

# UACM

Universidad Autónoma  
de la Ciudad de México

*Nada humano me es ajeno*

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELCTRÓNICOS Y  
DE TELECOMUNICACIONES

**Dispositivo electrónico para monitoreo a distancia a temperaturas  
del agua en tratamiento por luz UV**

TRABAJAJO RECEPTACIONAL  
PARA OBTENER EL TITULO DE  
LICENCIADA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE  
TELECOMUNICACIONES

PRESENTA

**YADHIRA OSIRIS DE LA ROSA MUNDO**

Director del trabajo recepcional

**Dr. José Lorenzo Calzado López**

Codirector

**M. en I. Omar Nieto Crisóstomo**

México, D.F. Marzo 2015.

## SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



## UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

### RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

#### DERECHOS RESERVADOS<sup>©</sup>

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

## Agradecimientos

A ti madre Yolanda Mundo Texocotitla, que me diste la más entrañable de las posesiones: la vida, te doy las gracias por creer siempre en mí, por tenerme la confianza, tolerancia, respeto y la paciencia, eres una de las mujeres que más admiro, respeto y quiero. Te agradezco que siempre me acompañes en todos los momentos difíciles de mi vida gracias. Recibe de parte de tu hija un pequeño homenaje por medio de este trabajo.

A ti padre Marcos Eduardo De La Rosa Oropeza, cumplí con lo te prometí terminar la carrera y mi trabajo recepcional, gratifico tus consejos y agradezco que me hallas inculcado las ganas de superarme día con día de manera intelectual, gracias por apoyarme espero demostrarte mi crecimiento y satisfacer las expectativas que tienes sobre mi te quiero mucho papá.

A mi hermana Karen Jacquelyn De La Rosa Mundo, eres una persona muy importante con la cual viví experiencias buenas y difíciles, juntas hemos vencido obstáculos ya que jamás nos hemos dado por vencidas hoy en día cada quien escogió su camino pero siempre buscando superarnos te deseo mucho éxito y confié mucho en ti te quiero mucho al igual que Alexa una personita preciosa que cambió el rumbo de tu vida y la de nuestra familia.

A ti mi pequeña hermanita Paola Michelle De La Rosa Mundo, gracias por tus consejos, tus palabras de aliento y tus valores, siempre apoyándome en cada una de mis travesías tanto buenas como malas, deseo para ti éxito y que nunca te des por vencida yo sé que podrás culminar tu carrera con méritos confié en ti y te deseo lo mejor te quiero mucho.

A mi Abuelita Celia Oropeza Vázquez, te agradezco que me inculcaras la persistencia, esta fue fundamental para terminar mi carrera profesional, eres un ejemplo de lucha, preparación y superación, gracias por tus consejos te quiero.

Al profesor M. en I. Omar Nieto Crisóstomo, agradezco mucho su paciencia, tolerancia, tiempo y esfuerzo, ya que contribuyó en gran parte que culminara mi trabajo recepcional gracias.

Al profesor Dr.Huitzilín Yépez Martínez, le agradezco todo el apoyo, durante toda la carrera, por sus consejos, por la confianza y por siempre creer en mí. Fue pieza importante para este trayecto espero haber cumplido con las expectativas que tenía de sobre mí, gracias por brindarme las enseñanzas académicas y por ser una persona profesional.

Al Profesor Dr. José Lorenzo Calzado López, por el tiempo, el apoyo y por proponerme el proyecto y por los aportes metodológicos que realizó a lo largo del trabajo recepcional.

A los lectores, gracias por brindarme su tiempo para leer mi trabajo recepcional, y por las observaciones realizadas sobre el mismo, esto de acuerdo a su experiencia.

A la Universidad Autónoma de la Ciudad de México, Gracias a esta institución logre cumplir con mi formación académica, misma que el día de mañana me permitirá defenderme y participar en la resolución de problemas que atañen a la ciudad y así lograr mejores condiciones para la humanidad.

Agradezco a la UACM por el apoyo recibido para la impresión y empastado de mi trabajo recepcional.

A los profesores de la UACM, les doy gracias por sus enseñanzas impartidas en las aulas de la UACM, ya que esto ayudo a obtener los conocimientos necesarios para culminar mi formación académica.

A mis compañeros, agradezco el apoyo de algunos de mis compañeros que estuvieron conmigo durante la carrera, por sus palabras y sustento intelectual.

Para una persona especial, gracias por apoyarme en la parte final de mi carrera, por la motivación, el entusiasmo, los consejos, la confianza y sobre todo por las lecciones de vida que me han permitido crecer como persona.

*Yadhira Osiris De La Rosa Mundo*

*Nada humano me es ajeno*

## Tabla de contenido

CAPÍTULO 1.....	10
1 Introducción.....	11
1.1 Justificación.....	12
1.2 Objetivo central del trabajo recepcional.....	12
1.2.1 Objetivos Particulares.....	13
1.3 Requerimientos de diseño.....	13
CAPÍTULO 2.....	15
2 Antecedentes y Marco Teórico.....	16
2.1 Importancia del agua.....	16
2.2 Características del agua para consumo humano.....	17
2.3 Esterilización de agua por medio de luz ultravioleta.....	18
2.4 Planta Piloto de Potabilización de Agua.....	20
2.5 Reactor Ultravioleta (UV).....	22
2.6 Elementos del sistema.....	22
2.7 Microprocesadores y microcontroladores.....	23
2.7.1 Fabricantes de microcontroladores.....	24
2.8 Comunicación entre dispositivos.....	25
2.8.1 Comunicación alámbrica.....	25
2.8.2 Ethernet.....	26
2.9 Capas ISO/OSI.....	27
2.9.1 Capa física.....	28
2.9.2 Capa de enlace de datos.....	28
2.9.3 Capa de red.....	29
2.9.3.1 Direcciones IP.....	29
2.9.4 Capa de transporte.....	31
2.9.4.1 Protocolos de comunicación.....	32
2.9.5 Capa de sesión.....	32
2.9.6 Capa de presentación.....	33
2.9.7 Capa de aplicación.....	33
2.9.8 Diferencias entre el modelo OSI y TCP.....	33
2.10 Protocolos utilizados en la pila TCP/IP de Microchip.....	34
2.11 Redes informáticas.....	36
2.12 Protocolo de Configuración Dinámica de Host (DHCP).....	37
2.13 Modelo Cliente-Servidor.....	37

2.13.1	Cliente .....	38
2.13.2	Servidor.....	39
2.13.3	Php.....	39
2.13.4	Mysql .....	40
2.13.5	Crear un usuario de Mysql.....	41
2.14	Phpmyadmin .....	41
2.15	Socket.....	42
2.15.1	Selección del puerto.....	43
2.16	Protocolo http.....	44
2.17	Características del router Linksys Cisco.....	45
CAPÍTULO 3.....		48
3	Análisis y diseño .....	49
3.1	Ethernet start kit PIC32MX795F512L.....	49
3.1.1	Características del PIC32MX795F512L.....	50
3.1.2	Ethernet starter kit .....	51
3.1.3	Dauther card o tarjeta de expansión .....	52
3.1.4	Conectar un microcontrolador a una red Ethernet .....	53
3.1.5	Módulo ADC .....	54
3.1.6	Descripción del timer.....	55
3.1.7	Pantalla de cristal líquido (LCD).....	57
3.1.8	Stack TCP/IP Microchip.....	58
3.2	Programación del socket.....	60
3.2.1	Explicación del protocolo TCP en la PILA .....	60
3.3	Pasos para construir un proyecto con el stack TCP/IP .....	62
3.3.1	Inicialización de la pila para socket TCP .....	63
3.3.2	Función de inicio.....	64
3.3.3	Multitarea Cooperativa.....	65
CAPÍTULO 4.....		66
4	Desarrollo e Implementación de las interfaces del sistema.....	67
4.1	Módulo ADC .....	67
4.1.1	Muestreo de la señal de temperatura.....	69
4.2	Librerías para la pantalla LCD.....	70
4.3	Programación del timer.....	71
4.4	Programación de la PIC 32.....	72
4.5	Librerías de stack Microchip .....	73

4.5.1	HWP PIC32_ETH_ETH795.h.....	74
4.5.2	TCPIP ETH795.h.....	74
4.5.3	HardaweProfile.h.....	75
4.5.4	GenericTCPClient.c.....	75
4.5.5	Main.h.....	75
4.6	Realización del SOCKET TCP.....	76
4.7	Diagrama de flujo para el proyecto.....	77
4.8	Creación del proyecto.....	78
4.9	Http 1.1 para la transferencia y captura en base de datos.....	80
4.10	Creación de la red LAN.....	81
4.10.1	Configuración del router Cisco.....	83
4.11	Montaje del Servidor distribución Linux (Ubuntu 10.04 LT).....	85
4.11.1	Configuración del servidor.....	86
4.11.2	Mysql.....	87
4.11.3	Base de datos.....	88
4.12	Script en el servidor.....	91
4.13	Configuración de correo electrónico.....	93
CAPÍTULO 5.....		94
5	Diseño de la interfaz del sensor de temperatura.....	95
5.1	Características del sensor LM35.....	95
5.1.1	Comportamiento lineal.....	95
5.1.2	Diseño del amplificador no inversor.....	96
5.1.3	Comportamiento lineal del sensor en el osciloscopio.....	97
5.2	Realización de las placas de experimentación (Press-N-Pee).....	98
5.2.1	Creación de la fuente de voltaje DC.....	99
CAPÍTULO 6.....		102
6	Resultados.....	103
6.1	Sistema completo.....	103
6.2	Interface con el Microcontrolador.....	104
6.3	Tabla de resultados de las temperaturas.....	105
6.3.1	Página WEB.....	106
6.4	Correo electrónico.....	107
6.5	Costo del proyecto.....	108
Conclusiones.....		109
Anexos.....		111

Glosario.....	117
Bibliografía.....	118

## Índice de Figuras

Figura 1. Bosquejo general.....	14
Figura 2. Esquema hidráulico de la planta piloto.....	20
Figura 3. Esquema del fotocatalizador propuesto.....	22
Figura 4. Conexión Ethernet.....	26
Figura 5. Esquema del modelo OSI.....	27
Figura 6. Comparativo de los Modelos TCP/IP y OSI.....	27
Figura 7. Clases de direcciones IP.....	30
Figura 8. Protocolos por cada capa de TCP.....	32
Figura 9. Señal de sincronía para establecer conexión.....	33
Figura 10. Jerarquización de la pila TCP/IP.....	35
Figura 11. Bosquejo del modelo cliente/servidor.....	38
Figura 12. Características del PHP instalado.....	40
Figura 13. Comprobación de la instalación de phpmyadmin,.....	42
Figura 14. Método http 1.1.....	45
Figura 15. Sistema de diseño del problema.....	49
Figura 16. Diagrama de pines del PIC usado.....	51
Figura 17. Componentes de la tarjeta.....	52
Figura 18. Descripción tarjeta de expansión.....	53
Figura 19. PIC 32 con la tarjeta de expansión.....	53
Figura 20. Diagrama de bloques del ADC.....	55
Figura 21. Pantalla LCD.....	57
Figura 22. Diagrama de conexión de la LCD.....	58
Figura 23. Stack para el microcontrolador.....	59
Figura 24. Diagrama del socket TCP.....	61
Figura 25. Diagrama de bloques del convertidor analógico-digital.....	67
Figura 26. Diagrama que muestra las entradas del convertidor.....	68
Figura 27. Configuración de los bits de cada paquete de registros.....	69
Figura 28. Método de cuantización.....	69
Figura 29. Resultado de la conexión de la LCD.....	71
Figura 30. Temporizador de tipo B (32 bits).....	71
Figura 31. Respuesta de un ping para comprobar la asignación de IP.....	75
Figura 32. Diagrama de flujo del proyecto.....	78
Figura 33. Creación del proyecto.....	80
Figura 34. Esquema de la topología implementada.....	82
Figura 35. Tabla de enrutamiento de router.....	83
Figura 36. Router físico.....	83
Figura 37. Muestra el estado del enrutador.....	83
Figura 38. Configuración avanzada.....	84
Figura 39. Parámetros de la red.....	84

Figura 40. Parámetros de la red. ....	85
Figura 41. Seguridad WAP. ....	85
Figura 42. Descripción de la tarjeta de red. ....	86
Figura 43. Respuesta del servidor. ....	86
Figura 44. Respuesta del servidor. ....	87
Figura 45. Bases de datos en Mysql. ....	88
Figura 46. Comando use. ....	89
Figura 47. Comando describe. ....	89
Figura 48. Comando select*from. ....	89
Figura 49. Comando donde se selecciona. ....	90
Figura 50. Script para la parte de conexión. ....	91
Figura 51. Script para la parte de recibe datos. ....	91
Figura 52. Script para la parte de recibe datos. ....	92
Figura 53. Representación del LM35. ....	95
Figura 54. Diagrama esquemático del amplificador NO-inversor. ....	96
Figura 55. Respuesta en el osciloscopio. ....	97
Figura 56. Diagramas con Proteus. ....	98
Figura 57. Muestra el proceso camino y grabado en la placa de baquelita. ....	98
Figura 58. Circuito final. ....	99
Figura 59. Fuente de voltaje alimentación. ....	99
Figura 60. Diagrama de pistas de la fuente ....	100
Figura 61. Placa y circuitos finales para la fuente. ....	100
Figura 62. Respuesta de la fuente de voltaje. ....	100
Figura 63. Circuito Final. ....	104
Figura 64. Etapas del microcontrolador. ....	104
Figura 65. Respuesta del sensor. ....	105
Figura 66. Resultado de la base de datos. ....	106
Figura 67. Página Web que observara el usuario. ....	106
Figura 68. Respuesta en el correo. ....	107

## Índice de Tablas

Tabla 1. Componentes de la planta potabilizadora UACM-SLT. ....	22
Tabla 2. Descripción de los microcontroladores de Microchip. ....	25
Tabla 3. Máscaras de las clases de redes. ....	31
Tabla 4. Desventaja y ventajas del modelo TCP/IP. ....	34
Tabla 5. Instrucciones para implementar en el socket TCP. ....	43
Tabla 6. Clasificación de puertos. ....	44
Tabla 7. Características del Router. ....	46
Tabla 8. Características del microcontrolador. ....	50
Tabla 9. Tabla de puertos entradas y salidas. ....	51
Tabla 10. Proporciona las características de las funciones de temporizador. ....	56
Tabla 11. Tabla de conexiones de la LCD. ....	58
Tabla 12. Descripción de las bibliotecas de stack que se utilizaran. ....	60

Tabla 13. Descripción de las funciones (.h).....	63
Tabla 14. Ejemplos de las librerías del stack.....	73
Tabla 15. Ventajas y Desventajas. ....	82
Tabla 16. Muestra los puertos abiertos.....	87
Tabla 17. Mediciones de temperatura para comprobar linealidad. ....	97
Tabla 18. Valores máximos y mínimos de la fuente.....	101
Tabla 19. Precios de los elementos del sistema. ....	108

# CAPÍTULO 1

*Introducción y justificación del problema.*

# 1 Introducción

En la actualidad existen un gran número sistemas de comunicación vía remota, los cuales nos proporcionan diferentes tipos de servicios, además de son implementados en diferentes entornos, es decir en grandes o medianas empresas para fines específicos. Existen múltiples alternativas de solución, dentro de la cuales tenemos a la LAVIEW, la cual es una plataforma diseño para este tipo de sistemas ya que ofrece librerías de funciones dedicadas a la adquisición de datos, la instrumentación, al análisis matemático de las mediciones y la visualización. LabVIEW se dedica especialmente a los bancos de pruebas y mediciones para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente pero de gran costo. Es por ello que en este trabajo recepcional, se propuso una solución similar; pero bajo otras condiciones tanto en costo como en diseño. Dándonos como resultado un sistema de comunicación vía remota Ethernet a través un microcontrolador.

Este producto final, permitirá resolver una de las necesidades que se tiene en la planta purificadora de agua de la UACM-SLT. Con ayuda de los productos y herramientas de Microchip y sus productos se desarrolló un software que efectuara el monitoreo remoto del sensor de temperatura. Esta aplicación consta de una tarjeta de control PIC32MX, con un módulo Ethernet programado en el ambiente de desarrollo MPLABX utilizando la pila TCP/IP.

