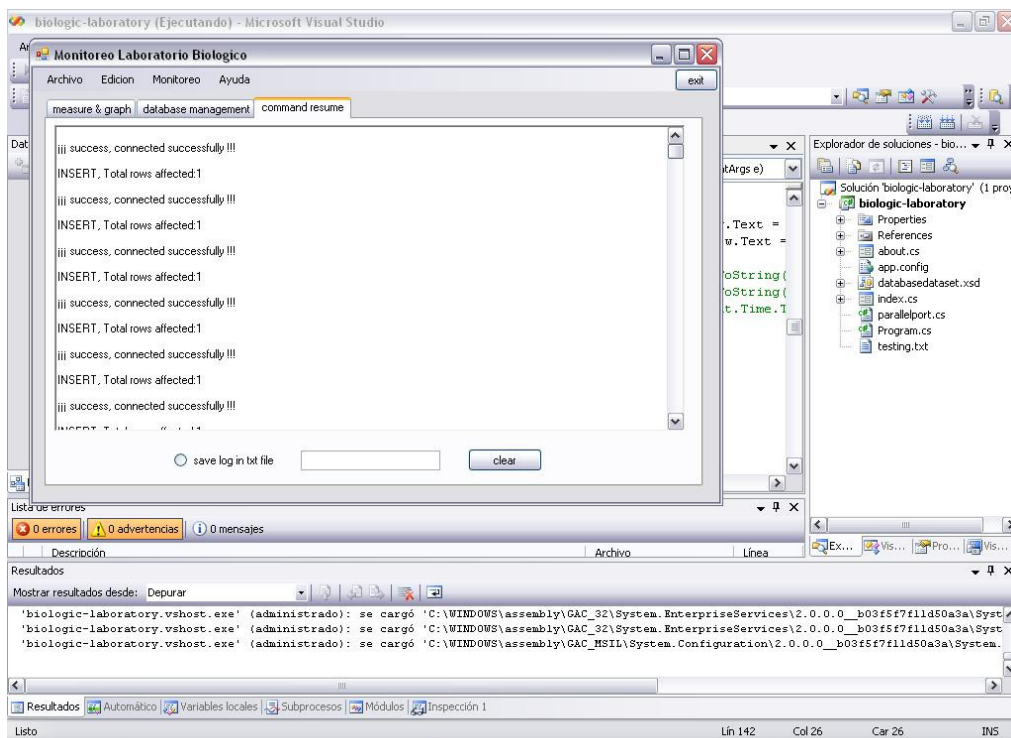


Figura 4.9 Conexión exitosa a la base de datos

Interfase receptora y de almacenamiento de las capturas realizadas.

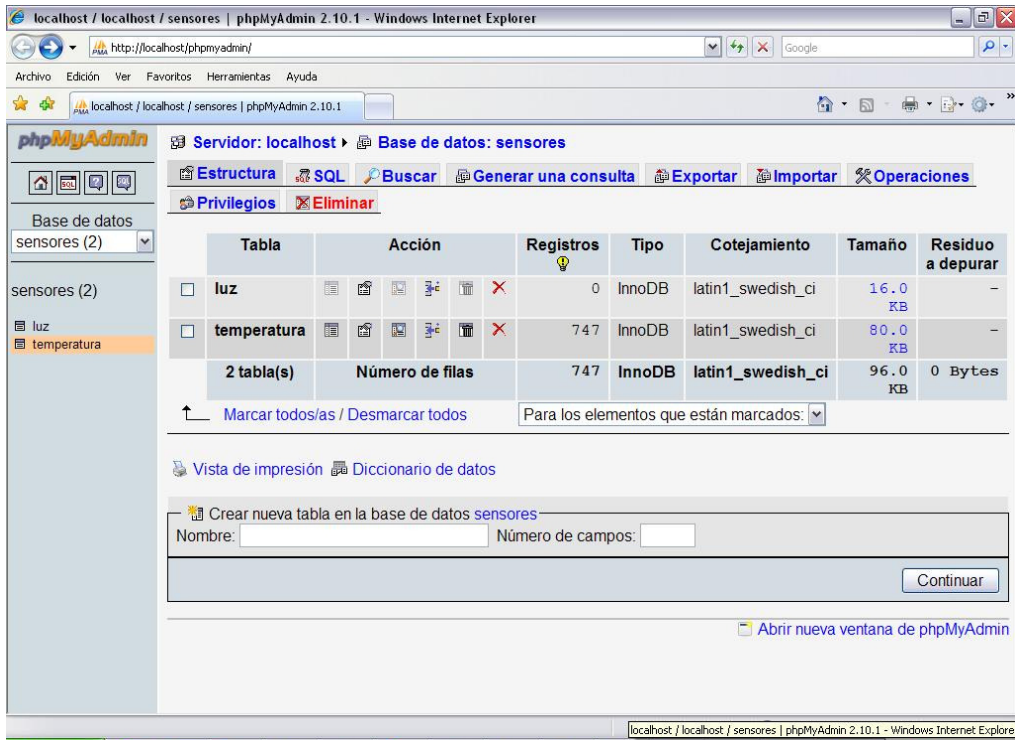


Figura 4.10 Base de datos estructurada y en uso

Es importante recordar que esta interfaz cuenta con las dos opciones de almacenamiento en archivos de texto y en conexión directa a la base de datos, esto con el fin de garantizar el respaldo de la información, por si algún motivo se pierde la conexión a la base de datos se encuentra el registro de archivos txt.

Capítulo 5

*Acceso visual a la información
previamente almacenada*

5.1 Lenguajes para la programación para Internet

El almacenamiento de datos es uno de los puntos importantes en este proyecto, puesto que gracias a éste, se logrará hacer un seguimiento del comportamiento de las variables a medir, y así poder graficar esos valores monitoreados y poder mostrarlos en Internet.

Para lograr el almacenamiento de datos, se realizan una serie de procesos, mediante los cuales se tratan los datos a graficar y almacenar.

La figura 5.1 se presentan las etapas que se llevan a cabo para almacenar los datos generados por los sensores utilizados y su tratamiento para el monitoreo.

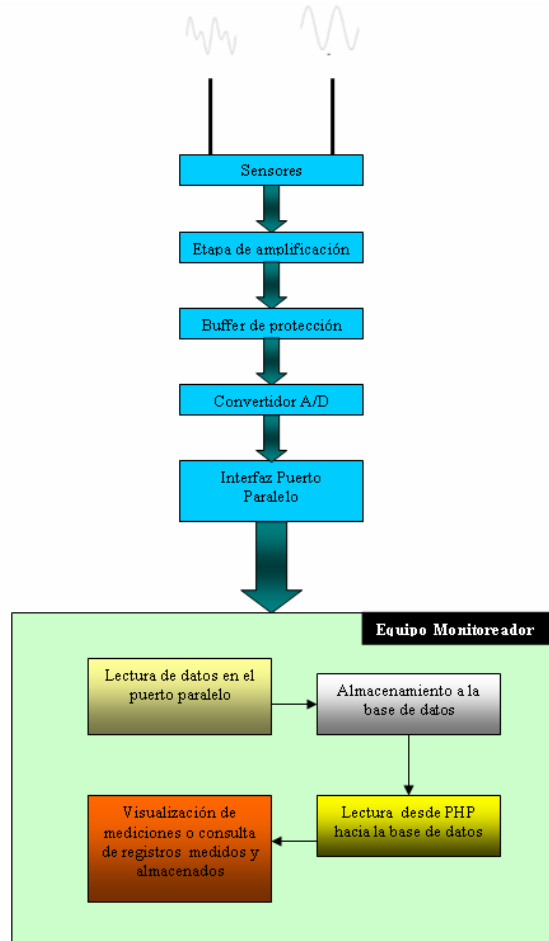


Figura 5.1 Diagrama general del sistema de monitoreo

5.1.1 PHP

Es el acrónimo de *HyperText PreProcessor*, que es un lenguaje tipo *Open Source*, embebido dentro de otro lenguaje de programación de páginas de Internet como HTML, siendo PHP un lenguaje utilizado del lado del servidor para hacer respuesta a peticiones realizadas por usuarios (o máquinas cliente).

PHP fue desarrollado por Rasmus Lerdof en el año de 1994, con la finalidad de mantener un control sobre los usuarios que accedían a consultar su currículum. Al siguiente año, surge una versión denominada *Personal Home Pages Tools*, versión que fue liberada para el público usuario, cuya funcionalidad principal era la de gestionar un libro de usuarios y un contador de visitas.

Posteriormente, se hicieron mejoras en las versiones existentes y surge así el PHP/FI, conocido también como PHP 3 (versión 3), que incluye manejo de formularios y conexión a base de datos como mySQL, siendo esta versión a partir de la cual empieza a surgir el potencial de PHP, para lo que es el manejo de datos y su destino en alguna base de datos, por lo que en los últimos años PHP ha sido el lenguaje mayormente usado en servidores de páginas Web.

La última versión de PHP disponible es la 5.2.4, que gracias a que es un lenguaje tipo *Open Source*, han existido mejoras y generación de varios códigos (Scripts), creados por la misma gente usuaria de este lenguaje, con lo que se publica y se liberan nuevas versiones.

La potencia y utilidad de PHP se orienta al procesamiento de información en formularios, diseño de páginas Web con contenidos dinámicos además del envío de cookies (registros que se generan durante la visita las páginas Web).

PHP permite el establecimiento de conexiones hacia diferentes tipos de manejadores de bases de datos, entre las cuales destacan:

- Adabas D
- Ingres
- Oracle (OCI7 and OCI8)
- dBase
- InterBase
- PostgreSQL
- Empress
- FrontBase
- Solid
- FilePro
- mSQL
- Sybase
- IBM DB2
- MySQL
- Velocis
- Informix
- ODBC
- Unix dbm

Además, soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados.

Como se mencionó, PHP es un lenguaje que puede ser embebido dentro de otro lenguaje, tomando el siguiente ejemplo, se pueden apreciar unas líneas de código donde se embebe PHP.

```
<html>
  <head>
    <title>Ejemplo PHP</title>
  </head>
  <body>
    <?php echo "Hola, este es un ejemplo con PHP!"; ?>
  </body>
</html>
```

Vemos que la forma en que se embebe PHP es utilizando las etiquetas (tags) de `<? ?>`, que son los separadores para identificar un lenguaje de otro, al momento de ser interpretado por un navegador de Internet (Browser).