Se diseñó e implementó un sistema de comunicación para acceder a información digital generada en forma remota. Como fuente de información se tiene un microcontrolador de 32 bits con el cual se hará el monitoreo, en el caso de esta tesis el proceso es la desinfección de agua por luz UV. El sistema desarrollado consiste en una red LAN, la cual cuenta con un router Cisco-Linksys, un servidor Linux (Ubuntu), además de la interfaz web con el usuario. La programación de la interfaz cliente socket-servidor, se llevo por medio del Stack Microchip.

## 1.1 Justificación

La tesis que se presenta se relaciona con dos proyectos que se desarrollan en la planta y laboratorio de agua en el plantel San Lorenzo Tezonco de la UACM y de los cuales se dará un breve resumen.

- Proyecto “Aplicaciones electrónicas en la potabilización de agua”, que consta de dos partes: por un lado, automatizar en la planta potabilizadora de agua y, por el otro, tener control remoto de dispositivos y análisis involucrados en experimentos sobre tratamiento de agua en los cuales es necesario medir parámetros físicos, químicos y biológicos *in situ*. Este proyecto tiene varias motivaciones, siendo la ubicación del laboratorio una de ellas, ya que se encuentra en una zona de servicios administrativos, alejada de la zona de aulas y oficinas de profesores, además de tener un acceso complicado. Al principio para ir al laboratorio era necesario salir del plantel actualmente el acceso es por el interior del plantel por un camino improvisado y aislado.
- Proyecto “Construcción de foto reactores para potabilización de agua”, tiene como objetivo la construcción de un contenedor de agua para desinfección con luz ultravioleta. Estos contenedores reciben el nombre genérico de foto reactores. Durante las pruebas de operación del fotoreactor, dos parámetros físicos son necesarios monitorear: intensidad de luz UV y temperatura. Una vez construido un fotoreactor con estos parámetros dentro de los rangos apropiados, se evaluará su capacidad de desinfección por análisis *in situ* de otros parámetros físicos, químicos y biológicos del agua.

## 1.2 Objetivo central del trabajo recepcional

El objetivo central de esta tesis es establecer una plataforma de comunicación para el monitoreo de los parámetros de medición del laboratorio. En caso particular, se ha elegido a la temperatura en un fotoreactor Ultravioleta (UV) como parámetro de prueba, sin embargo en la planta se cuenta con otras medidas

importantes tales como intensidad de UV, humedad, turbiedad, concentraciones de iones, pH, etc. Mismas que están fuera del alcance de esta tesis. El objetivo a largo plazo es controlar, ejecutar y monitorear diferentes experimentos sin necesidad de estar presente en la planta.

### 1.2.1 **Objetivos Particulares**

- Diseñar y programar el software del sistema inmerso de 32 en lenguaje C.
- Diseño y construcción de la base de datos.
- Montaje del servidor.
- Diseño de la página WEB.
- Creación del correo electrónico.

### 1.3 **Requerimientos de diseño**

Como requisitos para la plataforma de comunicación se tiene la siguiente lista de requerimientos:

- Capacidad de transmitir información de muy alto orden.
- Múltiples plataformas para acceder y procesar la información.
- Se contará con un nodo de telecomunicaciones en el laboratorio.

Se decidió partir de un microcontrolador, o PIC, debido a que los objetivos de los proyectos involucran la automatización de procesos de la planta potabilizadora, automatización que se proyecta hacer mediante PIC. La decisión de usar un PIC de 32 bits fue académica, ya que tanto los profesores involucrados con el laboratorio del agua, como esta tesista, no contaban con experiencia con esta familia de PIC, además de que se contaba con dicho dispositivo lo que ahorro costos en el proyecto. Se resolvió una plataforma Ethernet debido a que se contará en el laboratorio con un nodo de la red de telecomunicaciones interna de la UACM.

La toma de muestras de temperatura del agua se realizara cada minuto, una de las razones por las cuales se estableció esto fue por el hecho de que se colocaran en una base de datos, mismo tiempo que se consideró pertinente para no tener

una cantidad considerable de muestras, además que en este tiempo se contempla que la temperatura del agua no cambie de manera abrupta esto debido a la masa de agua con la que se cuenta. Por lo tanto es un tiempo razonable.

Debido a que el reactor es de plástico, sus paredes son superficies propicias para el crecimiento de colonias de bacterias. Es por esto que se debe tener un control de la temperatura del agua dentro del reactor. Por esta razón es que se requiere poner una alarma a cierta temperatura. Los experimentos realizados hasta ahora, muestran que cuando el agua alcanza los 55°C, se miden más bacterias que las que entraron. Por tal motivo que se solicita que la alarma se accione a los 55°C.

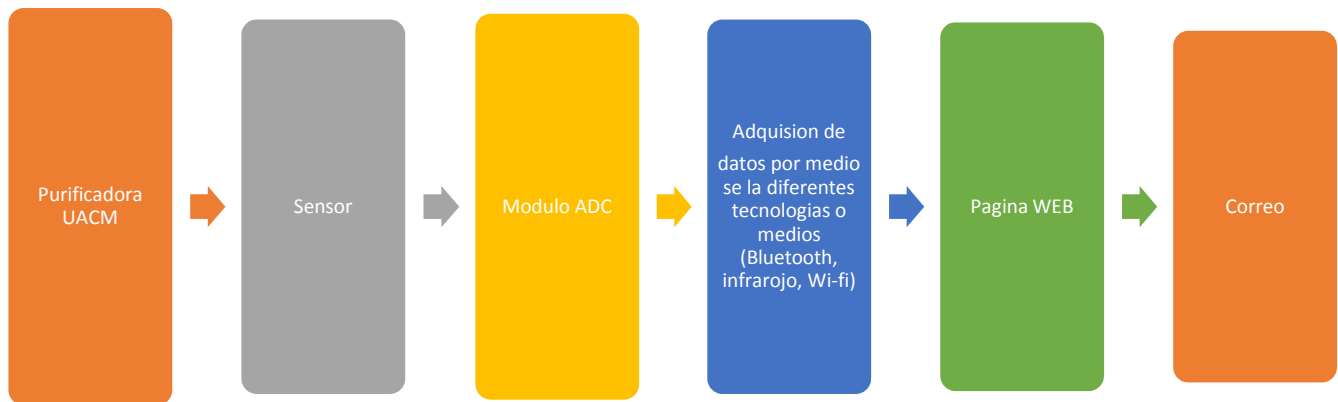


Figura 1. Bosquejo general.

En la figura 1 se muestra el sistema en forma general todo el sistema en conjunto. En el capítulo 2 se hace una investigación teórica de lo relacionado con estos bloques, mientras que en el capítulo 3 se hace una descripción detallada de los dispositivos, protocolos y tecnologías usadas en la tesis. Finalmente, en el capítulo 4 se describe la implementación del sistema.

# CAPÍTULO 2

*Antecedentes y Marco  
Teórico.*

## 2 Antecedentes y Marco Teórico

En este capítulo se explicaran algunos conceptos teóricos de cada uno de los elementos a utilizar durante todo el proyecto con la finalidad de conocer sus principales características.

### 2.1 Importancia del agua

El agua es un compuesto químico formado por dos partes de hidrógeno y una parte de oxígeno, en volumen. Puede tener en solución o en suspensión a otros materiales sólidos, líquidos o gaseosos. Su fórmula es  $H_2O$ , es un líquido incoloro, inodoro e insípido. A temperaturas elevadas,  $1500[^\circ C]$ , se descompone en hidrógeno y oxígeno. Por su elevada constante dieléctrica puede disociar y descomponer numerosas materias. Por su punto de fusión y de evaporación específicamente elevado, el agua es buena acumuladora del calor de la naturaleza y equilibra las fluctuaciones térmicas [1].

El agua es un compuesto que tiene propiedades muy peculiares, una de ellas es su gran poder de disolver, se le ha llamado "El solvente Universal", es por ello que casi nunca encontramos una agua "Pura". Normalmente el agua se clasifica según su origen y las sustancias que tiene en solución. El agua es un constituyente necesario de todas las células, animales y vegetales y la vida no puede existir en su ausencia.

El agua se clasifica según el uso que se le vaya a dar:

- Consumo humano o potable.
- Limpieza y lavado.
- Uso industrial.
- Generadores de vapor de alta y baja presión.
- Enfriamiento.
- Análisis químicos.
- Aplicaciones biológicas (libre de pirógenos).

Cada una de ellas tiene en forma disuelta, suspendida o coloidal, diversas sales minerales y gases en cantidades variables dependiendo de dónde procedan.

El agua tal como existe, pocas veces se puede usar en su forma natural, se requiere conocer sus características físicas, químicas, su naturaleza y cantidad de las sustancias disueltas o suspendidas que contengan, para acondicionar el agua particular al uso deseado.

## **2.2 Características del agua para consumo humano**

El agua para consumo humano es aquella que debe ser apta para la bebida e higiene y debe estar libre de organismos patógenos, concentraciones químicas, impurezas y de cualquier tipo de contaminación que cause problemas para la salud humana.

Por esta razón es indispensable asegurarse de la buena calidad del agua, factor determinante del estado de salud de una comunidad. Cuando el agua se encuentra contaminada, es imperativo o recomendable realizar algún tipo de tratamiento mediante un proceso que la transforme en agua segura para bebida. La Norma Oficial Mexicana NOM-127-SSA1-1994 establece los límites permisibles de calidad y los tratamientos de potabilización del agua para uso y consumo humanos, que deben cumplir los sistemas de abastecimiento públicos y privados o cualquier persona física o moral que la distribuya, en todo el territorio nacional. La calidad del agua se mide en términos de sus características físicas, químicas y biológicas.

**Características físicas:** Hacen referencia al olor, sabor, color y turbiedad.

**Características químicas:** Estas hacen referencia al contenido de minerales como el hierro (Fe) y el manganeso (Mn), y a otras sustancias que la hacen fácilmente identificables por su efecto sobre la ropa, ya que generalmente los contaminantes impide la disolución del jabón, como en el caso de alta presencia de carbonatos de calcio.

**Características biológicas:** El término biológico hace referencia a la presencia de organismos patógenos, como huevos, quistes, bacterias y virus, que se

encuentran presentes en el excremento humano, en la basura, en el agua estancada y en suelos contaminados con excrementos del hombre y los animales.

Aunque el agua se vea limpia, es necesario realizar un tratamiento que elimine los organismos patógenos, que no se ven a simple vista y que pueden causar enfermedades a las personas que consuman agua sin tratar.

- Pruebas utilizados para el control sanitario del agua

Las características físicas, químicas o biológicas del agua se pueden determinar mediante exámenes de laboratorio o inspecciones de terreno que permiten conocer si el agua está apta para su consumo desde el punto de vista sanitario, los principales exámenes que se realizan habitualmente son:

**Físicos:** Temperatura, turbiedad, color, olor y sabor.

**Químicos:** Relacionados con la potabilidad (pH, alcalinidad, acidez, dureza, etc.) Indicadores de contaminación (nitrógeno, oxígeno consumido, cloruros).

**Bacteriológicos:** Índice de Coliformes (totales o fecales).

### 2.3 Esterilización de agua por medio de luz ultravioleta

El término luz ultravioleta, como se dice comúnmente, es un método probado para tratar agua contaminada microbiológicamente; dicha contaminación es producida por microorganismos (parásitos, bacterias o virus). Se produce principalmente por la presencias de fenoles bacterias, virus, protozoos y algas unicelulares. Esta tecnología simple y segura es recomendable para flujos bajos en sistemas residenciales, para flujos comerciales de proyectos industriales. La desinfección en su sentido literal, significa libre de infección. La EPA o agencia de protección ambiental de Estados Unidos y la Organización Mundial de la Salud (OMS) define la desinfección del agua como la ausencia del indicador de bacterias coliformes. Por otro lado la esterilización implica la completa destrucción de todas las formas de vida. Para usos prácticos el término esterilizador es usado como un término general para describir la tecnología ultravioleta. La luz ultravioleta es una región de energía del espectro electromagnético que cae entre los rayos-X y la visible. La

UV por sí misma cae dentro de los rangos de 200 nanómetros (1 nanómetro  $\text{nm}=10^{-9}$  metros) hasta los 390 nanómetros. Como los niveles de energía aumentan al incrementar la longitud de onda, los rayos-X tienen más energía que los rayos ultravioleta y la luz UV tiene más energía que el espectro de luz visible. El espectro de UV está dividido en 4 regiones, que son designadas como sigue: Vacío, UV-A, UV-B, y UV-C.

La producción de luz ultravioleta puede lograrse a través de la conversión de energía eléctrica. Esta conversión se logra con una lámpara de vapor de mercurio de baja presión. La luz UV se produce como resultado del flujo de electrones a través del vapor de mercurio ionizado entre los electrodos de la lámpara. Debemos hacer notar que el brillo azulado que da la lámpara es debido al gas inicial dentro de la lámpara y no tiene ningún efecto germicida por sí solo. Estas lámparas son similares en diseño a las lámparas fluorescentes con algunas excepciones básicas. Las lámparas UV son producidas de cuarzo duro, en oposición al vidrio suave que encontramos en las lámparas fluorescentes. Este tubo permite una transmisión de UV por arriba del 90% de la energía radiada. Las lámparas fluorescentes también contienen un recubrimiento de fósforo dentro de la lámpara que convierten a la UV en luz visible.

Los microorganismos pueden agruparse en 5 grupos básicos: bacterias, virus, hongos, protozoarios y algas. En términos sencillos los microorganismos están formados por una pared celular, una membrana llamada citoplasma y el material genético de la célula así como el ácido nucleico. En este material genético o ADN (ácido desoxirribonucleico) el que ataca la UV. *Al penetrar la UV a través de la pared celular y el citoplasma, causa un reordenamiento molecular del ADN del microorganismo que hace imposible que se reproduzca.* Cada célula usa distintos mecanismos de defensa para esconderse, también se utilizaron distintos niveles de energía UV para su destrucción. Las lámparas UV emiten aproximadamente 90% de su energía irradiada a 253.7 [nm], que está muy cerca del pico germicida efectivo de 265 nm. El grado de destrucción microbiana es producto del tiempo de radiación, que es la residencia actual, o tiempo de contrato del agua en la cámara

de esterilización, y de la intensidad que es la cantidad de energía por unidad de área.

Los tubos de cuarzo sellan el flujo de agua hacia la lámpara, ofrecen temperaturas de operación más uniformes y permiten mayor transmisibilidad de energía en el agua. La variedad de aditamentos opcionales que pueden ser construidos con el esterilizador incluyen: aparatos de monitoreo de UV. A 253.7 [nm], controles de flujo para limitar el flujo en las unidades, alarmas visuales y auditivas (locales y remotas) para dar aviso de fallas de lámparas, sensores de temperatura alta para monitorear temperatura excesiva en la cámara del reactor o en el panel de control, un contador de horas para monitorear el tiempo de trabajo de la lámpara UV etc.

## 2.4 Planta Piloto de Potabilización de Agua

El objetivo de la planta es experimental y se aspira a dar servicio a la comunidad académica que desee trabajar en el tema del agua potable. Eventualmente podría transformarse en una planta de servicio que surta de agua al plantel. La figura 2 muestra un esquema general.

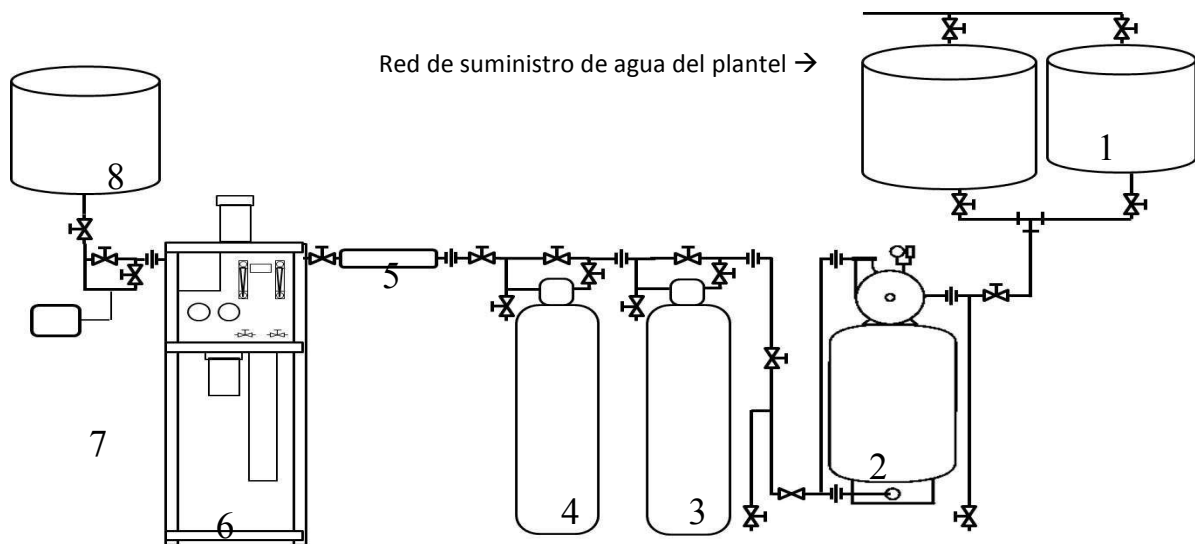


Figura 2. Esquema hidráulico de la planta piloto.

Se desea que la planta sea flexible y pueda configurarse según los experimentos a desarrollar. En particular, el área de investigación bajo la cual fue ideada la planta

se relaciona con la construcción de filtros, u otros equipos de purificación, con materiales y técnicas al alcance de las comunidades de la ciudad. Se espera que en el futuro diversos grupos académicos hagan uso de la planta.

Esta planta se ha proyectado considerando que el agua a tratar se tomará de la cisterna que proviene regularmente de pozos de almacenamiento de agua potable para consumo humano en la colonia Lomas de San Lorenzo ubicada en la delegación Iztapalapa del Distrito Federal. Sin embargo, el agua de la cisterna ocasionalmente es surtida por pipas de agua con origen diverso.

La Figura 2 muestra la instalación hidráulica en la planta configurada para producir agua desionizada. La tabla 1 muestra los componentes adquiridos con sus especificaciones técnicas, así como los consumibles en existencia.

<b>1</b>	<b>Juegos de tinacos de agua cruda y cloración.</b>	
<b>2</b>	Hidroneumático: Motobomba de ¾ HP Tanque con membrana de 72 litros	
<b>3</b>	Filtro de multicama 9"x48", poliéster reforzado con fibra de vidrio, 20litros/minuto, multiválvula manual. Material filtrante: zeolita mico z. Volumen del Tanque: 1.58 pie3. Volumen del Material Filtrante: 1 pie3.	
<b>4</b>	Filtro de carbón activado, flujo de 16 litros por minuto (6.6 gpm o 1.5 m3/h), poliéster reforzado con fibra de vidrio 9"x48", válvula manual Carbón activado bituminoso, marca filter clear o similar, para filtro de 24"X58" recto de fibra de vidrio	
<b>5</b>	UV	
	Lampara	25 [W]
	Dosis UV	30,000 [ $\mu$ W-s/cm2]
	Capacidad	25 lpm [1.5 m3/h o 6.6 gpm]
	Manga	Cuarzo
<b>6</b>	Ósmosis inversa de una membrana. Retiene hasta un 99% de minerales disueltos, bacterias, partículas y orgánicos con un peso molecular mayor a 300 una.	
	Rango de PH	6-9
	Turbidez máx.	1 NTU
	TDS	<2000 ppm
	Volumen en 24 hrs:	2200 galones (0.35 m3/h o 1.53 gpm)

	Presión en línea	30 psi
	Bomba	1 multi-etapas, 1HP
7	Ozonificador modelo OZORE8 considerando una dosificación de 1.2 gr/hr, 8 lpm.	
8	Tinaco de acero inoxidable para agua potable.	

Tabla 1. Componentes de la planta potabilizadora UACM-SLT.

## 2.5 Reactor Ultravioleta (UV)

Se cuenta con una planta experimental de potabilización de agua en el plantel de San Lorenzo Tezonco de la Universidad Autónoma de la Ciudad de México (UACM), ésta se encuentra instalada dentro en la zona denominada Servicios generales. Las investigaciones que se desarrollan tiene como uno de sus objetivos la construcción de dispositivos para purificación alternativos a los comerciales. Uno de esos dispositivos es una lámpara UV para desinfección. La figura 3 muestra el diseño del dispositivo que consta de: (1) contenedor o reactor, (2) lámpara, (3) sensor de temperatura, (4) sensor óptico de intensidad UV o turbiedad y (5) agitador.

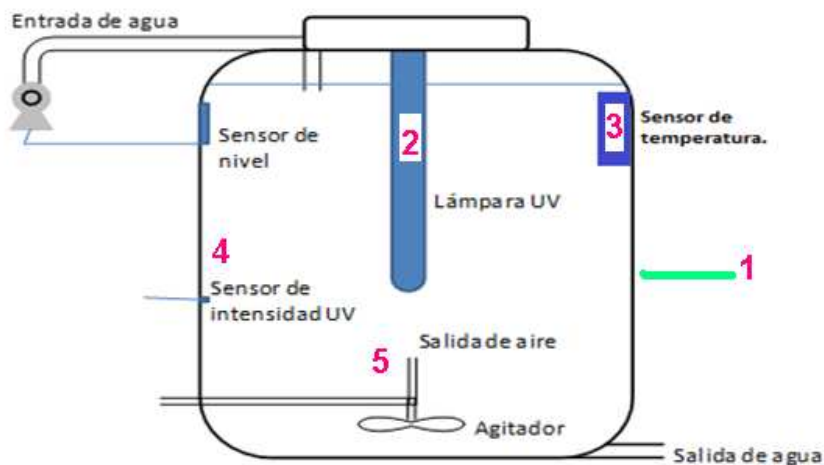


Figura 3. Esquema del fotocatalizador propuesto.

## 2.6 Elementos del sistema

Son numerosos los elementos que componen los distintos sistemas de automatización de residencias y edificios, desde una central de gestión para sistemas integrados hasta un mando automático a distancia. De todos los elementos que lo integran existen 5 básicos: controladores, actuadores, sensores,

conexión a internet y sistemas comunicación entre dispositivos y usuario. Enseguida se hace mención de dos elementos utilizados en el trabajo.

**Sensor:** Se define normalmente como el elemento que se encuentra en contacto directo con la magnitud que se va a medir. El sensor recibe la magnitud física y se la proporciona al transductor.

**Transductor:** De manera general podemos decir que es un elemento o dispositivo que tiene la misión de traducir o adaptar un tipo de energía en otro más adecuado para el sistema, es decir convierte una magnitud física, no interpretable por el sistema, en otra variable interpretable por dicho sistema. El transductor transforma la señal que entrega el sensor en otra normalmente de tipo eléctrico. El transductor suele incluir al sensor [10].

## **2.7 Microprocesadores y microcontroladores**

En tiempos pasados, los sistemas de control se implementaban utilizando exclusivamente la lógica de componentes, lo que hacía que fuesen dispositivos de gran tamaño y muy pesados. Para facilitar una velocidad más alta y mejorar la eficiencia de estos dispositivos de control, se trató de reducir su tamaño apareciendo así los microprocesadores. Siguiendo con el proceso de miniaturización, el siguiente paso consistió en la fabricación de un controlador que integrase todos sus componentes en un solo chip. A esto se le conoce con el nombre de microcontrolador.

En la actualidad existen muchos sistemas que se encuentran constituidos por microcontroladores, una de las características por cual son muy utilizados es por su pequeño tamaño, por su velocidad de transmisión, por sus módulos lo cuales permiten implementar módulos como conversión analógico-digital (ADC), transmisor-receptor universal síncrono-asíncrono (UART) entre otros, por lo cual son una gran alternativa de uso, como en el caso particular en sistemas de monitoreo para diferentes sensores tales como temperatura de agua o en algún otro elemento.

Para este apartado es de vital importancia conocer la arquitectura y características de los microcontroladores, ya que esto nos permitirá distinguir los procesos que se pueden realizar así como sus limitantes.

Un microprocesador es un circuito integrado que contiene la Unidad Central de Proceso (CPU) de un computador, que consta de la Unidad de Control y de los buses necesarios para comunicarse con los distintos módulos. Mientras que un microcontrolador, es un circuito integrado programable que contiene todos los bloques necesarios para controlar el funcionamiento de una tarea determinada:

- Procesador o CPU: que interpreta las instrucciones de programa.
- Memoria RAM para almacenar las variables necesarias.
- Memoria EPROM/PROM/ROM para el programa tipo.
- Puertos E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos.
- Generador de impulsos de reloj que sincroniza el funcionamiento de todo el sistema.

### 2.7.1 Fabricantes de microcontroladores

En la actualidad existen muchos fabricantes de microcontroladores tales como: Texas Instruments, Microchip, ST Microelectronics, Royal Philips Electronics, Infineon Technologies, Toshiba, Mitsubishi, NET Corp., Hitachi Corp., Motorola Semiconductors, Analog Devices, Freescale Semiconductor y Atmel corporation. Cada uno de ellos, que cuentan con ventajas y desventajas que son valorados de acuerdo a sus factores como:

1. Sus presentaciones.
2. La relación calidad/costo.
3. Disponibilidad comercial.
4. Tipo de herramientas disponibles.

En este caso en particular se empleó un microcontrolador de la marca Microchip, una de las razones por las cuales se inclinó por dicha elección, fue porque por el hecho de contar con la tarjeta de experimentación Starter Kit Ethernet de Microchip,

compuesta por un PIC de 32 bits sus características serán expuestas posteriormente a lo largo de los capítulos. Dentro de la marca de microchip, nos encontramos con varios modelos de PIC.

Estos se clasifican de acuerdo a la longitud de sus instrucciones dando lugar a 3 grandes tipos de PIC, en la tabla 2 se muestran. En esta tesis se emplea el microcontrolador de 32 bits (PIC32MX795F512L) del cual se darán a conocer sus especificaciones y características posteriormente con forme se desarrolla el trabajo.

Clasificación	Longitud de instrucciones	Modelos de PIC
<b>Gama baja</b>	8 bits	Pico 10, Pic12, Pic16, Pic18
<b>Gama media</b>	16 bits	Pic24F, Pic24H, dsPIC30, dsPIC33
<b>Gama alta</b>	32 bits	Pic32

Tabla 2. Descripción de los microcontroladores de Microchip.

## 2.8 Comunicación entre dispositivos

Se le denomina al conjunto de elementos que se emplean en el proceso de intercambio de información, mismos que unidos forman un sistema de comunicación. Los componentes vitales para que funcione tal sistema son:

**Emisor.-** Componente que transmite información hacia un punto.

**Receptor.-** Componente que recibe información.

**Canal.-** Es el medio por el medio del cual se realiza el intercambio de información entre el transmisor y receptor.

### 2.8.1 Comunicación alámbrica

También conocida como comunicación por cable, pues tiene lugar a través de líneas de cables (tradicionalmente de cobre), estos realizan la unión entre el emisor y el receptor. La información se transmite mediante impulsos eléctricos u ópticas.

- DSL.
- Fibra óptica.
- Power line Communications-Comunicaciones de la línea eléctrica y x10.
- Cable coaxial.

### 2.8.2 Ethernet

En septiembre de 1980, DIX publicó la primera versión de “Las especificaciones de la Red de Área Local Ethernet”, donde se describía la arquitectura e implementación y funcionamiento de Ethernet [2].

Ethernet es una tecnología de redes de área local LAN (Local Area Network) que transmite información entre computadores.

Cuando se escucha el término Ethernet se hace alusión no a una tecnología sino a las familias de tecnologías de redes de computadoras de área local (LAN) del tipo de acceso múltiple con detección de portadora y detección de colisiones (CSMA/CD) por ello el hecho de hablar en términos de plural.

Dicha tecnología implica tres categorías en velocidades:

1. 10Mbps Ethernet e IEEE 802.3: Especificación LAN que operan a 10Mbps sobre cable coaxial.
2. 100 Mbps Ethernet: Especificación LAN, mejor conocida como FAST ETHERNET, esta opera a 100 Mbps sobre par trenzado.
3. 1000 Mbps Ethernet Especificación LAN conocida como Gigabit Ethernet, opera a 1000Mbps (1Gbps) sobre fibra óptica y par trenzado.

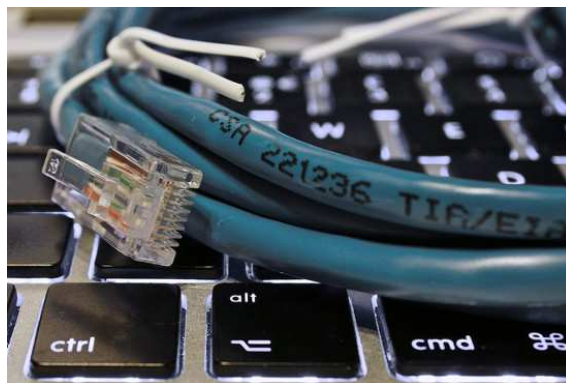


Figura 4. Conexión Ethernet.

## 2.9 Capas ISO/OSI

A finales de los 70s existía la urgencia de estándares para la interconexión de computadoras de diferente naturaleza, por tal motivo, la Organización Internacional para la Estandarización (International Organization for Standardization, ISO) desarrolló una arquitectura para la interconexión de sistemas abiertos, que sirviera como base para la definición de protocolos estándar, a esta arquitectura se le llamó OSI Reference Model (Modelo de Referencia para la Interconexión de Sistemas Abiertos).

Es el protocolo común utilizado por todos los ordenadores conectados a internet, para que estos puedan comunicarse entre sí. Hay ordenadores de clases diferentes; con hardware, software, medios y formas posibles de conexión diferentes.

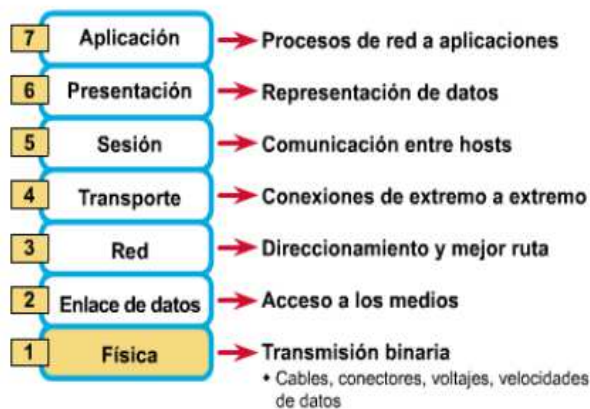


Figura 5. Esquema del modelo OSI.

Este protocolo se encarga de que la comunicación entre todos sea posible. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware.

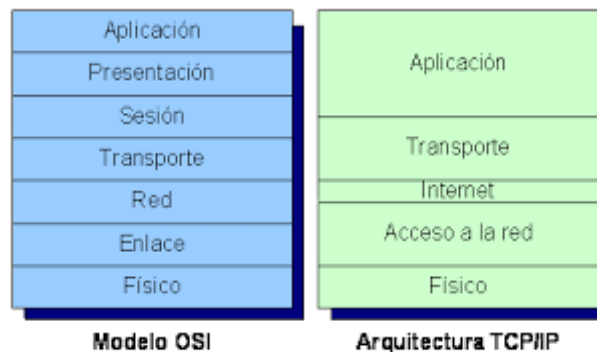


Figura 6. Comparativo de los Modelos TCP/IP y OSI.

La figura 6 muestra la equivalencia entre el modelo TCP/IP y el modelo OSI de comunicación. El modelo TCP/IP cuenta con menos capas a diferencia del modelo OSI. A continuación se dará una breve descripción de las diferentes capas modelo TCP/IP.

### 2.9.1 **Capa física**

Define las características del medio, su naturaleza, el tipo de señales, la velocidad de transmisión, la codificación, etc., en este caso, **Ethernet** es la tecnología que será utilizada como el medio de transporte de datos.