5.1.2 Tratamiento de los datos en PHP

Una vez leídos los datos del puerto paralelo y almacenados en la base de datos, entra la etapa de PHP, en donde se establecen funciones que realizan la conexión hacia la base de datos que contiene los registros de las mediciones realizadas por los sensores.

En las páginas PHP, que es donde se codifican los scripts (códigos) necesarios para acceder a bases de datos, procesar los datos enviados en forma de peticiones que realiza un usuario y que éstas se ejecutan del lado del servidor.

Para establecer la conexión hacia la base de datos, en PHP se define lo siguiente:

```
<?php
    $conexion=mysql_connect("localhost","root","") or die ("No conecto");
    mysql_select_db("sensores");

    $setabT="select * from temperatura";
    $qT=mysql_query($setabT);

    $setaL="select *from luz";
    $qL=mysql_query($setaL);
```

Para iniciar la conexión hacia la base datos se declara la variable `$conexion`, la cual hará uso del valor que le devuelva la función `mysql_connect()`, donde los

parámetros que lleva, son el nombre o dominio del servidor, un nombre de usuario y una contraseña, dichos parámetros son usados para administrar la base de datos.

PHP es un lenguaje que permite realizar configuraciones para la conexión hacia una base de datos de manera sencilla, es por ello que la forma de establecer la conexión no requiere de pasos o configuraciones complicadas, siguiendo con el código la función `mysql_selectdb()`; es la función que requiere como parámetro el nombre de la base de datos a usar, como vemos, sólo es necesario el nombre, por lo que no hay que especificar alguna ruta de ubicación de la base de datos, PHP establece de manera automática la conexión con el servidor y la base de datos que éste contenga.

Para nuestro caso, la base de datos que manejamos se llama *sensores*, que a su vez contiene dos tablas, una llamada *luz* y otra llamada *temperatura*, en donde se guardarán los valores correspondientes a las mediciones sensadas.

Para el caso de la tabla *luz* se definen los siguientes campos:

- *ide*
- *dia*
- *hora*
- *valorl*

que son los campos en donde se almacenan los datos de las mediciones hechas, así como también el día y la hora a la que se realizaron y lo más importante, el valor de dicha medición.

A las variables `$setabT` y `$setaL` se les asigna el valor de las cadenas `"select *from temperatura"` y `"select *from luz"` respectivamente, que en este caso son las instrucciones o queries que se usan para seleccionar las tablas de la base de datos elegida anteriormente con la función `mysql_selectdb()`; por lo que estas variables serán usadas como parámetros de otra función.

Después con la función `mysql_query()`; se manda la petición o query asignado a la variable parámetro, para el caso de la tabla *temperatura*, la variable parámetro es `$setabT` y el valor retornado por la función se guarda en la variable `$qT`, para el caso de la tabla *luz* la variable parámetro es `$setaL`, y lo retornado por la función se guarda en la variable `$qL`.

La siguiente parte corresponde al tratamiento de los datos obtenidos de las tablas *luz* y *temperatura* de la base datos para poder graficarlos y mostrarlos en la página Web.

```
while($senT=mysql_fetch_array($qT)) {
    switch($senT[0]) {
        case "temp":

            $sen1=$senT[3];

            break;

    }
}

while($senL=mysql_fetch_array($qL))

{
    switch($senL[0])
    {
        case "luz":
            $sen2=$senL[3];
            break;
    }

}

$valores2=array(
    " °C "=>$sen1 ,

    "% luz"=>$sen2,
);
```

La función `mysql_fetch_array` obtiene y almacena los valores en un arreglo definido por alguna variable de los registros relacionados con el query hecho anteriormente, para este caso se definen las variables `$senT` y `$senL`. En estos

arreglos se guardan los valores que contienen los campos de las tablas mandadas en el query, siendo cada campo un elemento del arreglo, esto es, los campos *ide*, *dia* y *hora* corresponden a los elementos 0,1 y 2 de cada arreglo, el elemento 3 corresponde al campo en donde se guarda el valor de la medición, siendo éste el que se debe tomar para poder graficarlo.

Lo que se hace entonces es que para cada arreglo, se usa un *switch* en donde para los campos de *ide* en cada tabla, existe un valor que identifica a que sensor corresponde, por lo que el *case* de cada *switch* debe identificar ese *ide* donde para el sensor de temperatura el *ide* usado es *temp* y para el de luminiscencia es *luz*. Ya identificados esos elementos de cada arreglo, se declaran las variables *\$sen1* y *\$sen2*, a las cuales se les asigna el valor del tercer elemento que corresponde al valor de la medición sensada, siendo *\$sen1* para los valores de temperatura y *\$sen2* para los valores de luz.

Después se declara un arreglo asociativo denominado *\$valores2*, donde tendremos las etiquetas que aparecerán en las gráficas, mostrando las unidades asociadas al valor de las mediciones, cada elemento de este arreglo tomará el valor de las variables *\$sen1* y *\$sen2*.

Por último tenemos la función *mysql_close()*; que se encarga de cerrar o terminar el enlace con la base de datos.

```
mysql_close();
```

5.1.3 Graficación en PHP

La siguiente parte corresponde a cómo se generan las gráficas de los valores medidos, por lo que para esto, se usarán los valores del arreglo que está apuntado por la variable *\$valores2*.

```
header("Content-type: image/gif");
// Definimos las dimensiones de la grafica

$im_w2 = 800; // Ancho de la imagen
$im_h2 = 370; // Alto de la imagen
$im_margen2 = 50; // Margen lateral
$origen2 = $im_h2-95; // Origen de las barras

// Creamos la imagen
$imagen2 = imagecreate($im_w2,$im_h2);

// Definimos los colores
$bg2 = imagecolorallocate($imagen2,245,245,245);
$color2 = imagecolorallocate($imagen2,0,0,0);
$color22 = imagecolorallocate($imagen2,100,100,0);
$sombra2 = imagecolorallocate($imagen2,195,195,195);
$gris2 = imagecolorallocate($imagen2,150,150,150);

// Obtenemos la cantidad de valores
$cant2 = count($valores2);

// Distancia entre las barras
$dist2 = ($im_w2 - ($im_margen2*2))/ $cant2;

// obtención de los valores a graficar
$max2 = max($valores2);
$min2 = min($valores2);
```

La etiqueta *header()*; es usada para definir la creación de una imagen, que para el caso de PHP se pueden crear imágenes formato gif o png, en este caso se usará una imagen tipo gif para graficar los valores de las mediciones.

Después de indicar el tipo de imagen a crear se definen las dimensiones de la imagen, que están dadas en números de píxeles, además de definir un margen y un origen los cuales serán los valores o píxeles de referencia para comenzar a crear la imagen.

La variable *\$imagen1* es usada para recuperar de las dimensiones del contenedor de la imagen que se crea con la función *imagecreate*, que recibe como parámetros las dimensiones de la imagen (el ancho y el largo) definidos previamente. La función *imagecolorallocate()*, sirve para definir los colores a usar en la imagen a crear, así como los colores de los elementos que aparecerán dentro de ella. Esta función recibe como parámetros las dimensiones del contenedor de la imagen, y la combinación de los colores

primarios (rojo, verde y azul) cuyos valores van desde 0 a 255 para obtener un tono deseado.

Ya que únicamente se estarán graficando dos valores, con la función *count()* se obtienen el número de elementos que están almacenados en el arreglo que es apuntado por la variable *\$valores2*, por lo que con esta función al tener el número total de elementos, se distribuirá la forma en que se muestran las gráficas de las mediciones.

Las funciones de *max* y *min* obtiene el valor de cada elemento que está apuntado por la variable *\$valores2*, siendo estos valores los que se usarán para graficarlos y mostrarlos en la imagen que se creará.

Los valores que se han obtenido es necesario reescalarlos al tamaño de la imagen, por lo que la operación de re escalamiento está definida por:

```
$escala2 = ($origen2 - 10)/$max2;
```

Para generar las barras se define lo siguiente:

```
// Obención de los números de las barras

imageline ($imagen2, 40, $origen2- ($max2*$escala2),
  $im_w2-40, $origen2- ($max2*$escala2), $sombra2);

imageline ($imagen2, 40, $origen2- ($min2*$escala2),
  $im_w2-40, $origen2- ($min2*$escala2), $sombra2);

// valores numéricos
imagestring ($imagen2, $f2, 35- ($f_w2*strlen($max2)),
  $origen2- ($max2*$escala2)- ($f_h2/2), $max2, $gris2);

imagestring ($imagen2, $f2, $im_w2-35,
  $origen2- ($max2*$escala2)- ($f_h2/2), $max2, $gris2);

imagestring ($imagen2, $f2, 35- ($f_w2*strlen($min2)),
  $origen2- ($min2*$escala2)- ($f_h2/2), $min2, $gris2);

imagestring ($imagen2, $f2, $im_w2-35,
  $origen2- ($min2*$escala2)- ($f_h2/2), $min2, $gris2);
```

Donde la función *imagline()* recibe los parámetros de tamaño del contenedor de la imagen, coordenada inicial en X, coordenada inicial en Y, coordenada final en X, coordenada final el Y y el color a usar.

Se puede apreciar que las dimensiones del contenedor están dadas por el resultado de *\$imagen2*, el inicio del trazado de las barras empieza a 40 píxeles en el eje X hacia la derecha del contenedor, la coordenada en Y está dada por el valor que se obtuvo del arreglo con los valores sensados, pero este valor es reescalado por el factor calculado en la variable *\$escala2*, la coordenada final de X está colocada a -40 píxeles de lo que es el ancho del contenedor, mientras que la coordenada Y final toma el mismo valor que la coordenada Y inicial.

Para el caso de las coordenadas en Y (tanto inicial como final), el valor usado es el obtenido por la función *max()*, que corresponde a la medición obtenida por el sensor de temperatura.

Para el caso de la segunda gráfica, el valor usado para las coordenadas en Y (de igual forma, final e inicial), el valor usado es el obtenido por la función *min()*, que corresponde al valor del elemento del arreglo que guardó el valor de la medición para el sensor de luz.