En esta capa se definen las especificaciones las cuales son: eléctricas, mecánicas, funcionales y procedimentales que permiten transmitir y recibir bits entre dos host. En seguida se mencionara a que hace referencia cada una de las especificaciones, con respecto a las eléctricas se toman en cuenta los niveles de voltaje. En las mecánicas se hacen referencia las distancias de transmisión máximas, los conectores físicos como en el caso del estándar "RS-232", el cual define conectores DB-9 y DB-25, así como una distancia de 2.5 [mts]. Las funcionales definen que terminal mecánica o física corresponde a cada terminal eléctrica y por último las procedimentales utilizan para definir como se deben activar, mantener y desactivar la capa física entre los sistemas terminales, de modo tal que permitan transmitir y recibir bits entre dos host [2].

### 2.9.2 **Capa de enlace de datos**

La capa de enlace de datos proporciona tránsito de datos confiable a través de un enlace físico. Al hacerlo, la capa de enlace de datos se ocupa del direccionamiento físico (comparado con el lógico), entrega ordenada de tramas y control de flujo, la topología de red, el acceso a la red, la detección de errores para asegurar la entrega de datos, para lo cual realiza comunicación vía tramas (frames) en las que se indica la dirección MAC origen y destino. Los bridges y switches son equipos de la capa 2 y usan tabla de direcciones MAC.

### 2.9.3 Capa de red

La capa de red es una capa compleja que proporciona conectividad y selección de ruta entre dos sistemas de hosts que pueden estar ubicados en redes geográficamente distintas. Es la responsable del envío y enrutamiento de paquetes de un nodo de red a otro; los switches y routers de la capa 3 usan direcciones MAC. En esta capa se emplean protocolos los cuales son: IPv4, IPv6, ARP, RARP, ICMP. Existen diferentes tipos de tráfico según la capa donde se encuentre y los cuales son:

- Capa 1.- Tráfico de bits.
- Capa 2.- Tráfico de tramas.
- Capa 3.- Tráfico de paquetes.

#### 2.9.3.1 Direcciones IP

Una dirección IP es un número de 32 bits que identifica de manera única a una interfaz de red dentro de Internet. Cada dirección IP tiene que ser única y para ello existe una organización llamada Internet Network Information Center, que se encarga de asignar las direcciones. Una dirección IP normalmente se representa, para hacerla más inteligible, por los valores decimales de cada uno de sus cuatro bytes separados por puntos. Por ejemplo, la dirección IP 11000000 10101000 00000001 00000001 se escribiría como 192.168.1.1. Hay tres tipos de direcciones IP:

- Unicast (para un equipo individual).
- Broadcast (para los equipos de la misma red).
- Multicast (para los equipos del mismo grupo multicast o de multidifusión).

Asimismo, las direcciones IP se clasifican en cinco clases, denominadas A, B, C, D, y E cuyos rangos se indican en la figura 7. La clase A dispone de 7 bits dentro del primer byte para direcciones de red, lo que hace un total de  $2^7=128$  redes, menos dos que no se pueden utilizar (la 0.0.0.0 y la 127.0.0.0 que se reserva para la función loopback). Cada una de estas 126 redes dispone de  $2^{24}=16.777.216$  direcciones que se pueden asignar a equipos (hosts) menos dos que no se

pueden utilizar. La primera es la que tiene todos los bits a 0 e identifica a la red. La segunda es la que tiene todos los bits a 1 y se utiliza como dirección de broadcast de esa red. Es decir, en una red clase A hay 16.777.214 direcciones IP disponibles para equipos [8].

La clase B dispone de 16.384 redes, cada una con 65.534 direcciones que se pueden asignar a equipos. Y la clase C, 2.097.152 redes de 254 equipos cada una. Por ejemplo, la dirección IP 192.168.1.1 pertenece a una red de clase C cuya dirección es la 192.168.1.0 y su dirección de broadcast es 192.168.1.255. Esta red dispone de 254 direcciones para hosts (de la 192.168.1.1 a la 192.168.1.254), de las cuáles la primera (192.168.1.1) suele reservarse para la puerta de enlace, normalmente un router.

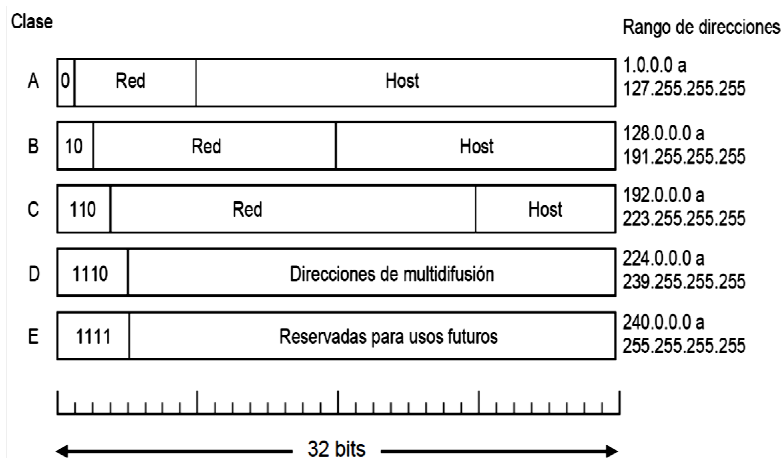


Figura 7. Clases de direcciones IP.

La primera parte de una dirección IP indica la red que alberga un conjunto de equipos y la segunda identifica a un equipo determinado dentro de esa red. Para indicar qué parte de la dirección corresponde a la red, lo que suele denominarse como prefijo de red, y qué parte corresponde al equipo se utiliza la máscara de red. Esta máscara es un número de 32 bits cuyos primeros bits, tantos como tenga el prefijo de red, valen 1 siendo el resto igual a 0. La tabla 3 muestra las máscaras de las tres clases anteriores [11]:

Clase	Mascara
<b>A</b>	255.0.0.0
<b>B</b>	255.255.0.0
<b>C</b>	255.255.255.0

Tabla 3. Máscaras de las clases de redes.

#### 2.9.4 Capa de transporte

En esta capa se encuentran involucrados los aspectos de calidad del servicio con respecto a la confiabilidad, el control de flujo y la corrección de errores. En ella se implementa el Protocolo de Control de Transmisión (TCP), ofrece maneras flexibles y de alta calidad para crear comunicaciones de red confiables, sin problemas de flujo y con un nivel de error bajo.

TCP es un protocolo orientado a la conexión. Mantiene un diálogo entre el origen y el destino mientras empaqueta la información de la capa de aplicación en unidades denominadas segmentos.

Significa que los segmentos de Capa 4 viajan de un lado a otro entre dos hosts para comprobar que la conexión exista lógicamente para un determinado período. Esto se conoce como conmutación de paquetes.

Funciones de la capa de transporte

- Direccionamiento y multiplexación.
- Control de flujo de extremo a extremo (entrega ordenada y retransmisión).
- Establecimiento y liberación de conexiones (para servicios orientados a conexión).

La capa de transporte involucra únicamente protocolos:

- El protocolo de control de transmisión (TCP).
- El protocolo de datagrama de usuario (UDP).

El protocolo de capa de transporte crea un flujo virtual de datos entre la aplicación de envío y la de recepción, que se identifica con un número de puerto de

transporte. El número de puerto identifica un puerto, una ubicación dedicada de la memoria para recibir o enviar datos.

#### 2.9.4.1 Protocolos de comunicación

Los protocolos de comunicación son conjuntos de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información. Para que dos equipos, ya sea dentro de la misma red o en otra, puedan comunicarse es necesario establecer y decidir cuáles son los protocolos indicados y así poder llevarlos a cabo, deberán hablar el mismo idioma, uno de los más importantes el protocolo TCP/IP, para que una computadora tenga comunicación deberá tener este instalado [3]. En la figura 8 se muestran los protocolos empleados en cada una de las capas del modelo TCP/IP.

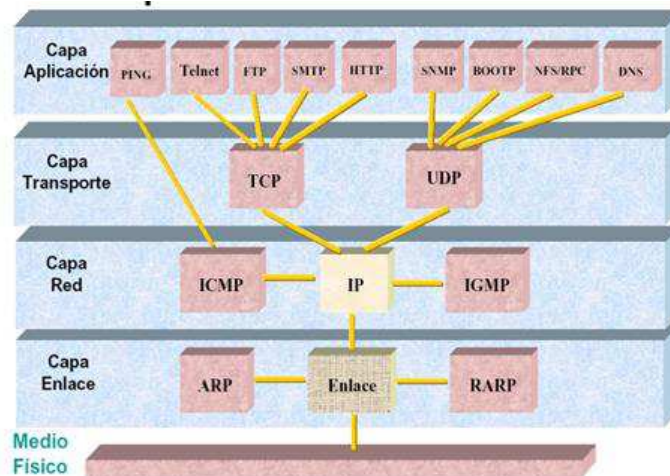


Figura 8. Protocolos por cada capa de TCP.

#### 2.9.5 Capa de sesión

La capa de sesión establece, administra y finaliza las sesiones entre dos hosts que se están comunicando. También sincroniza el diálogo entre las capas de presentación de los dos hosts y administra su intercambio de datos. Además de regular la sesión, la capa de sesión ofrece disposiciones para una eficiente transferencia de datos, clase de servicio y un registro de excepciones acerca de los problemas de la capa de sesión, presentación y aplicación. En la figura 9 muestra el envío de una señal de sincronía para establecer la sesión y su acuse de recibo (ACK) [2].

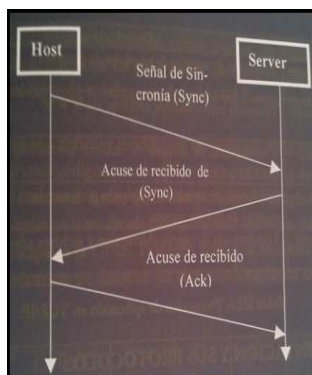


Figura 9. Señal de sincronía para establecer conexión.

### 2.9.6 Capa de presentación

La capa de presentación garantiza que la información que envía la capa de aplicación de un sistema pueda ser leída por la capa de aplicación de otro. De ser necesario, la capa de presentación traduce entre varios formatos de datos utilizando un formato común. Se encarga de definir el formato de los datos entre aplicaciones de red, así como es la codificación, cifrado compresión de los datos transmitidos.

### 2.9.7 Capa de aplicación

La capa de aplicación es la capa del modelo OSI más cercana al usuario; suministra servicios de red a las aplicaciones del usuario. Difiere de las demás capas debido a que no proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del modelo OSI. Algunos ejemplos de aplicaciones son los programas de hojas de cálculo, de procesamiento de texto y los de las terminales bancarias. La capa de aplicación establece la disponibilidad de los potenciales socios de comunicación, sincroniza y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos.

Soporta servicios como: correo electrónico, transferencia de archivos, bases de datos compartidas u otros tipos de servicios distribuidos.

### 2.9.8 Diferencias entre el modelo OSI y TCP

En la tabla 4 se hará mención de las diferencias entre los modelos de comunicación TCP/IP y OSI.

Similitudes	Diferencias	Comparación	Uso
Ambos se dividen en capas.	TCP/IP combina las funciones de la capa de presentación y de sesión en la capa de aplicación.	Los protocolos TCP/IP son los estándares en torno a los cuales se desarrolló la Internet, de modo que la credibilidad del modelo TCP/IP se debe en gran parte a sus Protocolos. En comparación, las redes típicas no se desarrollan normalmente a partir del protocolo OSI, aunque el modelo OSI se usa como guía.	Aunque los protocolos TCP/IP representan los estándares en base a los cuales se ha desarrollado la Internet, se utiliza el modelo OSI por los siguientes motivos:  Es un estándar mundial, genérico, independiente de los protocolos.  Es más detallado, lo que hace que sea más útil para la enseñanza y el aprendizaje. Al ser más detallado, resulta de mayor utilidad para el diagnóstico de fallas.
Ambos tienen capas de aplicación, aunque incluyen servicios muy distintos.	TCP/IP combina las capas de enlace de datos y la capa física del modelo OSI en una sola capa.		
Ambos tienen capas de transporte y de red similares.			
La tecnología es de conmutación por paquetes (no de conmutación por circuito).			

Tabla 4. Desventaja y ventajas del modelo TCP/IP.

## 2.10 Protocolos utilizados en la pila TCP/IP de Microchip

Microchip ofrece un conjunto de librerías que implementan la pila de protocolos TCP/IP, optimizados para los PIC18, PIC24 y PIC32 familia de microcontroladores dsPIC. Las librerías son un conjunto de funciones de software que proveen servicio a toda la pila de protocolos TCP/IP a las aplicaciones de los microcontroladores. Basada en el modelo TCP/IP, la pila se divide en múltiples capas, la misión de cada capa es proveer servicios a las capas superiores. De esta manera, cada capa debe ocuparse exclusivamente de su nivel inmediatamente inferior, a quien solicita servicios, y del nivel inmediatamente superior, a quien devuelve resultados. Por especificaciones, muchas de las capas están “en vivo”, en el sentido de que no sólo actúan cuando se solicita un servicio, sino también cuando los eventos como el tiempo de espera o la llegada de un nuevo paquete se producen. La pila TCP/IP de Microchip incluye las siguientes características clave [12]. Los protocolos soportados por capa se mencionan a continuación.

Nivel de enlace de datos: Protocolo de resolución de direcciones (ARP).

Nivel de internet: Protocolo de Internet (IP).

Nivel Red: Protocolo de Mensajes de Control de Internet (ICMP).

Nivel de transporte: Protocolo de Usuario Datagram (UDP) o Protocolo de Control de Transmisión (TCP).

Nivel de aplicación: Protocolo de Configuración Dinámica de Host (DHCP), Protocolo Simple de Administración de Red (SNMP), Protocolo de Transferencia de Hipertexto (HTTP), Protocolo de Transferencia de Archivos (FTP), Protocolo de Transferencia de Archivos Trivial (TFTP). La distribución protocolos por cada uno de los niveles de las capas se observan figura 10.

Incluye además los servicios:

- Soporte de sockets de TCP y UDP.
- Secure Sockets Layer (SSL).
- Servicio de nombres NetBIOS.
- DNS - Domain Name System (Sistema de nombres de dominio).
- Ethernet Device Discovery (Detección de dispositivos Ethernet).

La pila tiene un diseño modular y está escrito en el lenguaje de programación C. Implementaciones eficaces se pueden lograr en más o menos 28-34 KB de código, en función de los módulos utilizados [4].

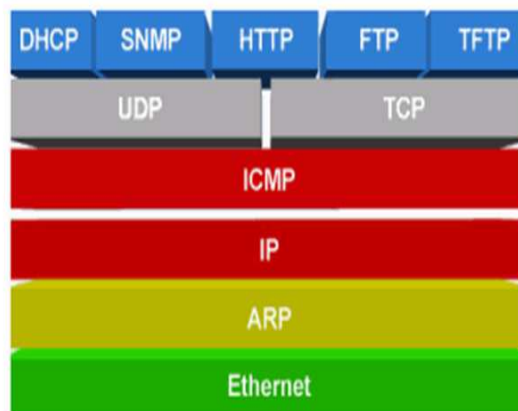


Figura 10. Jerarquización de la pila TCP/IP.

## 2.11 Redes informáticas

Una red de computadoras (también llamada red de ordenadores o red informática) es un conjunto de equipos (computadoras y dispositivos), conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, para compartir información (archivos), recursos (discos, impresoras, programas, etc.) y servicios (acceso a una base de datos, internet, correo electrónico, chat, juegos, etc.). A cada una de las computadoras conectadas a la red se le denomina un nodo [5].

- Según su cobertura: PAN, WAN, LAN y MAN.
- Según su topología: Estrella, bus, jerárquicas e híbridas.
- Según su relación funcional: Cliente-Servidor, igual a igual.

**PAN (red de área Personal):** Es una red de ordenadores usada para la comunicación entre los dispositivos de la computadora (teléfonos incluyendo los ayudantes digitales personales) cerca de una persona. El alcance de una PAN es de algunos metros. Se pueden conectar con cables a los USB y FireWire de la computadora. Una red personal sin hilos del área (WPAN) se puede también hacer posible con tecnologías de red tales como IrDA y Bluetooth [11].

**LAN (red de área local):** Una red que se limita a un área tal como una habitación, un solo edificio o una embarcación. Una LAN grande se divide generalmente en segmentos lógicos más pequeños llamados grupos de trabajo.

**MAN (red de área metropolitana):** Una red que conecta las redes de dos o más locales pero no se extiende más allá de los límites de la una ciudad.

**WAN (red de área mundial):** Es una red que cubre un área geográfica amplia y en gran parte de su estructura utiliza instalaciones de transmisión telefónicas.

Las distribuciones de Linux como Ubuntu son muy adecuadas para las funciones de los sistemas operativos de servidor. De hecho, Ubuntu tiene su propia edición de servidor diseñado específicamente para funcionar como un sistema operativo de servidor. Debido a su facilidad de instalación y uso, vamos a crear una red de

área local (LAN) con una mezcla de equipos con Windows para los usuarios finales y un servidor de Ubuntu.

## **2.12 Protocolo de Configuración Dinámica de Host (DHCP)**

DHCP utiliza un modelo cliente-servidor. El administrador de la red establece uno o varios servidores DHCP que mantienen la información de configuración de TCP/IP y la proporcionan a los clientes. La base de datos del servidor incluye lo siguiente:

- Los parámetros de configuración válidos para todos los clientes de la red.
- Un conjunto de direcciones IP válidas para su asignación a los clientes, junto con direcciones reservadas para su asignación manual.
- La duración de una concesión ofrecida por el servidor. La concesión define el período de tiempo de uso de la dirección IP asignada.

Al haber un servidor DHCP instalado y configurado en la red, los clientes habilitados para DHCP pueden obtener sus direcciones IP y los parámetros de configuración relacionados dinámicamente cada vez que inician una sesión y se unen a la red.

Básicamente el servicio DHCP funciona de la siguiente forma. Existe un programa servidor en un host de la red que escucha las solicitudes de los clientes y que en su configuración almacena tablas de posibles direcciones IP a otorgar además del resto de la información. Cuando un cliente requiere del servicio, envía una solicitud en forma de broadcast a través de la red. Todos los servidores alcanzados por la solicitud responden al cliente con sus respectivas propuestas, éste acepta una de ellas haciéndoselo saber al servidor elegido, el cual le otorga la información requerida. Esta información se mantiene asociada al cliente mientras éste no desactive su interfaz de red o no expire un plazo determinado [19].

## **2.13 Modelo Cliente-Servidor**

En el mundo de TCP/IP las comunicaciones entre computadoras se rigen básicamente por lo que se llama modelo Cliente-Servidor, éste es un modelo que intenta proveer usabilidad, flexibilidad, interoperabilidad y escalabilidad en las

comunicaciones. El término Cliente/Servidor fue usado por primera vez en 1980 para referirse a PC's en red [8]. En la figura 11 se muestra el modelo cliente/servidor y en el cual se pueden observar los elementos principales y elementales.

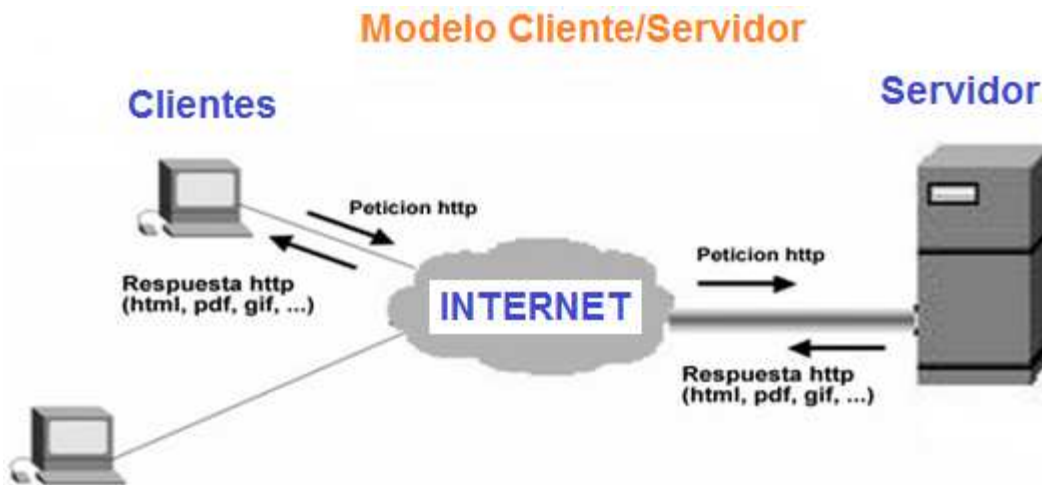


Figura 11. Bosquejo del modelo cliente/servidor.

### 2.13.1 Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término **front-end**. Normalmente maneja todas las funciones relacionadas con la Manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

### 2.13.2 Servidor

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término **back-end**. Regularmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.

### 2.13.3 Php

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Características de php:

- Soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, Mysql, Oracle, Informix, PostgreSQL, entre otras.
- Su sintaxis es muy similar a la del ASP.
- Ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML.
- El código PHP va incrustado dentro del código HTML.

Se deberán instalar los siguientes paquetes para php: php5 libapache2-mod-php5 php5-cli php5-mysql, todos ellos se instalaran con el fin de contar con todas las herramientas que dicho lenguaje ofrece, todo se hizo mediante línea de comandos para poder realizar hacer la instalación se deberán colocar lo que aparece a continuación para cada uno de los paquetes [6].

```
root@yadhiraserver$ apt-get install php5
```

Todos aquellos script que se realizan en php serán alojados en directorio del servidor `www/var`.

**root@yadhiraserver: /var/www/**

En la figura 12 se observan las características del php instalado en el servidor.

The screenshot shows a web browser window with the address bar displaying `192.168.1.202/test.php`. The page content includes the PHP logo and a table of configuration details.

PHP Version 5.3.2-1ubuntu4.24	
<b>System</b>	Linux yadhiraserver 2.6.32-57-generic-pae #119-Ubuntu SMP Wed Feb 19 01:20:04 UTC 2014 i686
<b>Build Date</b>	Apr 4 2014 02:29:24
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	disabled
<b>Configuration File (php.ini) Path</b>	/etc/php5/apache2
<b>Loaded Configuration File</b>	/etc/php5/apache2/php.ini
<b>Scan this dir for additional ini files</b>	/etc/php5/apache2/conf.d
<b>Additional ini files parsed</b>	/etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/iconv.ini, /etc/php5/apache2/conf.d/imap.ini, /etc/php5/apache2/conf.d/interbase.ini, /etc/php5/apache2/conf.d/ldap.ini, /etc/php5/apache2/conf.d/mcrypt.ini, /etc/php5/apache2/conf.d/memcache.ini, /etc/php5/apache2/conf.d/ming.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/odbc.ini, /etc/php5/apache2/conf.d/openssl.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini, /etc/php5/apache2/conf.d/pdo_sqlite.ini, /etc/php5/apache2/conf.d/pgsql.ini, /etc/php5/apache2/conf.d/soap.ini, /etc/php5/apache2/conf.d/sockets.ini, /etc/php5/apache2/conf.d/ssh2.ini, /etc/php5/apache2/conf.d/streams.ini, /etc/php5/apache2/conf.d/tidy.ini, /etc/php5/apache2/conf.d/xmlrpc.ini, /etc/php5/apache2/conf.d/xsl.ini
<b>PHP API</b>	20090626
<b>PHP Extension</b>	20090626

Figura 12. Características del PHP instalado.

#### 2.13.4 Mysql

Mysql es un sistema gestor de base de datos, es utilizado para el desarrollo de páginas web. En él se emplean algunos comandos cada uno de ellos con una tarea específica por lo que a continuación se enlistaran las instrucciones más utilizadas.

**mysql> show databases.-** Permite observar la base de datos existentes.

**mysql> create database prueba.-** Para poder crear una base de datos se coloca lo anterior.

**mysql> use prueba.-** Con esta instrucción podemos hacer uso y referencia a nuestra base, es decir si ya se tiene una base de datos que se llama "prueba", antes de poder trabajar con ella se debe de seleccionar y para ello se teclea dicho comando dentro de mysql.

**mysql> drop database prueba.-** Borrar una base de datos, en caso de que equivocarse y ya no querer o cambiar de nombre se utiliza lo anterior con la finalidad de poder borrarla y crear una nueva.

**mysql> create table datos (nombre varchar (20), edad int (10)).**- Para crear una tabla se deberá colocar lo anterior, dentro se colocaran el nombre del parámetro, así como de qué tipo es y cuantos espacios se les asignaran.

**mysql> drop table datos.**- Para borrar la tabla "datos".

**mysql> describe datos.**- Muestra una breve descripción de los campos de la tabla

**mysql> select \* from datos.**- Muestra el contenido de la tabla.

### 2.13.5 Crear un usuario de Mysql

Se creó el usuario de yadhira con la intención de colocar la base de datos dentro de la misma así como darle ciertos privilegios, a continuación se observa cómo se lleva a cabo este proceso.

**mysql> create user 'yadhira'@'localhost'**

Para otorgarle TODOS los privilegios a un usuario de mysql, sobre la tabla "pruebabase" de la base de datos "datos" al usuario "yadhira" colocamos lo que aparece a continuación.

**mysql> use pruebabase;**

**mysql> grant all privileges on pruebabase to yadhira;**

## 2.14 Phpmyadmin

Phpmyadmin es una herramienta de administración de mysql vía web, lo cual ayuda y facilita la manipulación de la base de datos, debido a su interfaz gráfica ya que contiene iconos y elementos fáciles y manejables por el administrador, para acceder a ella es necesario colocar lo siguiente en cualquier navegador de preferencia [7].

➤ **192.168.1.202/phpmyadmin**

En la figura 13 se observa la respuesta de colocar la IP mencionada anteriormente, en la cual se puede ver una ventanilla de inicio para acceder a la base de datos por lo que se deberá colocar el usuario y contraseña correspondiente y así

identificarse y entrar al sistema y realizar las modificaciones y cambios pertinentes por el administrador.

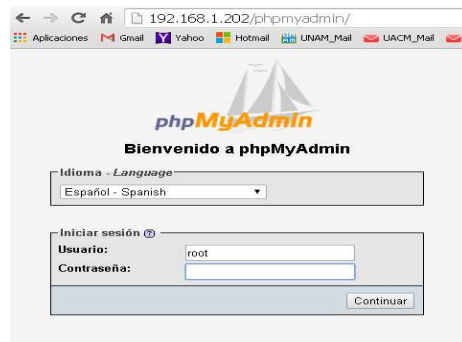


Figura 13. Comprobación de la instalación de phpmyadmin,

## 2.15 Socket

Socket es un módulo de software en donde dos programas pueden intercambiar información entre dos computadoras para lograr la comunicación de forma confiable se define un puerto.

*La implementación de socket radica en la capa de transporte. Cada segmento TCP contiene los números de puerto fuente y destino con el fin de identificar la aplicación emisora y la receptora. Estos dos valores, combinados con las direcciones IP fuente y destino, identifican cada conexión de forma única. La combinación de un número de puerto y una dirección IP se conoce como socket. Por tanto, el cuarteto compuesto por la dirección IP del Cliente, el número de puerto del Cliente, la dirección IP del Servidor y el número de puerto del Servidor; forma la pareja de sockets que identifican cada conexión TCP [3].*

Para realizar la transferencia de paquetes por medio de Ethernet como método de transmisión de datos, se implementó un socket, el cual nos permite compartir información a través de un puerto. El cliente y el servidor deben ponerse de acuerdo sobre el protocolo que utilizarán.

Se deben definir las propiedades de las comunicaciones en las que se ve envuelto un socket, esto es, el tipo de comunicación que se puede dar entre cliente y servidor. Estas pueden ser:

- Fiabilidad de transmisión.
- Mantenimiento del orden de los datos.
- No duplicación de los datos.
- El "Modo Conectado" en la comunicación.
- Envío de mensajes urgentes.

Se define mediante los parámetros:

- Protocolo de red y de transporte.
- Dirección de red.
- Número de puerto.

Para la programación de socket existen funciones y variables ya definidas en la tabla 5 se hace mención de ellas. Para su implementación es necesario colocarlas de manera adecuada y ordenada esto permitirá una buena conexión y funcionamiento del socket requerido. Existen dos tipos de socket en este proyecto se utilizara el tipo stream, por lo que es importante conocer lo que realiza a continuación se hace una breve referencia.

Tipo **SOCK\_STREAM**: para comunicaciones fiables en modo conectado, de dos vías y con tamaño variable de los mensajes de datos.

Primitiva	Uso
<b>SOCKET</b>	Crear un extremo [end point] de la comunicación.
<b>BIND</b>	Asociar una dirección local a un socket (IP: puerto).
<b>LISTEN</b>	Anunciar la disponibilidad para aceptar conexiones.
<b>ACCEPT</b>	Bloquear el servidor hasta que llegue una solicitud de conexión
<b>CONNECT</b>	Intentar establecer una conexión.
<b>SEND</b>	Enviar datos a través de la conexión.
<b>RECEIVE</b>	Recibir datos provenientes de la conexión.
<b>CLOSE</b>	Cerrar conexión

Tabla 5. Instrucciones para implementar en el socket TCP.

### 2.15.1 Selección del puerto

Cada computadora o equipo informático que se encuentre conectado a una red, utiliza unas especies de puerto de comunicación por las que recibe y envía información a otros equipos diferentes.

Para poder realizar varios intercambios de datos de forma simultánea, existen 65536 puntos de salida y entrada, estos son conocidos como puertos y se identifican de forma numérica.

La definición técnica es: Un número de 16 bits, empleado por un protocolo host a host para identificar a que protocolo del nivel superior o programa de aplicación se deben entregar los mensajes recibidos.

Se encuentran organizados mediante las reglamentaciones asignadas por la IANA (Agencia de Asignación de Números de Internet) en tres categorías:

Puertos	Descripción
Puertos entre el 0 y el 1023 son puertos reservados	Usos específicos que se encuentran reglamentados, el sistema operativo los abre para permitir su empleo por diversas aplicaciones mediante los llamados protocolos "Bien conocidos", por ejemplo: HTTP, FTP, TELNET, IRC, POP3, etc.
Puertos 1024 y 49151 son denominados "Registrados"	Son usados para cualquier aplicación.
Entre los números 49152 y 65535 son denominados "Dinámicos o privados"	Son los usados por el sistema operativo cuando una aplicación tiene que conectarse a un servidor y le realiza la solicitud de un puerto.

Tabla 6. Clasificación de puertos.

## 2.16 Protocolo http

Para la transferencia y consulta de la base de datos se utiliza el protocolo HTTP 1.1, el cual permite realizar dos acciones tales como: request y response estos nos permiten realizar la transferencia de datos del servidor al PIC. Es decir que con la implementación de dicho protocolo se realizara él envió de las temperaturas a la base de datos mismas que serán almacenadas en el servidor y para observar su comportamiento de manera gráfica se hará la consulta por medio de la página WEB. En la figura 14 se muestra el proceso de dicho protocolo.

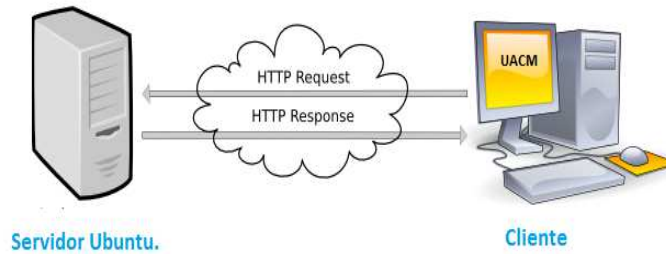


Figura 14. Método http 1.1.

Métodos existentes para el protocolo HTTP:

POST: Envía un elemento al servidor.

PUT: Solicita colocar información en la URL identificada.

DELETE: Solicita al servidor que elimine la entidad indicada por el URL.

TRACE: Permite obtener un réplica del mensaje enviado, de esta forma el cliente sabe con certeza que recibió el servidor.

OPTIONS: Permite al cliente consultar que métodos hay asociados a un recurso.

GET: Descarga un elemento.