Para colocar dentro de la imagen los valores que representa la gráfica se usa la función *imagestring()*, la cual, lleva como parámetros las dimensiones del contenedor, el tipo de letra o fuente, coordenada en X, coordenada en Y, la cadena a mostrar y el color. En este caso se hace uso de la función *strlen()* para convertir a una cadena de texto el contenido de una variable, en este caso, se convertirá a cadena de texto los valores que tiene las variables *\$max* y *\$min*.

Los parámetros para las cadenas de texto que se ingresarán a la imagen están dados por:

Acceso visual a la información previamente almacenada

```
// Definimos la fuente
$f2 = 30;
// Obtenemos el ancho y alto de la fuente
$f_w2 = imagefontwidth($f2);
$f_h2 = imagefontheight($f2);
```

Por último, definimos el ancho de las barras y de la sombra que aparece detrás de ellas

```
// Definimos el ancho de las barras
imagesetthickness($imagen2,200);
```

```
//sombra de las barras
imagesetthickness($imagen2,1);
imageline($imagen2,10,$origen2,$im_w2-10,$origen2, $negro2);

imagegif($imagen2);

imagedestroy($imagen2);
```

?>

La función *imagesetthickness()* recibe los parámetros de las dimensiones del contenedor de la imagen y el ancho que ocupará el elemento dentro de la imagen, en este caso serán las barras y las sombras de las gráficas.

La función *imagegif()* crea la imagen en la página de Internet (o sea el archivo procesa.php), y la función *imagedestroy()* se encarga de liberar la memoria que almacena los valores y parámetros que se usan para la creación de la imagen.

La figura 5.2 es el diagrama de flujo del tratamiento de los valores de las mediciones en PHP.

Acceso visual a la información previamente almacenada

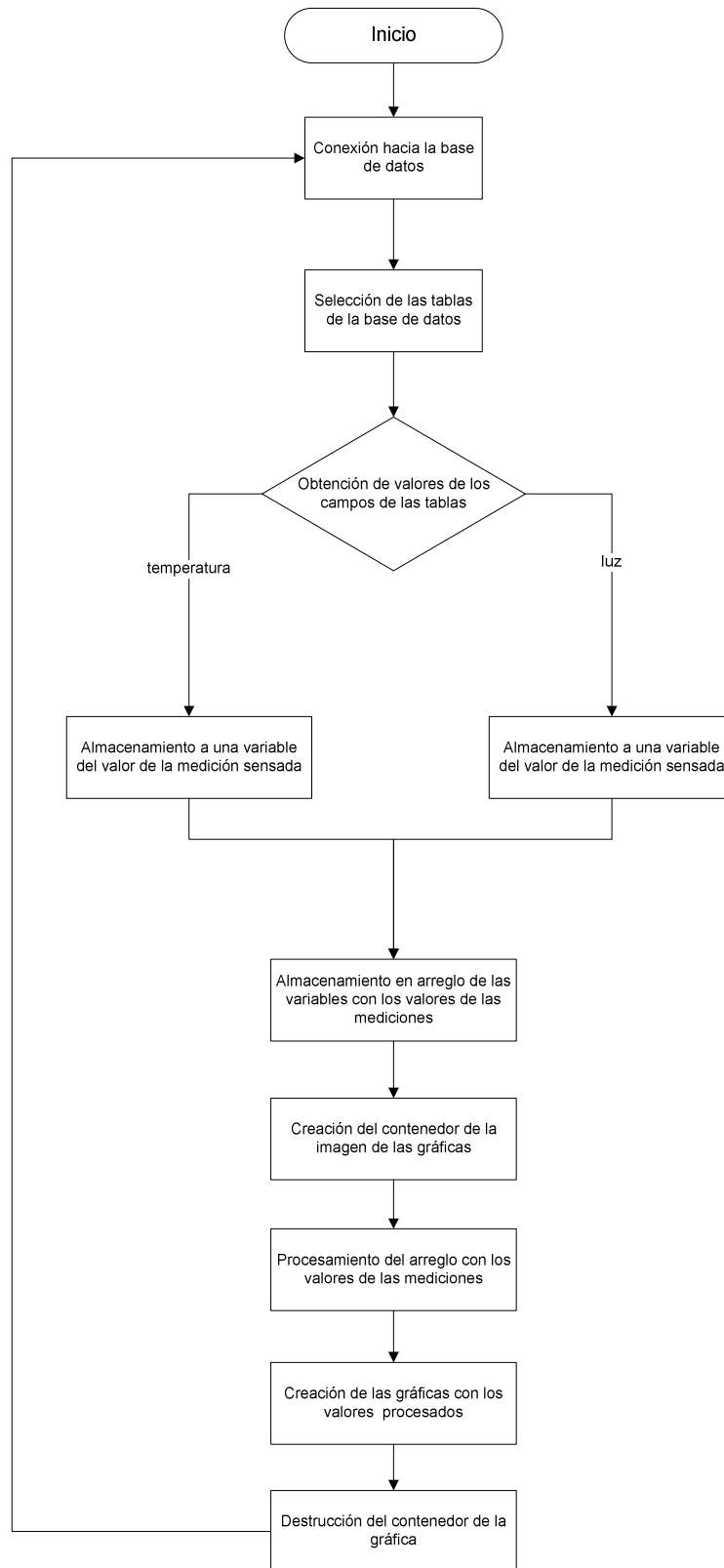


Figura 5.2 Procesamiento en PHP

5.2 Configuración del Servidor WAMP

En la actualidad existe gran variedad de programas para la implementación de servidores Web, en ocasiones dependiendo el hospedaje que se desee dar, se elige algún programa servidor en especial, siendo los más usados el IIS de Microsoft, Apache, Tomcat, por mencionar algunos.

La gran clasificación de éstos, es que unos son del tipo Open Source, mientras que como el caso del IIS, no es libre.

Algunas características de los servidores de páginas Web son, que toda la información que se procesa se ejecuta del lado del servidor (figura 5.3), haciendo que los usuarios de algún servicio que visitan determinado sitio, no presenten procesos lentos al momento de realizar alguna acción.

Todas las peticiones que escuchan los servidores Web las hacen a través del puerto 8080 (80/TCP), y estos mensajes son enviados a través del protocolo HTTP.

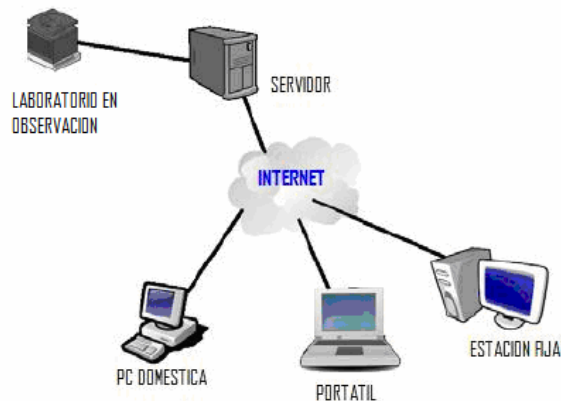


Figura 5.3 Servidor Web

Con respecto a las bases de datos, existen los programas usados para trabajar con los datos, dichos manejadores de bases datos pueden ser del tipo Open Source o *no comerciales*.

Existen varios programas con los cuales tratar los datos, siendo los más utilizados:

- SQL Server
- MySQL
- ORACLE
- Informix
- Paradox

Todos ellos funcionan bajo las mismas características y estándares, donde el lenguaje utilizado por todos ellos es el que está basado en el lenguaje SQL (Structured Query Language). Este lenguaje permite realizar una serie de operaciones que son muy recurridas con el manejo de datos, dichas operaciones son:

- Definición de Tablas
- Actualización hacia la base de datos
- Definición de vistas
- Definición de privilegios

La forma en la que se encuentra estructurada una base de se presenta en la figura 5.4

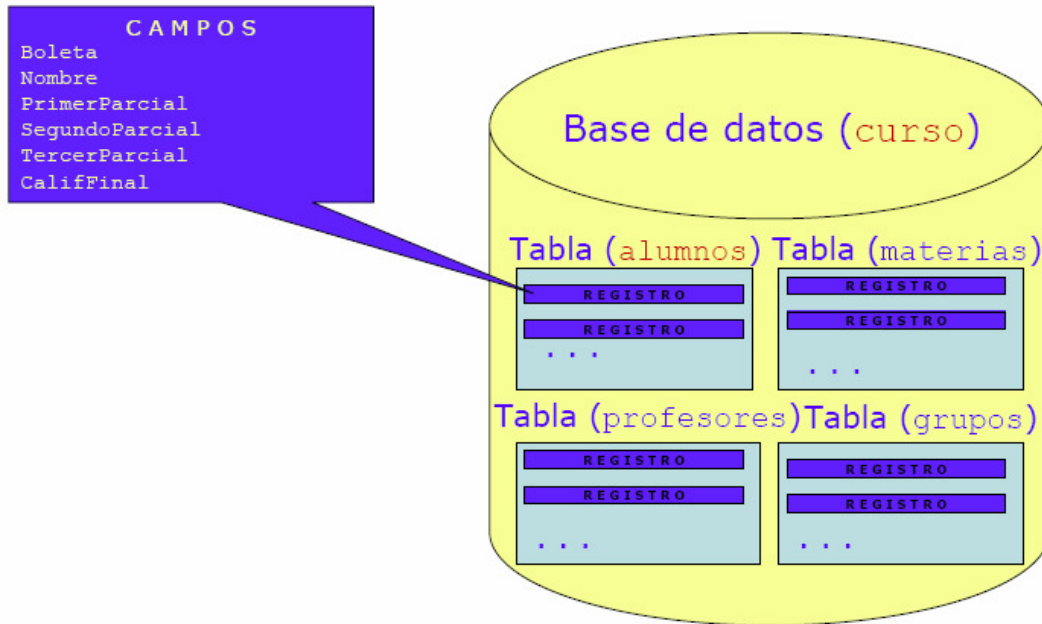


Figura 5.4 Estructura de una base de datos

Donde vemos que se tiene un nombre para la base de datos, las tablas que contendrán la información a respaldar en una serie de campos, en donde estos podrán ser de cualquier tipo de dato, dependiendo la información a tratar.

En los últimos años, la tendencia para el uso de servidores de páginas Web se ha inclinado a favor de los programas Open Source, y en este rubro, Apache se encuentra en poco más del 63% de los servidores de todas las páginas Web del mundo, según la compañía Netcraft de Inglaterra.

Con esto, los programas para el manejo de páginas Web, recaen en lo que es PHP, y este lenguaje de Internet toma mucha más fuerza con otra herramienta como MySQL, para el manejo de datos, puesto que PHP se comporta muy flexible con las implementaciones y conexiones para el uso de bases de datos en Internet.

Es por eso que en este caso usaremos un servidor Apache, con PHP y MySQL, montado en un sistema operativo de Windows XP Professional.

Esos programas se pueden instalar por separado y configurarlos de acuerdo a las necesidades que uno tenga, pero en nuestro caso trabajamos con un programa llamado WAMP (figura 5.5), que incluye todos estos programas y se simplifica la configuración de cada programa para que funcione.

El nombre de WAMP viene de las siglas Windows Apache MySQL y PHP. A continuación se describe el proceso de instalación y configuración necesaria para montar el servidor Web.

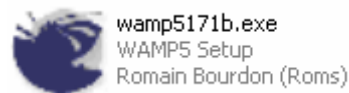


Figura 5.5 Programa instalador de WAMP

Existen varias versiones de WAMP, al igual que todos los programas presenta actualizaciones y nuevas versiones Beta que son liberadas cada determinado tiempo. En este caso usamos la penúltima versión (5.171b), por ser más estable que la última.

Para instalar WAMP, ejecutamos el archivo de instalación (figura 5.6):

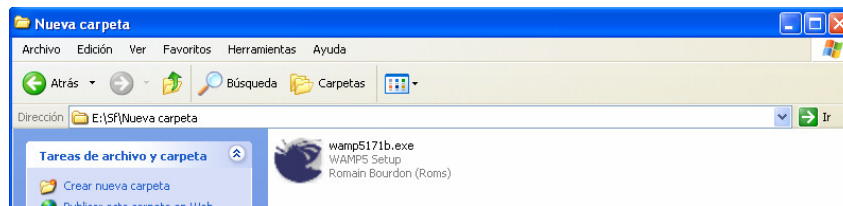


Figura 5.6 Instalación de WAMP

Al ejecutarlo nos aparece la ventana de la figura 5.7



Figura 5.7 Inicio de instalación de WAMP

Damos clic en siguiente, y nos aparecen los términos y condiciones del programa (figura 5.8), damos clic en Aceptar.

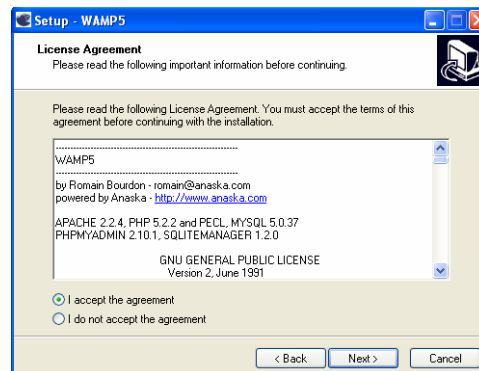


Figura 5.8 Condiciones de uso del programa

Elegimos una ruta destino para instalar todos los archivos necesarios (figura 5.9), entre ellos, la ruta de donde se encontrará la carpeta del hospedaje de las páginas Web, las carpetas para los archivos del manejador de la base de datos (MySQL) y también archivos de configuración de PHP y Apache. Por default se crea una carpeta llamada WAMP.

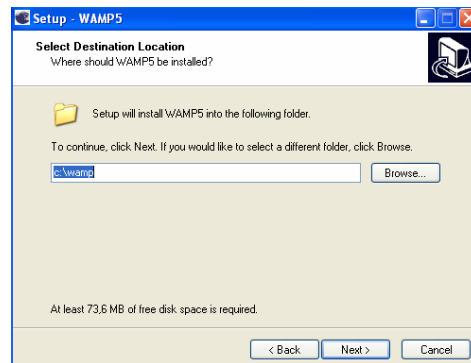


Figura 5.9 Ventana de destino de la instalación

Después nos aparece una ventana en la cual se nos indica si deseamos que el servidor se inicie cada vez que se que se prenda la computadora donde se instaló (figura 5.10), esto es, por si existe alguna aplicación que debe estar activa todo momento, y supongamos que se presenta un sistema de energización del equipo y se reinicia, con esto, al activarlo se asegura que se restablezca el servidor y se continúe con el proceso.

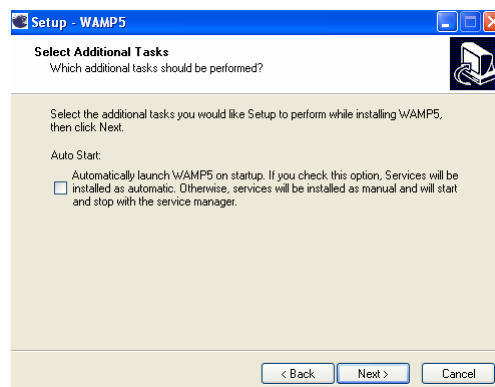


Figura 5.10 Configuración de inicio automático

Al dar clic en siguiente, aparece la ventana de la figura 5.11

Acceso visual a la información previamente almacenada

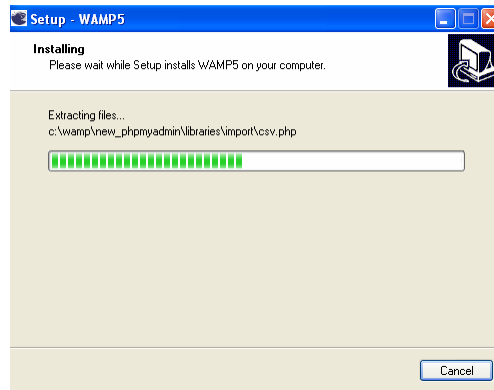


Figura 5.11 Etapa de instalación

En algún punto de la instalación, aparece una ventana en donde se deberá elegir una ruta para la creación de una carpeta que contendrá las páginas Web (figura 5.12), y archivos necesarios para procesar los datos y poder graficar los estados en tiempo real de los sensores.

Por default se crea la carpeta www.

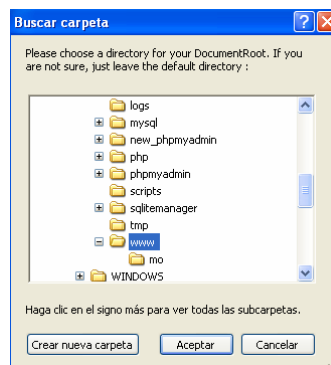


Figura 5.12 Ruta destino del almacenamiento de páginas Web

Una vez elegida la ruta de hospedaje para el servidor (figura 5.13), se nos pide crear un nombre para el servidor, esto es, con este nombre será con el que los usuarios podrán acceder al sitio.

Acceso visual a la información previamente almacenada

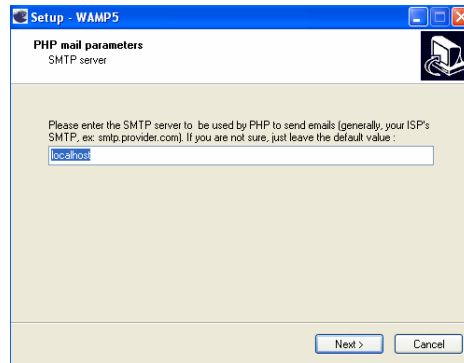


Figura 5.13 Nombre del servidor

A continuación, aparece una ventana donde podemos crear una dirección de correo para los servicios de correo que presenta PHP.

Para estos últimos casos, podemos dejar los valores por default que aparecen, pero si se tiene planteado usar este servidor para fines comerciales, es conveniente manejar otros nombres que no sean los de default.

Por último se debe elegir un navegador de Internet, para que cada vez que se inicie el servidor, se utilice por default (figura 5.14).

En este caso se usará el Internet Explorer, pero puede ser cualquier otro (Fire Fox, Opera, Safari, NetScape, etc.)

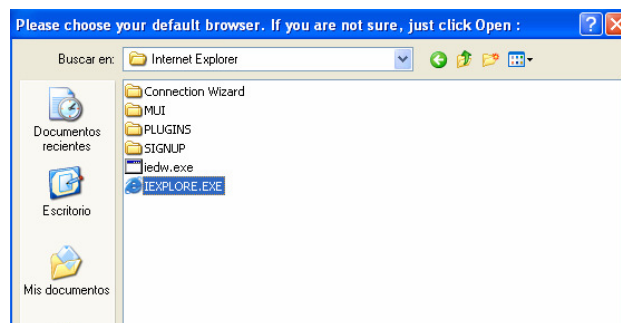


Figura 5.14 Configuración del navegador de Internet a usar

Elegido el navegador de nuestra preferencia, se da por terminada la instalación de WAMP (figura 5.15).



Figura 5.15 Fin de la instalación

Podemos ejecutarlo para ver que se hayan instalado los servicios necesarios para su uso e implementación del proyecto.

Los elementos que usaremos son el *LocalHost*, para acceder a las páginas subidas al servidor, el *phpMyAdmin* que se usará para configurar y diseñar la base de datos y el acceso al directorio *www* para cualquier modificación que se realice (figura 5.16).



Figura 5.16 Ejecución de WAMP

Como uno de los objetivos de este proyecto es mostrar los resultados que se obtiene de monitorear un proceso, es necesario que se habilite una librería para poder hacer uso de elementos de inserción de gráficos en PHP, es por ello que se debe agregar la librería **gd2**.

Para ello damos clic izquierdo sobre el icono de WAMP, elegimos configuración de PHP, en la opción de Extensiones de PHP buscamos la que sea **php_gd2** (figura 5.17).

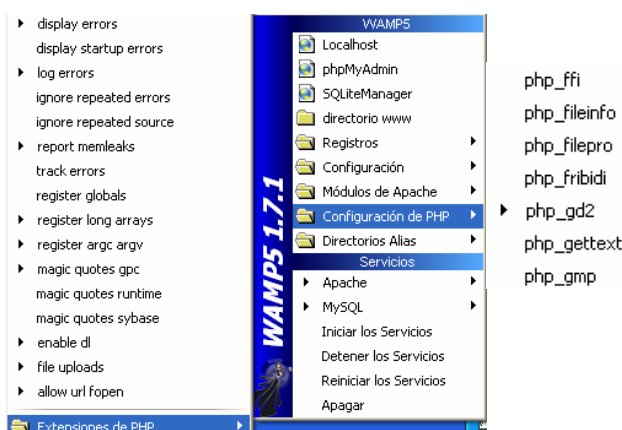


Figura 5.17 Habilidad de la librería gráfica

5.2.1 Diseño y creación de la base de datos

Anteriormente se explicó cómo está compuesta una base de datos, que son las tablas y los campos. Para este caso, se diseña una base de datos alusiva a los elementos que estamos manejando, como son sensores para monitorear variables como temperatura y luminosidad además del día y hora de su monitoreo, por lo que a partir de esta información podemos bosquejar el tipo de estructura que llevará la base de datos (figura 5.18).