HEAD: Solicita ver las cabeceras que se enviarán con un elemento concreto.

### 2.17 Características del router Linksys Cisco

Se creó una red local LAN todo ello con la finalidad de colocar, todos los dispositivos que en este caso fueron: Microcontrolador PIC 32, Servidor, la PC que controla de manera web algunas aplicaciones alojadas en el servidor, y con respecto al microcontrolador por medio de ella se utiliza el software de programación del PIC. Para ello se utilizó un router Cisco, el cual cuenta con especificaciones que se muestran en la tabla 7.

Especificaciones	Descripción
<b>Estándares</b>	IEEE 802.3, IEEE 802.3u, IEEE 802.11b, borrador 802.11g
<b>Puertos</b>	1.- Un puerto 10/100 RJ-45 para cable / módem DSL
<b>1.-Internet:</b>	
<b>2.-LAN:</b>	2.- Cuatro 10/100 RJ-45 puertos conmutados Un puerto de alimentación
<b>Botón</b>	Un botón de reinicio
<b>Tipo de cableado</b>	Categoría 5 Ethernet cableado de red o mejor
<b>Potencia de transmisión</b>	15 dBm
<b>Modulación</b>	
<b>1.- 802.11b:</b>	1.- CCK, DQPSK, DBPSK
<b>2.- Borrador 802.11g:</b>	2.- OFDM
<b>Protocolos de red</b>	TCP/IP , NetBEUI
<b>1.-Leds indicadores</b>	1.- Power, DMZ, Diag
<b>2.-WLAN</b>	2.- Act, Link
<b>3.-LAN</b>	3.- Link/Act, Full/Col, 100
<b>4.-Internet</b>	4.- Link/Act, Full/Col, 100

Tabla 7. Características del Router.

A continuación se enuncian dos de los estándares que utiliza el router, los cuales son:

**802.11b.**-Uno de los estándares IEEE (*Instituto de Ingeniería Eléctrica y Electrónica*) para hardware de red inalámbrica. Los productos que se adhieren a un estándar IEEE específica trabajarán con los demás, incluso si son fabricados por diferentes empresas. El estándar 802.11b especifica una velocidad de transferencia de datos máxima de 11 Mbps, una frecuencia de operación de 2,4 GHz y codificación WEP para la seguridad.

**802.11g.**-Una propuesta, pero la extensión no ratificado todavía de la norma IEEE 802.11 para el hardware de redes inalámbricas. Los proyectos de especificaciones 802.11g utilizados por Linksys especificar una velocidad máxima de 54 Mbps de transferencia de datos utilizando *OFDM (Multiplexación por División de*

*Frecuencias Ortogonales*), una frecuencia de operación de 2,4 GHz, compatibilidad con dispositivos IEEE 802.11b, y el cifrado WEP para la seguridad.

# CAPÍTULO 3

*Análisis y diseño.*

### 3 Análisis y diseño

Revisando y analizando las diferentes aplicaciones y características del PIC 32 en especial la del módulo Ethernet y tomando en cuenta que hoy en día el internet es muy utilizado por el ser humano, se ha pensado en diseñar e implementar un sistema que integre estas tecnologías en una aplicación útil para la planta purificadora de agua de la UACM SLT, que permita monitorear del sensor de temperatura en forma remota. Para conseguir desarrollar el proyecto se dividió el sistema total en sistemas parciales, de esta manera se logró reducir el trabajo, en la figura 15 se observa el diseño que tendrá el sistema.

Actividades a realizar:

1. Programación del microcontrolador PIC32: ADC, timer y LCD, socket cliente.
2. Implementación de socket servidor.
3. Creación de la red LAN.

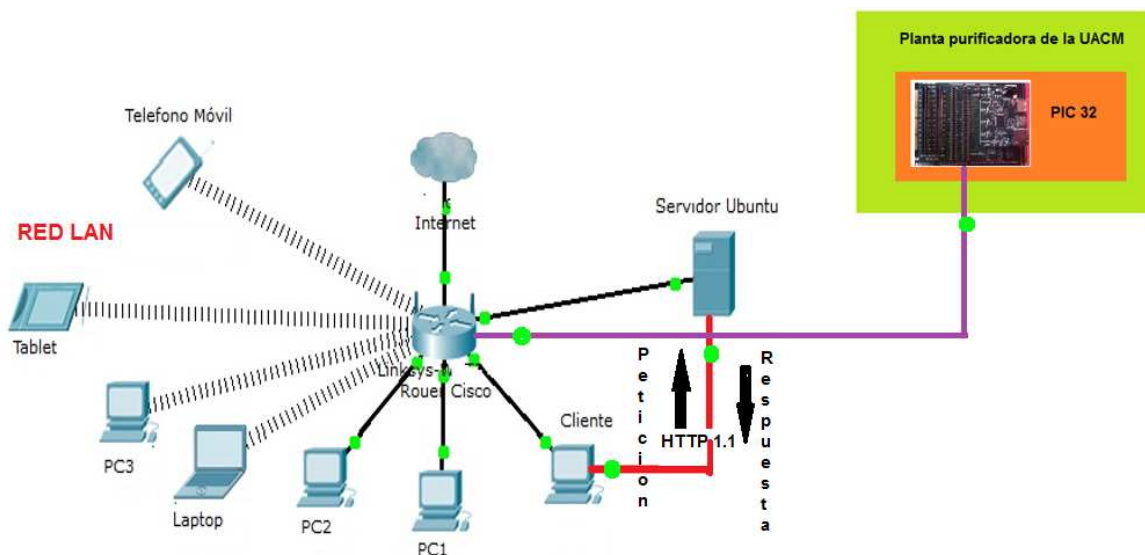


Figura 15. Sistema de diseño del problema.

#### 3.1 Ethernet start kit PIC32MX795F512L

PIC 32 este es un microcontrolador que cuenta con diferentes módulos mismos que usaremos, tales como; el módulo ADC, el cual nos permitirá realizar la

conversión analógica, el modulo Ethernet el cual nos proveerá la conexión a internet al igual que el modulo del timer.

El fabricante de Microchip Technology Inc, cuenta con una serie de microcontroladores (MCU's) cuya característica es la integración del núcleo MIPS32 "Mk4" de 80 [MHz], con 512 [KB] en memoria Flash y 32 [KB] en memoria RAM [20]. La gama de la familia PIC32 se conforma de microcontroladores de 64 y 100 pines una de sus características es que involucran una compatibilidad con de funcionamiento con MCUs PIC 8,16 y 32 bits debido a la flexibilidad y re-configuración de sus pines, herramientas de desarrollo y sus varios periféricos

### 3.1.1 Características del PIC32MX795F512L

Se trabajara con el PIC32MX795F512L, por lo cual se deberán conocer sus características y así poder programar los módulos tales como: ADC, Timer. Dicho microcontrolador cuenta con 7 puertos (A, B, C, D, E, F y G) con 16 bits, en la tabla 9 se muestran dicha distribución y como son pueden ser utilizados si como entrada o salida o ambos. El microcontrolador PIC32MX795F512L, contiene diversos periféricos, en la tabla 8 se enlistan sus principales características

Dispositivo	PIC32MX795F512L
Pines	100
Frecuencia	80[MHz]
Memoria de programa (kb)	512+12 <sup>(1)</sup>
Memoria de datos (kb)	128
USB	1
Ethernet	1
CAN	2
Temporizadores / Captura / Comparación	5/5/5
Canales DMA Programable / edicado	8/8
UART (2,3)	6
SPI (3)	4
I <sup>2</sup> C <sup>M(3)</sup>	5
10-bit 1 Mbps ADC (Canales)	16
Comparadores	2

Tabla 8. Características del microcontrolador.

Puertos	Tamaño	Pines	Tipo	E=Entrada S=Salida
A	16 Bit	16	RA0-RA15	E/S
B	16 Bit	16	RB0-RB15	E/S
C	16 Bit	16	RC0-RC15	E/S
D	16 Bit	16	RD0-RD15	E/S
E	16 Bit	16	RE0-RE15	E/S
F	16 Bit	16	RF0-RF15	E/S
G	16 Bit	4	RG0, RG1, RG4, RG15	E/S
G	16 Bit	2	RG2 Y RG3	E

Tabla 9. Tabla de puertos entradas y salidas.

En la figura 16, se puede observar el diagrama de terminales del microcontrolador y se observan los 100 pines con los que cuenta, algunos de los cuales serán ocupados para el ADC, LCD y sensor de temperatura.



Figura 16. Diagrama de pines del PIC usado.

### 3.1.2 Ethernet starter kit

La tarjeta de desarrollo es una aplicación para desarrollo rápido cuenta con un microcontrolador e interfaces de comunicación tales como: Ethernet y USB. Enseguida se muestran algunas de sus características:

1. PIC32MX795F512L microcontrolador de 32 bits.
2. PIC32MX440F512H microcontrolador USB para la depuración de a bordo.
3. Indicador de energía LED verde.

4. Cristal de a bordo para la sincronización del microcontrolador precisión (8 MHz).
5. Conectividad USB para las comunicaciones del depurador de a bordo.
6. Indicador depurador LED naranja.
7. Tres pulsadores interruptores para entradas definidas por el usuario.
8. Tres LED indicadores definidos por el usuario.
9. USB tipo A conectividad receptáculo para las aplicaciones basadas en host PIC32.
10. Jumper de energía en modo HOST.
11. Puerto RJ-45 Ethernet.
12. Ethernet indicador de velocidad 10/100 de bus LED.
13. 50 MHz del oscilador Ethernet PHY.
14. 32 kHz oscilador (opcional). Para real-time clock (RTC).
15. Host USB y fuente de alimentación de OTG para la alimentación de las aplicaciones PIC32 USB.

En la figura 17 se puede observar los elementos con los que cuenta la tarjeta, mismos que fueron enlistados anteriormente.

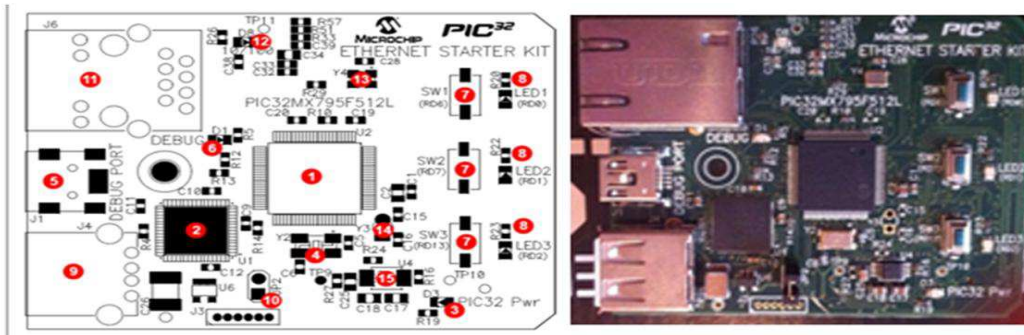


Figura 17. Componentes de la tarjeta.

### 3.1.3 Dauther card o tarjeta de expansión

El Starter kit se le pueden agregar varias tarjetas, una de ellas es la de expansión de puertos a continuación se describen sus características. Cuenta con dos terminales J10 y J11 cada una de ellas con 60 entradas como se muestra en la figura 18. Solo se utilizaran alguno cuantos para cada etapa, conforme se

desarrolle y se realizan cada una de las interfaces. Se hará referencia a cuales fueron los que se utilizaron en cada una de las etapas.

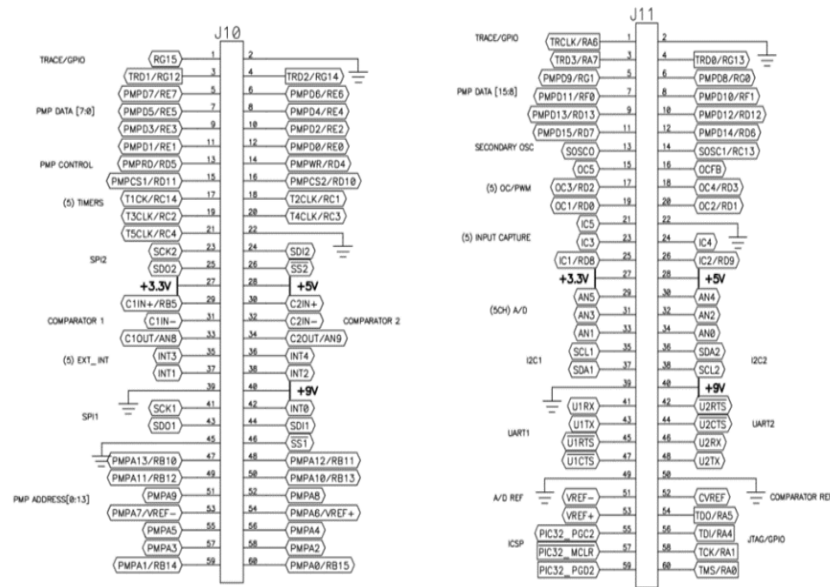


Figura 18. Descripción tarjeta de expansión.

En la figura 19 se muestra los módulos starter kit conectado a la tarjeta de expansión.

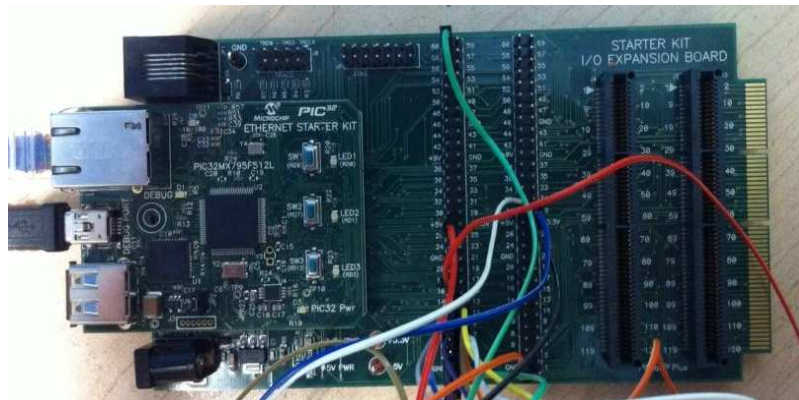


Figura 19. PIC 32 con la tarjeta de expansión.

### 3.1.4 Conectar un microcontrolador a una red Ethernet

El objetivo de conectar un Microcontrolador a una red Ethernet para recibir o enviarle instrucciones al PIC desde una página WEB a través del protocolo HTTP puede hacerse de dos maneras: la primera es utilizar un controlador externo que

implementa ya todo lo necesario para la comunicación Ethernet, el Microcontrolador se conecta al módulo Ethernet a través de una interfaz como RS-232, SPI, etc. En este grupo se encuentran dispositivos como: ENC28J60, ENC624J600, Rabbit, EZ Web Lynx, SitePlayer, etc. La otra opción es utilizar un Microcontrolador que lleve ya integrado el controlador Ethernet, Microchip dispone de varios PICs con estas características como el PIC18F67J60, PIC18f97J60, PIC32MX795FXXX, etc.

En el caso de utilizar un módulo externo, la ventaja es que se puede utilizar cualquier Microcontrolador para realizar una aplicación que envíe y reciba datos desde Internet, al estar todo lo referente a la comunicación Ethernet en el módulo externo, no se requiere implementar nada más que la aplicación en el Microcontrolador y las funciones necesarias para el envío de datos serie al controlador, con el consiguiente ahorro de recursos en el PIC.

Si se utiliza un Microcontrolador que incorpore el controlador Ethernet no se requiere utilizar un componente extra para la gestión de la comunicación Ethernet. Sin embargo se debe incluir en el código del Microprocesador el conjunto de librerías o Stack TCP/IP que suministre el desarrollador del compilado.