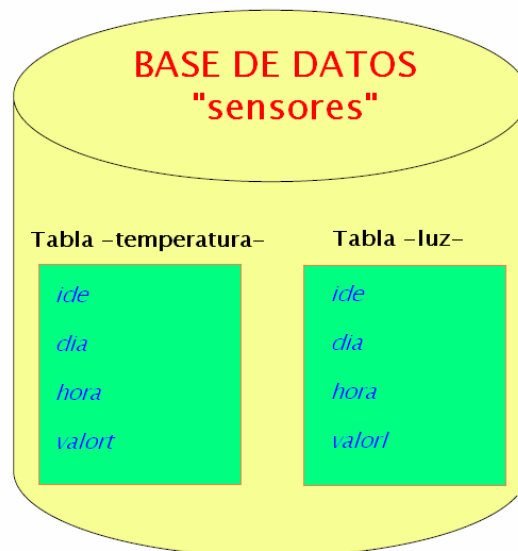


Figura 5.18 Base de datos propuesta

La figura anterior muestra cuáles son los elementos que formarán la base de datos, puede observarse que la base de datos lleva el nombre de *sensores*, y contiene las tablas llamadas *temperatura* que está compuesta por los campos *ide*, *dia*, *hora* y *valor* y la tabla *luz* con los campos *ide*, *dia*, *hora* y *valor*.

Para crear la base de datos se utiliza el *phpMyAdmin*, que no es más que un gestor y auxiliar gráfico para crear bases de datos (figura 5.19).

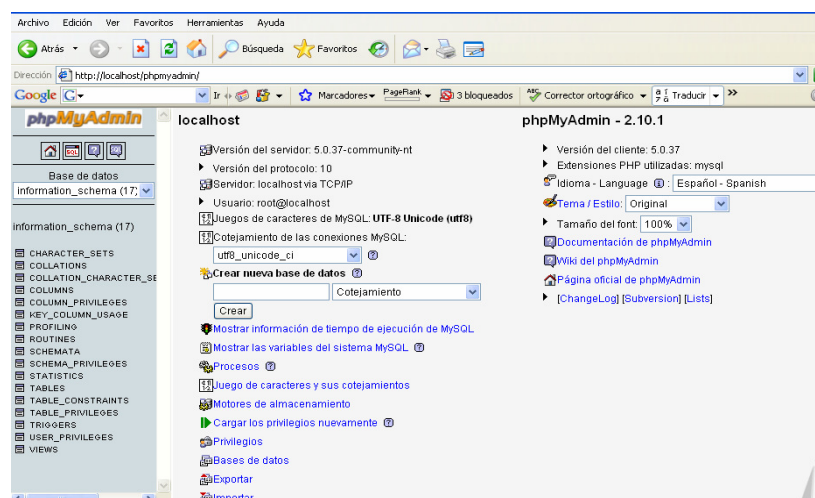


Figura 5.19 Interfaz phpMyAdmin

Una vez abierto el phpMyAdmin, se crea la base de datos, esbozada en el diagrama de la figura 5.18, donde la base de datos llevará el nombre de *sensores*, por lo que en el campo donde dice *Crear una nueva base de datos* ingresaremos el nombre de sensores del tipo cotejamiento (consulta) y finalmente se da clic en crear.

Nos aparecerá la ventana de la figura 5.20

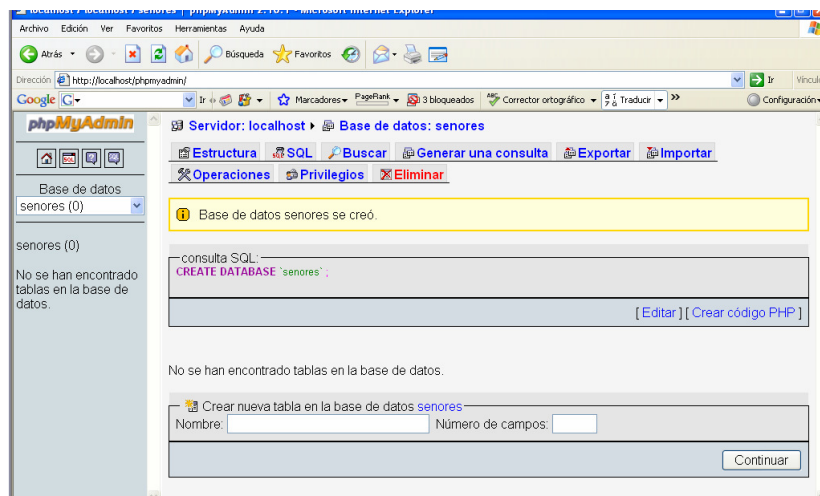


Figura 5.20 Creación de la tabla mediciones

En esta ventana se pide crear una tabla con determinado número de campos, para este caso se crearán dos tablas la de *temperatura* y la de *luz*, en ambos casos, los tres primeros campos de las tablas tendrán el mismo nombre, mientras que para el último y cuarto campo para la tabla de *temperatura*, el campo se llamará *valorl* y para el la de *luz* el campo se llamará *valorl*.

Al dar clic en continuar aparecerá otra con un recuadro con 4 filas, que corresponde a cada campo creado en la tabla, es aquí en donde se configuran las características y propiedades de los campos (figura 5.21).

Acceso visual a la información previamente almacenada

Servidor: localhost > Base de datos: senores > Tabla: mediciones

Campo	Tipo	Longitud/Valores ¹	Cotejamiento	Atributos	Nulo	Predeterminado ²	Extra					
	VARCHAR				not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
	VARCHAR				not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
	VARCHAR				not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>
	VARCHAR				not null			<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="checkbox"/>

Comentarios de la tabla:

Motor de almacenamiento:

Cotejamiento:

Figura 5.21 Creación y configuración de los campos de la base de datos

La forma en que se irán llenando los campos es de suma importancia, ya que esos nombres son usados para el procesamiento en PHP, para este caso, los campos serán los siguientes:

- ide: será del tipo char, con una longitud de 10, los demás parámetros se dejan tal cual.
- dia: será del tipo char, con una longitud de 10, los demás parámetros se dejan tal cual.
- hora: será del tipo char, con una longitud de 10, los demás parámetros se dejan tal cual.
- valorl: (para la tabla *luz*) será del tipo float, con una longitud de 10, los demás parámetros se dejan tal cual.
- valort: (para la tabla *temperatura*) será del tipo float, con una longitud de 10, los demás parámetros se dejan tal cual.

Damos clic en grabar y en la siguiente página aparecerá de manera gráfica la tabla creada en la base de datos (figura 5.22).

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción						
<input type="checkbox"/>	ide	char(10)	latin1_swedish_ci		No			<input type="checkbox"/>						
<input type="checkbox"/>	dia	char(10)	latin1_swedish_ci		No			<input type="checkbox"/>						
<input type="checkbox"/>	hora	char(10)	latin1_swedish_ci		No			<input type="checkbox"/>						
<input type="checkbox"/>	valort	float			No			<input type="checkbox"/>						

Figura 5.22 Campos creados para la tabla temperatura

Con esto ha quedado creada la base de datos que estará almacenando los valores que se estén obteniendo de la medición del puerto paralelo.

5.3 Consulta de datos almacenados

Una vez almacenados los datos de las mediciones realizadas, es posible verificar y consultar esos valores registrados un día en específico (figura 5.23), para ello se ha diseñado un método por el cual se pueda realizar dicha consulta a ese día, de tal forma que se pueda ver el valor de las variables de temperatura y luminosidad, así como la hora en la cual fue ingresada dicha medición a la base de datos.

Por ello se crea un archivo llamado *cons.php*, el cual está compuesto por código html y código php, que será el encargado de regresar los valores y resultados de la búsqueda.

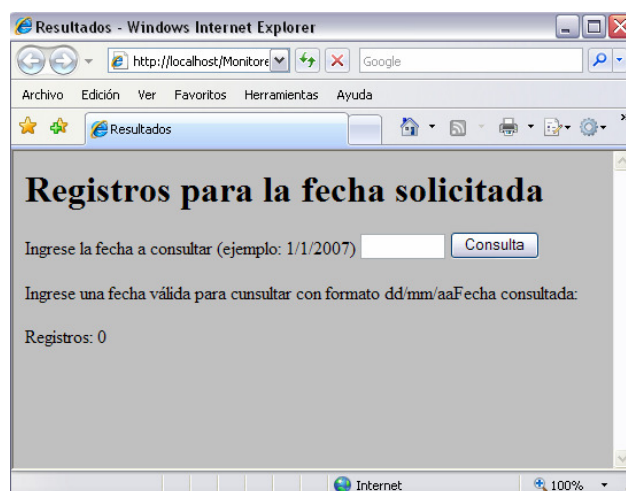


Figura 5.23 Página para la consulta de registros

Como se mencionó en el capítulo de PHP, es posible embeber (agregar) código PHP dentro de código HTML o viceversa, para este caso embeberemos código html en php, por ser ésta la extensión del archivo que se servirá para hacer las consultas.

Los formularios son elementos usados en páginas Web dinámicas, que bien pueden ser usados para trabajar datos y usarlos en alguna base datos, ya sea para consultar, agregar o realizar cualquier otra operación con los datos que pueda haber en una base de datos.

El archivo *cons.php* está diseñado de la siguiente manera:

```
<html>
<head>
  <title>Resultados </title>
</head>
<body>
<body bgcolor=silver>
<h1>Registros para la fecha solicitada</h1>

<form action="cons.php" method="post">
Ingrese la fecha a consultar (ejemplo: 1/1/2007)
<input type="text" name="fecha" maxlength="10" size="8" />
<input type="submit" value="Consulta"/>
</form>
```

Las líneas que están dentro de las etiquetas *head*, son usadas para que aparezca un título en la barra de título del explorador usado, en este caso el título será *Resultados*.