### 3.1.5 Módulo ADC

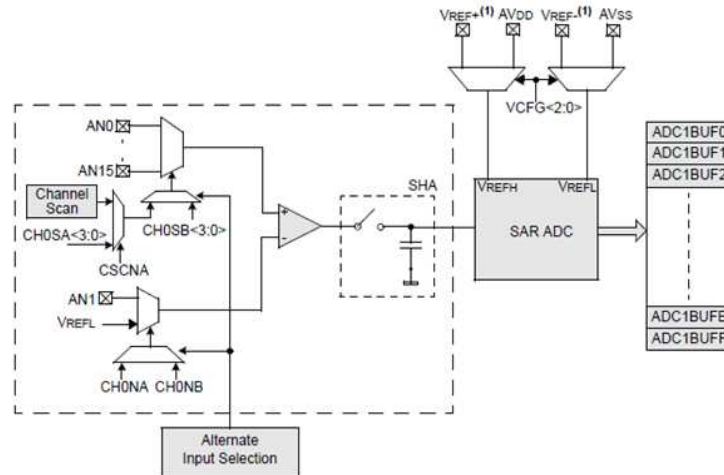
Como es muy frecuente el trabajo con señales analógicas, éstas deben ser convertidas a digital y por ello muchos microcontroladores incorporan un conversor A/D, el cual se utiliza para tomar datos de varias entradas diferentes que se seleccionan mediante un multiplexor.

Con hasta 16 canales y puede transmitir 1[Msp] y 10 bits de longitud por muestra

Características del módulo:

- Tiene 16 entradas analógicas configuradas hacia las salidas del puerto A (RA0-RA15).
- Hasta 1Mbps velocidad de conversión.
- Entradas analógicas están multiplexadas con dos Mux.
- ADC de 10 bits conectado memoria de 16.

En la figura 20 se muestra el diagrama por etapas de módulo ADC el cual cuenta con un interruptor, comparador, operacional entre otros, todo ello con la finalidad de conocer su funcionamiento elemental.



Note 1: VREF+ and VREF- inputs can be multiplexed with other analog inputs.

Figura 20. Diagrama de bloques del ADC.

**Registros del ADC.-** Es trascendental conocer cada uno de los registros y así poder asociarle un valor para que puedan realizar las acciones e instrucciones que se desean llevar a cabo, para ellos se creara una tabla la cual contendrá los criterios utilizados y configurados en el PIC para realizar y activar el módulo ADC.

### 3.1.6 Descripción del timer

La familia de dispositivos PIC32MX tiene dos tipos diferentes de contadores de tiempo, dependiendo de la variante del equipo. Los temporizadores son útiles para generar eventos de interrupción precisos basados en los intervalos de tiempo periódicos para el software de aplicaciones o los sistemas operativos en tiempo real. Otros usos incluyen conteo de pulsos externos o medición de la sincronización exacta de los acontecimientos externos mediante característica puerta del temporizador. Con ciertas excepciones, todos los temporizadores tienen el mismo circuito funcional. Los temporizadores se pueden clasificar en dos tipos, a saber:

**Tipo A Timer** (temporizador de 16 bits síncrono /asíncrono/contador con puerta).

**Tipo B** Timer (16-bit, 32-bit de temporización síncrona/contador con puerta y Eventos Especiales Trigger).

Todos los módulos de temporizador incluyen las siguientes características comunes:

- 16-bit del temporizador / contador
- Fuente de reloj interna o externa seleccionable por software
- La generación de interrupciones programable y prioridad
- Recinto contador de pulsos externos

Aparte de estas características comunes, cada tipo de temporizador ofrece las siguientes características adicionales:

**Tipo A:**

- Temporizador asíncrono / contador con un oscilador integrado
- Operativo durante el modo bajo consumo de energía de CPU
- Prescalers seleccionables Software 1:01, 1:08, 1:64 y 1:256.

**Tipo B:**

- Posibilidad para configurar un timer de 32 bits de temporizador / contador
- Prescalers configuradas por software 1:01, 1:02, 1:04, 1:08, 1:16, 1:32, 1:64 y 1:256
- Capacidad de disparo Evento.

En la tabla 10 se resumen las características de los tipos de timer que tiene el PIC.

Tipos De Temporizador Disponibles	Oscilador Secundario	Reloj Externo Asíncrono	Reloj Externo Síncrono	16-Bit Síncrono Temporizador / Contador	32-Bit Síncrono Temporizador / Contador	Cerrada Temporizador	Especial Evento Trigger
Tipo A	Si	Si	Si	Si	No	Si	No
Tipo B	No	No	Si	Si	Si	Si	Si

Tabla 10. Proporciona las características de las funciones de temporizador.

### 3.1.7 Pantalla de cristal líquido (LCD)

La pantalla LCD es uno de los medios que se utilizara para observar el resultado del sistema, ya que en ella se desplegaran las temperaturas monitoreadas por el sensor cada minuto, debido a su composición y diseño ya que puede mostrar elementos alfanuméricos que serán captados por el sistema de monitoreo, y así tanto el personal como los posibles visitantes a la planta purificadora puedan observar los resultados generados.

Las pantallas de cristal líquido LCD tienen la capacidad de representar cualquier elemento alfanumérico permitiendo representar la información que genera cualquier dispositivo electrónico de una manera fácil y económica

Características de la pantalla táctil.

- Contenido de Display de 16 caracteres \* 2 líneas.
- Muestreo de 400 Hz.
- Contraste ajustable.
- Alimentación de 5[V].
- Cada carácter compuesto de 5 columnas\*8 filas.



Figura 21. Pantalla LCD.

Para conectar la LCD con el PIC es necesario conocer el diagrama de conexiones, por tal razón en la tabla 11 se especifica a que pines que se utilizaran para conectar la pantalla al microcontrolador, además que en la figura 22 muestra el diagrama de dichas conexiones.

Entradas	Entrada PIC 32
E	R <sub>D4</sub>
R/W	R <sub>D5</sub>
Rs	R <sub>B15</sub>
V <sub>c</sub>	V <sub>EE</sub>
V <sub>CC</sub>	3,3 [V]
GND	Masa
D <sub>B0</sub>	R <sub>E0</sub>
D <sub>B1</sub>	R <sub>E1</sub>
D <sub>B2</sub>	R <sub>E2</sub>
D <sub>B3</sub>	R <sub>E3</sub>
D <sub>B4</sub>	R <sub>E4</sub>
D <sub>B5</sub>	R <sub>E5</sub>
D <sub>B6</sub>	R <sub>E6</sub>
D <sub>B7</sub>	R <sub>E7</sub>

Tabla 11. Tabla de conexiones de la LCD.

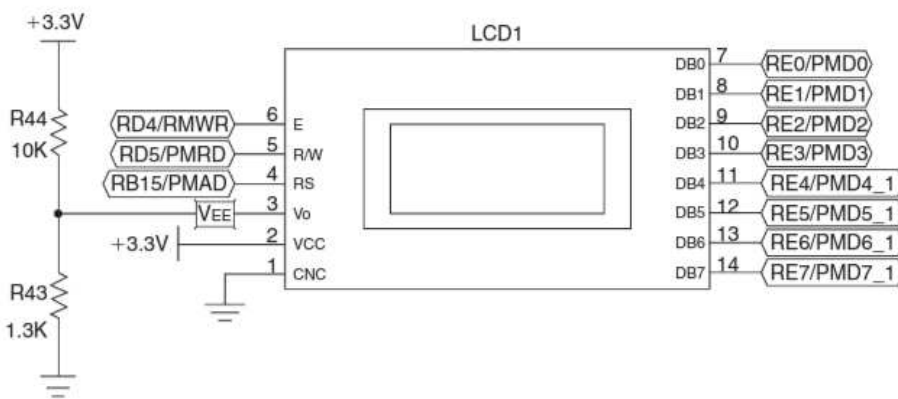


Figura 22. Diagrama de conexión de la LCD.

### 3.1.8 Stack TCP/IP Microchip

Se utilizó el Stack TCP/IP que ofrece Microchip, el cual contiene diversas librerías, es decir un conjunto de programas de aplicaciones estándar basadas en TCP/IP HTTP Server, cliente de correo, etc. [17].

Además de ofrecer las librerías que implementa los protocolos de comunicaciones hasta la capa de transporte, incluye ciertas aplicaciones de muestra. En la figura 23 se muestra el modelo TCP/IP con referencia a los protocolos del stack Microchip.

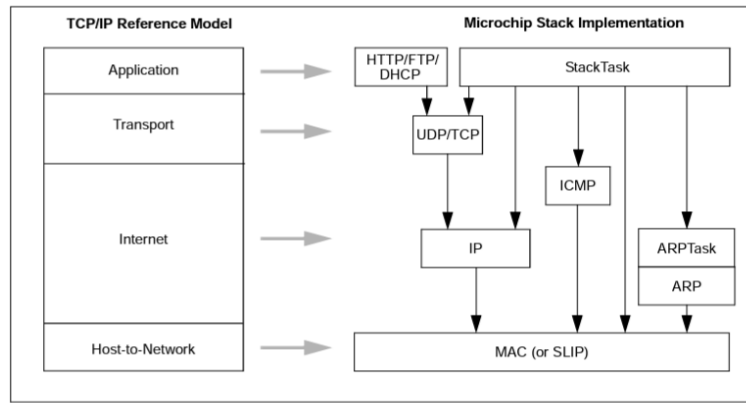


Figura 23. Stack para el microcontrolador.

La tabla 12 describe las librerías que proporciona la pila de protocolos que son empleas para el stack TCP/IP para cada uno de los protocolos.

Librerías de la cabeceras (.h)	Descripción
<b>TCP.h</b>	Funciones que suministran confiabilidad, ofreciendo transporte en la transferencia de aplicaciones de datos con control de flujo.
<b>TCPIP.h</b>	En esta cabecera se encuentran declaradas todos los (.h) del programa, permitiendo acceder a todas las cabeceras mediante una sola llamada.
<b>IP.h</b>	Estructuras y funciones del protocolo IP, usadas por el programa, para la comunicación de datos a través de una red de paquetes conmutados.
<b>ICMP.h</b>	Definiciones que establecen el control y notificación de errores del protocolo IP.
<b>DHCP.h</b>	Función que permiten a las estaciones de trabajo de una red IP obtener sus parámetros de configuración de manera automática.
<b>FTP.h</b>	Funciones que permiten la comunicación tipo FTP entre el usuario y el microcontrolador.
<b>ARP.h</b>	Funciones que permiten encontrar la dirección hardware o MAC correspondiente a una determinada dirección IP.
<b>GenericTCPServer.h</b>	Funciones que utiliza el stack periódicamente para escuchar las conexiones entrantes.
<b>MAC.h</b>	Estas funciones proporcionan acceso al controlador Ethernet del PIC32MX795F512L.
<b>Delay.h</b>	Declaraciones que establecen retardos específicos, que son de gran utilidad para el comportamiento del programa a realizar.

<b>SNTP.h</b>	Definiciones que actualizan el tiempo interno del programa cada 10 min usando la dirección "pool.net.org" que es un servidor de tiempo de red.
<b>Telnet. h</b>	Declaraciones que establecen los servicios de telnet.
<b>Stacktsk.h</b>	Funciones que permiten manejar los paquetes de recepción realizando un pre procesamiento para luego ser enviados a las capas superiores.
<b>TCPPerformance.h</b>	Funciones que realizan únicamente pruebas, para establecer una conexión, enviando imitaciones de paquetes de la memoria ROM.
<b>TFTPc.h</b>	Funciones que establecen servicios poco confiables de carga y descarga de archivos a otras aplicaciones.
<b>Tick.h</b>	Funciones útiles para a la administración del cronometro interno del microcontrolador, así como la inicialización de los relojes (Timers), conversión a escala de tiempo, entre otras.

Tabla 12. Descripción de las bibliotecas de stack que se utilizaran.

## 3.2 Programación del socket

La programación del socket requirió de la exploración de las librerías además de tener definidos los parámetros para la realización del socket, los cuales son: servidor, cliente y puerto.

### 3.2.1 Explicación del protocolo TCP en la PILA

TCP es un protocolo de capa de transporte estándar que se describe en el RFC 793. Proporciona conexiones basadas en flujo confiable sobre redes poco fiables, y constituye la base para HTTP, SMTP, y muchas otras normas de protocolo.

Las conexiones realizadas sobre la transferencia de datos TCP ganan garantía a expensas de rendimiento.

- Las conexiones se realizan a través de un triple proceso de apretón de manos, lo que garantiza una conexión uno-a-uno. Los nodos remotos anuncian la cantidad de datos que están listos para recibir, y todos los datos transmitidos deben ser acreditados.
- Si un nodo remoto no reconoce la recepción de datos, es retransmitida automáticamente. Esto asegura que los errores de red, como perdido, dañado o paquetes fuera de orden son corregido automáticamente.

Para Microchip TCP IP Stack, la aplicación debe volver al bucle principal de pila en orden para que esto suceda. Del mismo modo, el nodo remoto no puede transmitir más datos hasta que el dispositivo local haya recibido el acuse de recibo y que el espacio esté disponible en la memoria intermedia. Cuando una aplicación local necesita leer más datos, debe volver al bucle principal de pila y esperar a que llegue un nuevo paquete.

El diagrama de flujo de TCP a continuación proporciona una visión general del uso del módulo de TCP.

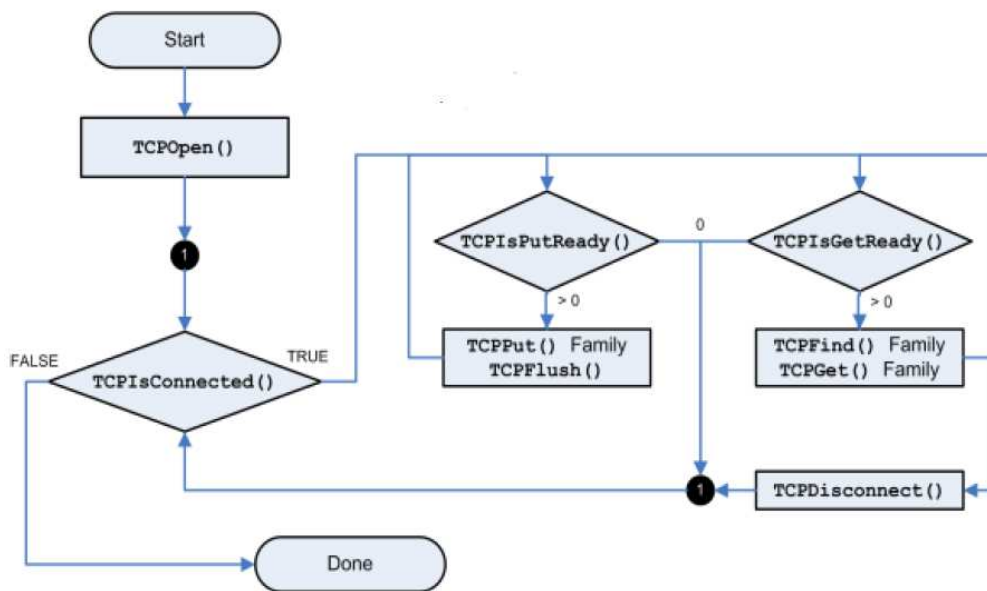


Figura 24. Diagrama del socket TCP.

Para realizar el programa **GenericTCPClient** es necesario conocer algunas funciones que nos permitirán realizar de manera completa la conexión del socket, ya que cada una de ellas cumple con un propósito en general que permiten que se lleven a cabo de manera satisfactoria cada una de las tareas predeterminadas. A continuación la descripción de cada uno de ellas.

Los conectores se abren con **TCPOpen** con esta función se puede abrir un socket de escucha a esperar a que las conexiones de cliente, o puede hacer una conexión del cliente al nodo remoto.

La función que nos permitirá escribir datos es la siguiente **TCPisPutReady** para comprobar cuánto espacio está disponible. Luego, llame a cualquiera de los TCPPut familia de funciones para escribir datos cuando haya espacio disponible.

La función **TCPFlush** es para transmitir datos de forma inmediata. Como alternativa, regresar al bucle principal de pila.

La función **TCPisGetReady** es para llamar datos y para determinar cuántos bytes están listos para ser recuperados. A continuación, utilice el TCPGet familia de funciones para leer los datos de la toma, y/o el TCPFind familia de funciones para localizar los datos en el búfer. Cuando se terminan los datos, el retorno al bucle principal de pila de espera de más datos para llegar.

Función para cerrar una conexión dentro del socket **TCPDisconnect**, volver al bucle principal de pila y esperar a que el nodo remoto para reconocer la desconexión. Sockets de cliente volverá al estado de reposo, mientras sockets de escucha esperarán a una nueva conexión.