Después, dentro de las etiquetas *body* es donde se desarrolla todo lo relacionado a la consulta, con esto aparecerá un texto que dice *Registros para la fecha solicitada*, y abajo aparecerá un cuadro de texto, que es donde se

ingresará la fecha para la cual se va a consultar la relación de esa fecha con las mediciones y hora en que se realizaron.

La etiqueta de *form action* es usada para redireccionar a la página cons.php, una vez realizada la consulta, el método *post* indica cómo es que será tratado el dato a enviar, esto es, lo que se escriba será enviado en un solo paquete hacia la parte del código php.

La etiqueta de *input*, es usada para crear la caja de texto, que recibirá el valor de la consulta (para este caso será la fecha con formato dd/mm/aa), donde se especifica que esa caja de texto será del tipo *text* por el dato que maneja, y se usará una variable llamada *fecha*, que de igual manera esta variable tendrá que ser declarada en la parte de php para usarse en la consulta a la base de datos.

La otra etiqueta de *input* corresponde a lo que será el botón encargado de enviar el dato para realizar la consulta.

Posteriormente sigue la parte del código de php:

```
<?php
$fecha=$_POST[fecha];
trim($fecha);

if(!$fecha)
{
    echo"Ingresa una fecha válida para consultar con formato dd/mm/aa";
}

@ $db = mysql_pconnect("localhost", "root", "");
mysql_select_db("sensores");

$sql = "select *from luz where dia='$fecha'";
$resultado = mysql_query($sql);
$num_resultados = mysql_num_rows($resultado);

$sql2="select *from temperatura where dia='$fecha'";
$resultado2=mysql_query($sql2);
$num_resultado2=$resultado2;
```

```

echo "Fecha consultada: ".$fecha;
echo "<p>Registros: ".($num_resultados*2)."</p>";

for ($i=0;$i <$num_resultados;$i++ )
{
    $row = mysql_fetch_array($resultado);
    echo "<p>". " Hora:      ";
    echo ($row["hora"]);
    echo "<br>§ luminosidad:      ";
    echo ($row["valorl"]);

    $row2=mysql_fetch_array($resultado2);
    echo "<p>". "Hora:      ";
    echo ($row2["hora"]);
    echo "<br>Temperatura en °C:      ";
    echo ($row2["valort"]);

}

?>

</body>
</html>

```

En esta parte se reasigna a la variable *\$fecha* el valor ingresado como fecha, después, ese valor se somete a un filtrado, esto es, se prepara el contenido de la variable de tal forma que sea posible usarlo sin que genere algún error durante la consulta. La función *trim()* sirve para quitar los espacios en blanco que pudiera traer el valor ingresado.

En caso de no haber ingresado algún valor, se manda un mensaje de error indicando que se debe ingresar un valor de una fecha, este mensaje aparece debido a que se compara el estado de la variable *\$fecha*, si ésta está vacía, el condicional *if* evalúa el valor y dependiendo el valor, muestra o no el mensaje.

Posteriormente se realiza el enlace con la base de datos llamada *sensores*, el procedimientos es similar al caso del enlace para el tratamiento de datos

explicado en el capítulo 4, a diferencia de que ahora se usa un condicional que es el que permitirá obtener los resultados para la fecha deseada a consultar.

Las variables *\$sql* y *\$sql2* se envían como parámetros a la función *mysql_query()*, almacena la cadena que será usada para hacer el query a la base datos, como se busca mostrar los campos relacionados al valor que toma la variable *\$fecha* de la tablas *temperatura* y *luz*, se usa el condicional *where* para indicar que se devuelvan los valores de los campos que corresponden al de la fecha ingresada.

La función *mysql_num_rows()*, devuelve a las variables *\$num_resultados* y *\$num_resultados2*, el número de filas totales que corresponden al valor del campo que tiene el valor de la fecha ingresada para hacer la consulta, este valor es muy importante, ya que con el se mostrarán en la página todas las relaciones para la fecha ingresada y sus valores de los otros campos, como será el valor de las mediciones y su hora de realización.

El ciclo *for* es usado para listar todos los valores y relaciones encontradas para la fecha consultada, donde la función *mysql_fetch_array()* es usada para obtener los valores de toda la fila que corresponde al query realizado (o sea la consulta de la fecha), por lo que se almacena en las variables *\$row* y *\$row*, para así usarse en el ciclo *for*, que comienza a mostrar todas las filas con la relación de la fecha, hasta el número de fila que arroja la función *mysql_num_rows()*.

Cada *echo(\$row[])* y *(\$row2[])*, mostrará el valor del campo durante la realización del ciclo *for*, enlistando en la página Web todos y cada uno de los valores de los campos *hora*, *valorl* y *valort*, que coinciden con el de la fecha consultada.

5.4 Sitio Web

Hasta el apartado anterior, se describe cuál es el funcionamiento de todo el sistema de monitoreo, desde su etapa de adquisición de la información (los datos), hasta la forma en que estos datos se representan de manera gráfica, así como también un método de búsqueda de alguna fecha en específico, elemento que es muy importante, a la hora de querer tener información de algún acontecimiento, en este caso tener un registro del comportamiento de las mediciones realizadas.

Es por ello que para todo esto se implanta un servidor Web, que será el encargado de permitir y dar acceso a toda la información generada y almacenada en a base de datos.

El sitio de Internet creado para dar solución a la visualización de toda esta información, puede ser implementado en una red LAN (Red de Área Local) o bien si se dispone de salida hacia Internet, es posible llevarlo hacia esa instancia.

La forma de diseño de este sitio de Internet, permite al usuario realizar una navegación simple y ordenada, para lo que es la consulta de los datos, permitiendo conocer y navegar por otros temas muy relacionados a lo que es el monitoreo vía Internet.



Figura 5.24 Entrada al sitio de Internet

La figura 5.24 muestra lo que es la página principal para acceder a los contenidos relacionados al monitoreo en tiempo real.

Al entrar en el enlace, aparece la siguiente página en donde se puede ver que hay un menú, que permite la navegación hacia otras páginas dentro del mismo sitio (figura 5.25), donde los enlaces o links más relevantes para este caso, son el de **Monitoreo** y **Consulta de datos**, ya que estos son los que llevan a cabo todo el proceso de monitoreo y seguimiento del comportamiento de los sensores colocados en el área de monitoreo.

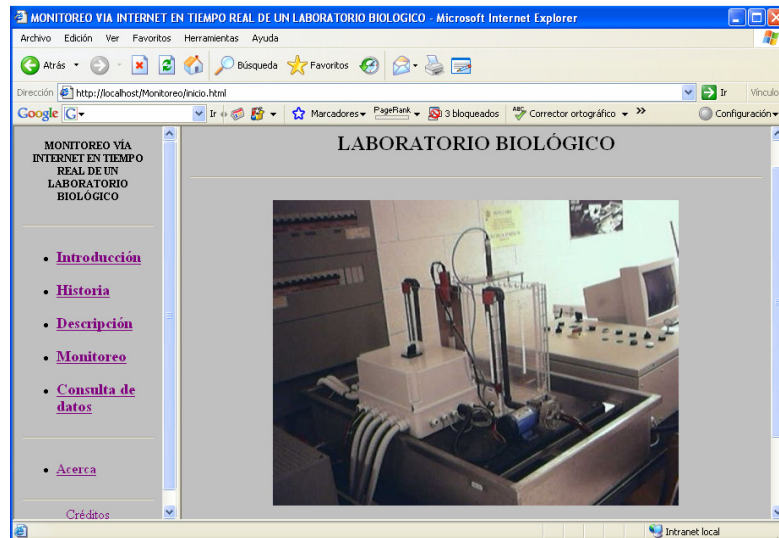


Figura 5.25 Navegación por el sitio de Internet

Para el caso de los tres primeros links del menú, éstos solo contienen información y descripciones relacionadas al sistema de monitoreo, siendo que éstos no realizan alguna acción concreta con lo que es el monitoreo.

5.4.1 Estructura del Sitio Web

Todo el sitio Web creado para monitorear, está compuesto de varias páginas que se enlazan unas con otras, ya que el diseño está basado en el uso de *frames* (marcos o recuadros), esto es, con la finalidad de que siempre y durante la navegación se esté en la misma página, ya que por cuestiones prácticas y de presentación, es más cómodo y servicial estar siempre en el mismo ambiente a tener que estar regresando entre las páginas.

Este tipo de programación de páginas de Internet requiere que se tengan muy en claro cada una de las partes que conforma al sitio, y que puede resultar algo confuso al momento del diseño.

En el capítulo de la instalación de Servidor (en este caso WAMP), se explica que todo lo que será contenido de páginas y archivos a usarse en Internet

deben estar en una misma carpeta, es por esto que para el diseño de los frames, todos los archivos .html o .php deberán estar en la misma carpeta.

En la estructura del contenido de los archivos .html es donde se diseñan los frames, para que se muestre todo el contenido en la misma página, a excepción del link de búsqueda, todo lo demás está en la misma página.

La figura 5.26 muestra un diagrama de cómo están enlazadas todas las páginas del sitio.

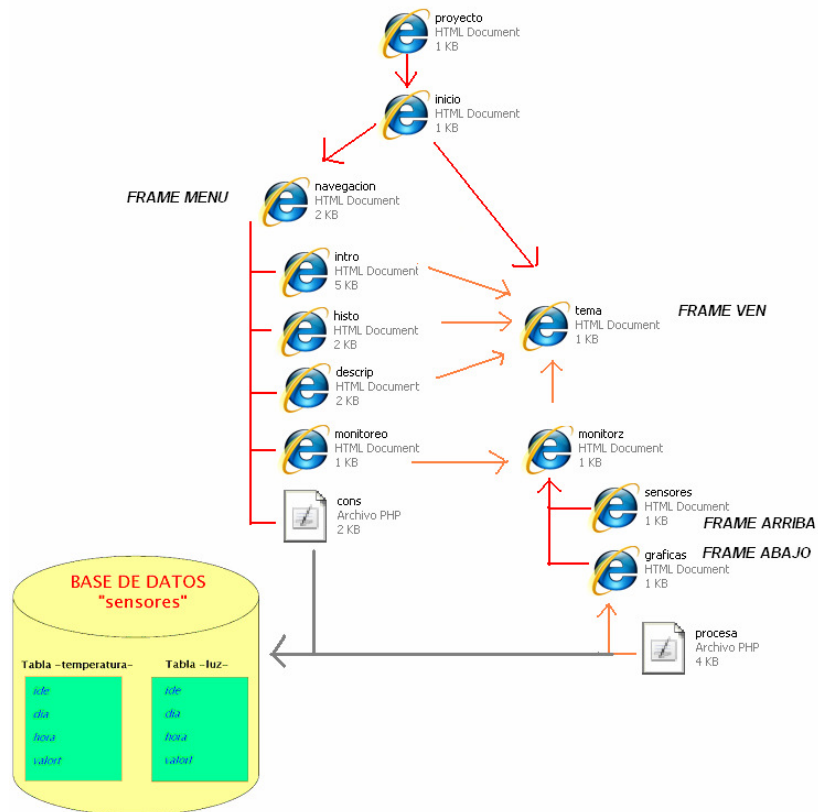


Figura 5.26 Estructura del sitio Web

Al momento de ingresar a la página de *proyecto.html*, está el enlace a lo que es la entrada a la página de *inicio.html*, donde esta página está formada por dos frames verticales indicado por las flechas rojas, uno llamado *nvegación.html*, con un identificador de frame llamado **menú** y el otro frame de la página *tema.html* con un identificador de frame llamado **ven**.

En el frame de **ven** es en donde aparecerá todos los contenidos de los enlaces que aparecen en el frame **menú**, por lo que cada vez que se elige un link, su contenido se despliega en el frame **ven**, estos despliegues se indican con las flechas naranjas.

Para el caso de *monitoreo.html* se hace un enlace a la página llamada *monitoreoz.html*, que está compuesto por dos frames, *sensores.html* con etiqueta de **arriba**, que es usado para Mostar los títulos de la gráfica de resultados, y otro llamado *graficas.html* con etiqueta **abajo**, que es el encargado de hacer referencia al archivo *procesa.php*, el cual lleva a cabo los procesos para graficar los valores.

Estos frames se despliegan en la página de *monitorz.html* y siendo en esta página, donde se incluye una línea de código como ésta: **<meta http-equiv="refresh" content="5" />**, que sirve para actualizar las mediciones que se están recibiendo en el archivo de *procesa.php* cada 2 segundos, y todos estos resultados se muestran tal y como se ve en la figura 6.1

Por último, está el archivo *cosn.php*, que es el encargado de realizar las consultas hacia la base de datos (flechas grises figura 5.26), de alguna fecha en específico y mostrase en la misma ventana.

Capítulo 6

Resultados

6.1 Alcances logrados

Con todo lo planteado a lo largo de este trabajo, es posible implementar el sistema de monitoreo, independientemente de la aplicación u orientación bajo la cual se vaya desarrollando, las facultades y características de este proyecto hacen que sea un diseño que puede ser escalable en cuestión de capacidad, según las necesidades que se presenten para el usuario final.

Con las pruebas realizadas en el sistema propuesto a lo largo de este trabajo, fue posible visualizar los resultados de las mediciones generadas por los sensores usados, además de consultar los registros ingresados a lo largo de una jornada.

En la figura 6.1 se puede ver cómo es la visualización de los resultados generados y representados en las gráficas.

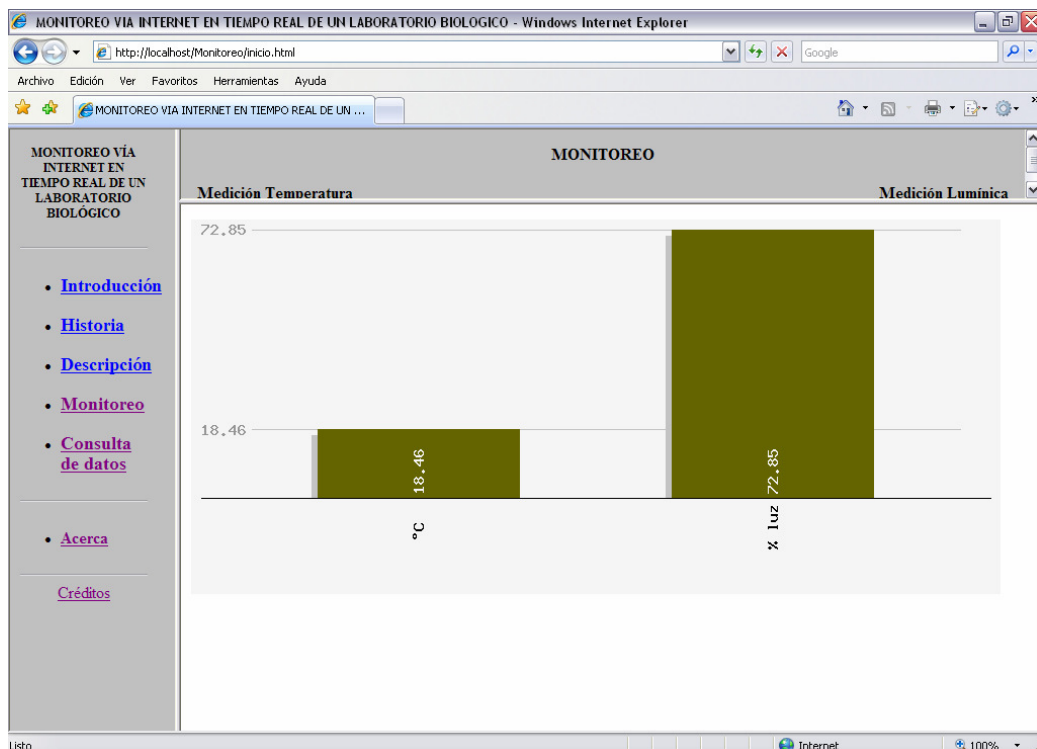


Figura 6.1 Visualización de las mediciones

El formato de la gráfica se presenta de una forma en la que se pueden visualizar los valores de los sensores usados para el monitoreo. La página se actualiza de forma automática para refrescar los datos de las gráficas, y así poder dar el seguimiento de las condiciones y variables que se están monitoreando.

La otra aportación que da este sistema, es la de poder realizar consultas a los registros de los valores medidos previamente, para poder así conocer el comportamiento y dar seguimiento de las variaciones de las condiciones ambientales del laboratorio, esta etapa de consulta se realiza mediante el ingreso de una fecha en específico. En la figura 6.2 se muestra la página en donde se realiza la consulta de alguna fecha en específico.

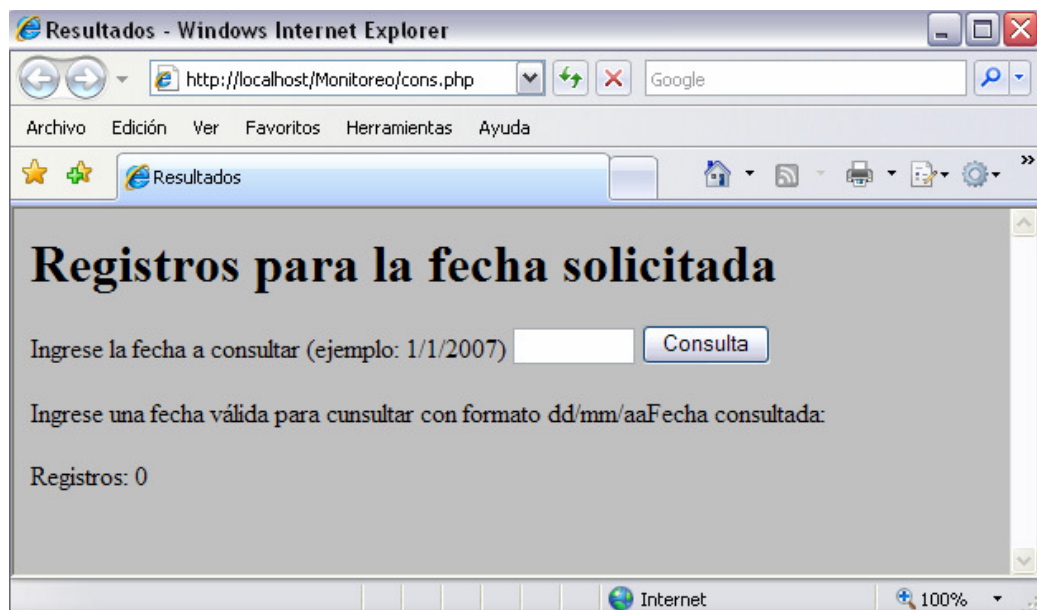


Figura 6.2 Página para la consulta de registros

Si se realiza una consulta para una fecha en específico, se ingresa en el campo y se manda la consulta, y se regresa en esta misma página todos los registros correspondientes a esa fecha ingresada, como se observa en la figura 6.3.

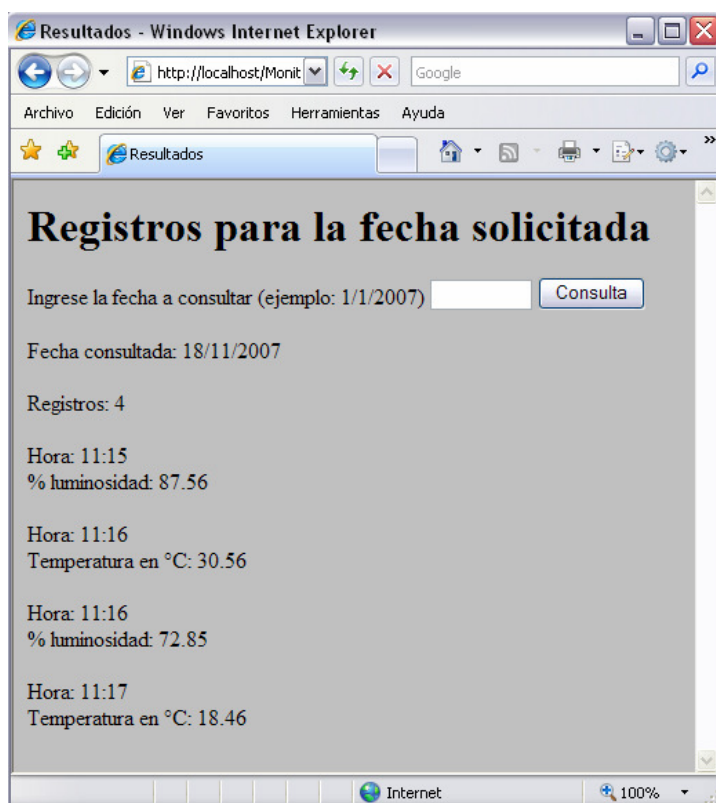


Figura 6.3 Resultados de una consulta

Se puede observar que se muestran la cantidad de registros correspondientes a la fecha a consultar, además de que se obtienen los registros de las últimas mediciones procesadas.

En la figura 6.1, los valores que aparecen en las gráficas corresponden a los últimos valores visualizados en la consulta realizada, una temperatura de **18.46 °C** y un porcentaje de iluminación del **72.85%**, con esto se puede constatar que **las mediciones realizadas se procesan de tal forma que los resultados y las visualizaciones gráficas sean en tiempo real** (visto desde la perspectiva de “ingeniería del software”), con lo que se puede decir que **no hay retardos en todo el proceso de monitoreo del laboratorio**, siendo un sistema eficiente para trabajarse en Internet.

Las áreas que pueden verse beneficiadas con sistemas de monitoreo como el propuesto, son muy variadas y van desde las sencillas como pudiera ser un sistema de calefacción de una habitación, hasta un sistema de monitoreo de

emisiones de GEI (Gases de Efecto Invernadero), donde éste puede ser un proyecto global que podría permitir ver, en qué cantidades se emiten en diferentes países estos gases causantes de fenómenos climáticos y organismos, como el IPCC (Intergovernmental Panel on Climate Change), podrían dar seguimiento y tomar acciones en algunos casos.

El propósito bajo el cual, está diseñado este sistema **Monitoreo vía Internet**, es el de dar la facilidad de poder hacer un seguimiento del comportamiento de un lugar, para este caso se propuso un laboratorio biológico, en donde es de suma importancia conocer el estado y las condiciones bajo las que se encuentran los objetos u organismos que se tiene ahí, de tal manera que dicho seguimiento se haga de forma práctica, rápida y lo más importante, de la forma más eficiente como lo es a través del Internet.

Con las pruebas realizadas de este sistema de monitoreo, este último factor ha funcionado de manera correcta en una red de área local (LAN), por lo que sólo es cuestión de que se configure y se le dé salida hacia Internet, para así expandir la cobertura y sitios disponibles para el monitoreo.

Por lo anterior, se puede decir que se ha cumplido con el objetivo planteado al principio de este trabajo, que es monitorear a través de Internet, las condiciones bajo las cuales se encuentra un laboratorio y las etapas que conforman este sistema de monitoreo.

Algo a destacar en este sistema de monitoreo, es que al estar desarrollado en plataformas Open Source, su implementación no resultaría tan costosa, empezando por que no es necesario pagar por las licencias de su uso, siendo esto un punto muy considerable a favor para este sistema de monitoreo.

6.2 Posibles mejoras

Debido a que constantemente surgen nuevos métodos de programación, nuevos sistemas y dispositivos de adquisición de datos o bien la situación y el entorno bajo el cual se implementa el sistema, es muy posible que surjan modificaciones al sistema inicialmente propuesto en este trabajo. Además de que está presente el caso de hacer más eficiente al sistema de monitoreo, y esto va dirigido en el sentido de que se desea mejorar el rendimiento de la adquisición y procesamiento de datos.

En este caso, posiblemente a futuro, la forma en la que se guardan los datos en la base de datos sea un punto tomar en cuenta para su modificación, ya que la forma que se presenta en este trabajo, se hace uso de un elemento auxiliar como enlace entre los valores medidos y la base de datos, por lo que se contempla la posibilidad de modificar la etapa de adquisición de datos y realizar el enlace directo hacia la base de datos.

Para monitoreo de variables como las propuestas en este sistema, no es muy necesario realizar transferencias y muestreos demasiado rápidos, por ser fenómenos de una naturaleza de cambio paulatino, pero si en algún caso se requiere estar tratando información, cuyos cambios en su comportamiento son demasiado rápidos, se deberá hacer una sustitución de la interfaz, mediante la cual se ingresan los datos a la computadora, esto es, se sustituirá la interfaz para puerto paralelo por una de USB, que es una interfaz que maneja velocidades mucho más altas que las del puerto paralelo, lo cual estaría orientado a aplicaciones en donde las señales generadas por el fenómeno o evento a monitorear, se generan y varían de una manera muy rápida.

Otra posible mejora muy significativa a realizar, es el de la implementación de notificación vía SMS hacia un dispositivo móvil, y poder estar notificando el estado del lugar monitoreado.

6.3 Costos

Además de las propuestas presentadas en este proyecto, la idea de optimización de recursos y medios para la realización del sistema de monitoreo, es necesario estimar el costo neto de la instalación y adecuación del sistema, de acuerdo a las necesidades del cliente.

En este caso, se utilizaron dos sensores para representar el funcionamiento del proceso de monitoreo, un sensor de temperatura y una fotocelda que representa para este caso al sensor de luz.

La circuitería usada para trabajar con los sensores resulta ser práctica, por lo que los costos correspondientes a hardware, no son tan excesivos.

A continuación se presenta una relación de costos de materiales para el sistema de monitoreo presentado en este proyecto.

<i>Material</i>	<i>Costo unitario (pesos mexicanos)</i>
Sensor temperatura	\$20
Celda solar	\$70
UA 741 (x4)	\$4
Cable	\$10
Potenciómetros (x4)	\$8
Placa para montaje del circuito	\$15
Microcontrolador HCS08	\$49
Cable para puerto paralelo	\$35
	TOTAL \$247

Aunado a lo anterior se debe considerar la parte de programación del software. Para estos casos se suele considerar la cantidad de líneas de código usadas para alguna función.

Los lenguajes usados para programar las aplicaciones que se llevan a cabo en el proceso de monitoreo fueron:

- C#
- PHP

En el mercado de los precios de lenguajes de programación se obtienen los costos totales de un desarrollo en base a la cantidad de líneas (o kilo líneas) de código.

Para el caso de c# la cantidad aproximada de líneas de código fue de 300 kilo líneas.

El costo aproximado de una kilo línea es de cerca de \$2 uds, que en pesos mexicanos es cerca de \$23 pesos, siendo el costo neto de \$6900.

Para el caso de PHP, fueron cerca de 180 kilo líneas dando un costo de \$4140 pesos.

El costo total de esta implementación sería de \$11 287 pesos, siendo un precio que si se compara con otros sistemas de monitoreo es realmente una buena alternativa.

Algo que cabe destacar, es que este sistema al estar desarrollado con lenguajes que son libres (open surce), al momento de llevarse con un usuario final, no tiene que pagar por licencias o permisos de uso.

6.4 Conclusiones y recomendaciones

Se logró visualizar a través de una página Web el comportamiento de las variables sensadas, además de poder consultar aquellas mediciones realizadas en una fecha determinada. De igual forma se logró observar dicho comportamiento localmente.

Nos fue posible también comprobar que a pesar de que la interfaz de puerto paralelo pudiera ser algo obsoleta en cuanto a uso, para aplicaciones de este tipo ha resultado ser de mucha utilidad, aunado a que el comportamiento de fenómenos como la temperatura presentan cambios poco perceptibles al ser humano.

Se realizaron pruebas satisfactorias, detalladas anteriormente (ver tema 6.1), dichas pruebas fueron comprobadas, vía Internet, vía red de área local y localmente. Con esto, es un hecho que se lograron los objetivos planteados para este proyecto. Esto no implica las mejoras al proyecto, previamente expuestas.

En la actualidad, la mayoría de las actividades que se realizan como puede ser la compra de algún artículo, la comunicación entre dos personas, la búsqueda de información o el trámite de algún documento, se realiza vía Internet, es por ello que se planteó el desarrollo de este proyecto, de tal manera que no solo en la computadora que se encarga de procesar los datos de las mediciones se observe el resultado de éstas, sino que se da la posibilidad a esta aplicación para que se pueda consultar desde cualquier parte del mundo gracias al Internet.

Este tipo de proyectos pueden ser de gran utilidad, por las características que presenta, las recomendaciones que hacemos en el supuesto caso que se llegase a retomar los lineamientos y planteamientos hechos aquí, son que se sigan las bases presentadas y lo que concierne a la parte de visualización de resultados, puede ser tratada por algún diseñador que le de una mejor presentación al sitio Web.

Cabe mencionar que este sistema de monitoreo es posible implementarlo en otras plataformas de sistema operativo, como puede ser el caso de Linux, pero se decidió montar bajo Windows debido a que es un sistema operativo muy usado y más fácil de manejar y aunado a que se dirigió a usuarios finales, se trata de que sea lo más sencillo y amigable para ellos.

La posibilidad de dar más alcance a este proyecto consiste en hacer uso de un microcontrolador de más capacidad, en el sentido de que se cuente con un mayor número de canales ADC para el tratamiento de las señales analógicas, por eso recomendamos el uso del microcontrolador HCS12 que cuenta con 16 canales para este propósito.

Para finalizar esperamos que este proyecto presentado en este trabajo sea comercializable a un mediano plazo, y de esta forma poder contribuir en los diferentes sectores industriales o de cualquier otra índole a que mejoren sus métodos de monitoreo, esto gracias a que la adaptabilidad y eficacia de este proyecto puede ser implementado a diversos ámbitos, teniendo un carácter multipropósito en distintas áreas, mismas donde se requiera un monitoreo, como el que se ha planteado, propuesto y se ha llevado a cabo satisfactoriamente.

Bibliografía

REFERENCIAS BIBLIOGRÁFICAS:

- [1] Boylestad Nashelsky
Electronica Teoria de Circuitos y Dispositivos Electrónicos
Prentice Hall
- [2] Floyd
Fundamentos de Electrónica Digital
Limusa
- [3] Ceballos Fco. Javier
Microsoft C# Curso de Programación
Alfa Omega
- [4] Alexander Aulbach
Manual de Referencia de PHP
- [5] Steve Suehring
MySql Bible

DOCUMENTACION WEB:

- [1] Documentación oficial PHP
www.php.net
- [2] Documentación oficial MySql
www.mysql.com
- [3] Documentación interfase puerto paralelo
www.modelo.edu.mx/univ/virtech/circuito/paralelo.htm
<http://mimosa.pntic.mec.es/~flarrosa/puerto.pdf>