### **3.3 Pasos para construir un proyecto con el stack TCP/IP**

Hay varios archivos de base que deben ser incluidos en todos los proyectos que utilizan la pila TCP/IP de Microchip. Por lo que enseguida se observa una breve descripción de las cabeceras las cuales son prototipos de las funciones.

Librerías (.h) (.c)	Descripción
ARP.c y ARP.h	Estos archivos son utilizados por la pila para descubrir la dirección MAC asociada a una determinada dirección IP.
Delay.c y delay.h	Estos archivos se utilizan para proporcionar los retrasos para algunas funciones de la pila. Tenga en cuenta que lo mejor sería que No utilice estos retrasos en su propio código, ya que crean condiciones de bloqueo.
Archivos de la capa física	Estos archivos se utilizan para habilitar una capa física especificada. Más información sobre los archivos que desea incluir se puede encontrar en la configuración de hardware.
Helpers.c y Helpers.h	Estos archivos contienen funciones auxiliares utilizadas para tareas de pilas diversas.
IP.c y IP.h	Estos archivos proporcionan funcionalidad de capa de Internet para la pila.
StackTsk.c y StackTsk.h	Estos archivos contienen el código para inicializar la pila y realizar las devoluciones de llamada que mantienen la pila va.
Tick.c y Tick.h	Estos archivos implementar un temporizador de garrapata que se utiliza para implementar algunas funciones de temporización dentro de la pilar.
HardwareProfile.h	Este archivo de configuración se utiliza para configurar las opciones de hardware.
TCPIPConfig.h	Este archivo de configuración se utiliza para configurar las opciones de firmware.
MAC.h	Este archivo de encabezado proporciona macros y estructuras relativas a la capa MAC de hardware.
TCPIP.h	Este es el principal archivo de inclusión para la pila. El archivo principal debe incluir TCPIP.h.

Tabla 13. Descripción de las funciones (.h).

### 3.3.1 Inicialización de la pila para socket TCP

Se comenzara la inicialización del hardware. Esto incluye pasadores de PPS, osciladores, LEDs, LCDs, cualquier SPI o módulos PMP está utilizando para controlar el chip MAC/PHY hardware, etc. Enseguida se mencionaran los pasos que se deben llevarse a cabo para realizar la inicialización de la pila TCP/IP:

- Llamar a las funciones de inicialización de hardware para la biblioteca.

- TickInit se debe llamar primero.
- Se inicializará el temporizador de Tick que administra su tiempo pila.
- Luego llame a cualquiera de las funciones de inicialización adicionales que requieren hardware inicialización.
- La mayor parte de las variables de la aplicación relacionados con pila-son almacenada en el AppConfig estructura.
- En este punto, usted debe inicializar la estructura AppConfig con sus valores por defecto, o proporcionar otro medio de inicializar la estructura AppConfig.
- Por último, se puede inicializar la pila llamando a la función StackInit (). Esta función llamará automáticamente la inicialización funciones para otros protocolos de firmware si se han habilitado en TCPIPConfig.h (es decir TCPIInit el **Protocolo TCP**, HTTPInit para HTTP2. Después StackInit () ha sido llamado, puede llamar a otra específica de la aplicación funciones de inicialización del firmware [17].

### 3.3.2 Función de inicio

Una vez que el programa se ha inicializado, debe escribir un bucle infinito que se encargará de las tareas de dicha aplicación. Dentro de este bucle, hay dos funciones importantes que deben ser incluidas y las cuales son: **StackTask** y **Stack Aplicaciones**.

La función **StackTask** se encargará de la transmisión y la recepción de paquetes de datos. Esta función también ruta los paquetes que se han recibido a la aplicación correspondiente función de nivel de protocolo para manejarlo.

La función **StackApplications** llamará a los módulos de aplicación cargados. Por ejemplo, si una aplicación está utilizando un HTTP2 servidor, StackApplications llamarán automáticamente el HTTPServer función para procesar cualquier tarea HTTP2 que están en cola.

La mayoría de las subtareas dentro pila tarea y aplicaciones se implementan como funciones multitarea cooperativa Estado-máquina controlada. Dado que estos sub-

tareas consiste en varios pasos (que puede ocurrir en distintos momentos) este sistema de devolución de llamada se asegura de que no es una tarea solo se monopolizar el control del procesador.

Dentro de este bucle principal, es posible que también desee sondear para cualquier cambio de E/S en su aplicación y llamar a las tareas específicas de la aplicación que ha implementado. Para evitar causar desbordamientos de búfer en el hardware o el protocolo violaciones ritmo al que debe tratar de poner en práctica sus propias tareas de aplicación en las funciones de devolución de llamada con desencadenantes basados en la sincronización. Un método para hacer esto se describe en el tema siguiente.

Llamando StackTask muy poca frecuencia podría limitar su rendimiento, sin embargo, ya que cada llamada de StackTask puede recuperar un paquete (como máximo) de la memoria intermedia de paquetes. Del mismo modo, las tareas de aplicación que son dependientes del tiempo (como una respuesta ICMP ping) pueden producir resultados no deseados si StackApplications no se llama con frecuencia suficiente.

La cantidad de tiempo que el bucle principal tarda en completar una iteración depende de varios factores. Si los datos están listos para ser transmitida, o si se recibe un paquete de datos recibido, la función StackTask tomará más tiempo de lo que sería de otro modo. Cada protocolo adicional incluida en su aplicación hará que el bucle principal para tomar un tiempo adicional, así, con la cantidad de tiempo para cada variación de la longitud del estado de la máquina estatal más corto en la tarea de la más larga.

### 3.3.3 **Multitarea Cooperativa**

Si implementa la pila TCP/IP utilizando un enfoque multitarea cooperativa, debe realizar llamadas periódicas a las funciones de tareas para transmitir/recibir paquetes y para mantener la funcionalidad de protocolo. Para evitar conflictos con la pila, se debe escribir las tareas propias de una manera que les permita dar el procesador si no se necesita. Si crea una tarea o aplicación protocolo con múltiples pasos, puede ser beneficioso para dividirlos entre los estados [17].

# CAPÍTULO 4

*Desarrollo e Implementación de las interfaces del sistema.*

## 4 Desarrollo e Implementación de las interfaces del sistema

En este capítulo se hace mención del desarrollo que se llevó a cabo para la programación de la tarjeta ETHERNET (PIC 32 Ethernet) en cada uno de sus módulos.

### 4.1 Módulo ADC

Para poder programar el convertidor analógico-digital es necesario conocer sus etapas en figura 25 se observan a groso modo sus bloques. Además el modulo cuenta con sus propios registros mismos que se mencionaran a continuación.

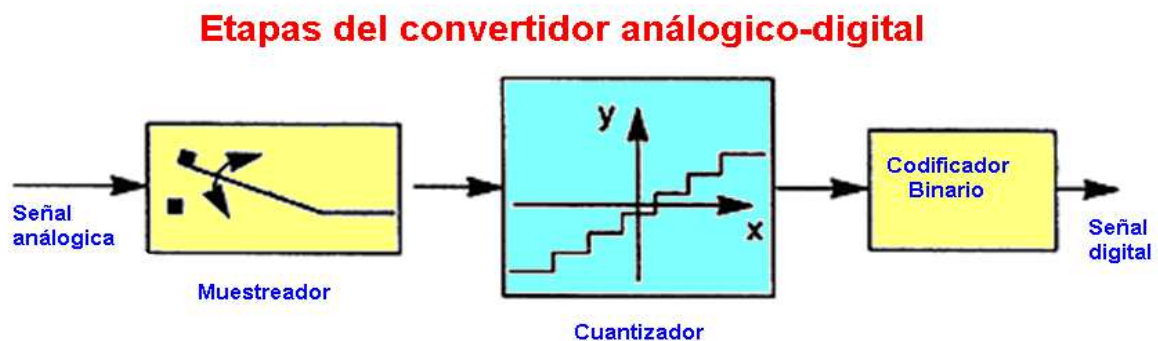


Figura 25. Diagrama de bloques del convertidor analógico-digital.

- AD1CON1: ADC Control Registro 1.
- AD1CON2: ADC Control Registro 2.
- AD1CON3: ADC Control Registro 3.
- AD1CHS: ADC Selección de entrada de registro.
- AD1PCFG: ADC Puerto registro de configuración.
- AD1CSSL: ADC Escaneado selección de entrada de registro.

Los registros de configuración son de 32 bits, en la figura 26 se muestra el diagrama de entradas del ADC, como se puede apreciar son 16 entradas, solo se ocupara una debido a que solo se colocara un sensor, en este caso en particular utilizara la AN2, esto está indicado en la configuración ya que se define la variable TSENS 2.

**#define TSENS 2** // En esta parte se selecciona la entrada a utilizar en este caso es será la entrada A2 correspondiente al 32 j11.

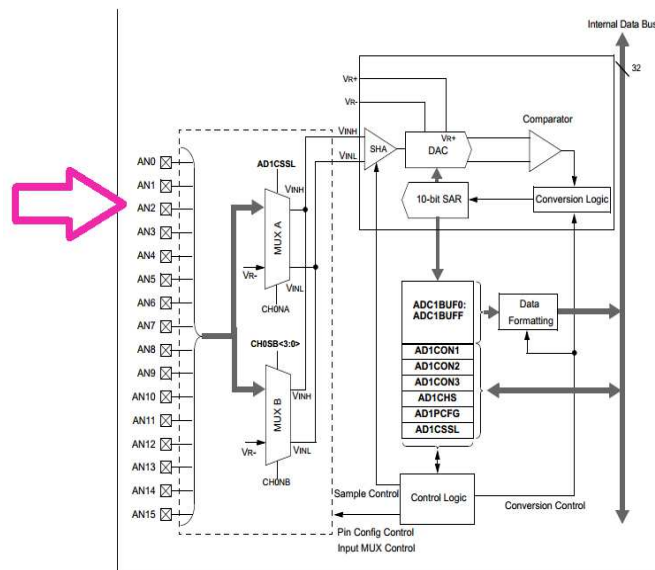


Figura 26. Diagrama que muestra las entradas del convertidor.

Se realizó la configuración de los registros del módulo ADC, los cuales están compuestos por grupos de 16 bits cada uno, mismos que cumplen con una función específica, por lo cual fue sustancial analizar a detalle cada uno, en la figura 27 se observa el valor asignado para cada bit es decir se tiene la opción de “0” o “1”, anteriormente se hizo alusión a cada uno de los registros a continuación solo se muestra el valor que se les dio a cada bit, después de asignarle dicho valor, se hizo una conversión a valor hexadecimal, pues en el programa del ADC se colocara de esta manera.

Por cada 4 bits se hizo la conversión posteriormente esto será asignado utilizado en la sintaxis del programa, para hacer notar esto en seguida se mostrarán algunas líneas del programa para hacer notar la aportación de la tabla y la configuración de bits.



$$\text{Voltaje por grado centigrado} = 10 \left[ \frac{mV}{^{\circ}C} \right]$$

$$\text{Factor de escala} = 100 \left[ \frac{V}{^{\circ}C} \right]$$

$$\text{Resolucion}_{ADC} = \frac{\text{Maximo voltaje} * \text{Factor de escala}}{\text{Maximo numero}} \dots \dots \dots (1)$$

Lo obtenido de la ecuación (1) nos indicara valor digital proporcional al voltaje analógico. Se tiene que  $2^n - 1$

$$\frac{1.5[V] \times (100)}{511} = 0.2935 X \dots \dots \dots (2)$$

En la ecuación (3) se toma en cuenta la parte del amplificador no inversor, es decir se divide entre la ganancia que es 3.3, esto con la finalidad de poder tener el valor adecuado en la resolución del convertidor analógico digital y X será el valor analógico que se ha tomado por la entrada del sensor, cada una de la tomas de muestra pasaran por dicha ecuación y así poder ir construyendo la tabla de valores que se almacenarán en una base de datos dentro de un servidor.

$$\frac{0.2935X}{3.3} = 0.0889X \dots \dots \dots (3)$$

#### 4.2 Librerías para la pantalla LCD

Cómo paso previo a la implementación de la comunicación PIC-red LAN, se utilizó una pantalla LCD como salida del PIC. Esto con la finalidad de probar el módulo ADC y la captura de datos desde el sensor.

Para activar la LCD, se creó un proyecto el cual, estuvo compuesto por librería con extensión (.c) y (.h)

Para la configuración y pruebas de la pantalla LCD, se hizo:

- Explore.h
- Explore.c
- LCD.h
- LCDlib.h

Después de tener en cuenta todas las conexiones para la LCD, se hizo la prueba de manera todo ello con el propósito de poder observar que prendía y así poderla incorporar al programa final que contenga, la función de configuración así como las librerías pertinentes que hacen que se active y funcione.

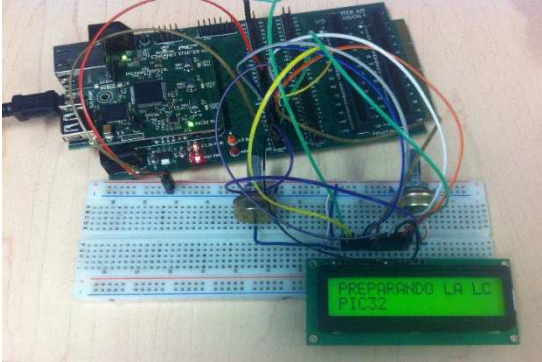


Figura 29. Resultado de la conexión de la LCD.

### 4.3 Programación del timer

En la figura 30 se observa el bosquejo esquemático del timer a emplear, además de que se muestran sus componentes mismos que serán controlados a través de sus registros.

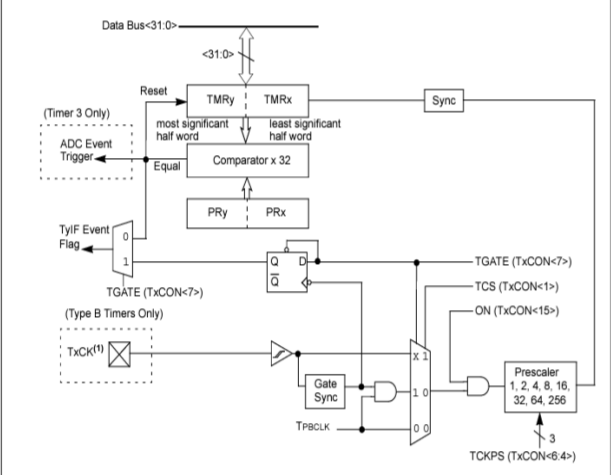


Figura 30. Temporizador de tipo B (32 bits).

Como se puede observar en el diagrama se utilizaron dos timer de 16 bits, enseguida se colocaran los principales registros del Timer, cada uno de ellos se asignó de acuerdo a nuestras necesidades.

```
T4CON=0x0; //
```

```
T5CON=0x0; //
```

```
T4CONSET=0x0038; // Activar el modo de 32 bits, prescaler 1:8, Fuente de reloj interna periférica
```

```
TMR4=0x0; // limpiar los contenidos del TMR4 y TMR5
```

```
PR4=0x7FFFFFFF; // Carga PR4 PR5 y registra con valor de 32 bits
```

```
T4CONSET= 0x8030; // Iniciar timer 45
```

#### **4.4 Programación de la PIC 32**

Para la programación del Microcontrolador, se tomó como referencia la versión de programación TCP/IP que facilita el fabricante de manera libre. Para que el programa cumpla una de las normas de la programación “todo sistema debe ser estructurado”, por lo que se ha dividido el código en funciones que serán utilizadas para el funcionamiento la aplicación.

En una primera fase, las temperaturas medidas se almacenarán en un microcontrolador (PIC32MX795F512L) que será el encargado de realizar la conversión de analógico al digital ya que cuenta con un módulo ADC. Así mismo, el PIC cuenta con un timer qua ayuda a realizar la parte de control de tiempo. Se utilizó una LCD para mostrar el comportamiento del sensor.

Como siguiente etapa se utilizó un router Cisco Linksys, esto con la finalidad de crear una pequeña red LAN, la cual se configuró por medio de su interfaz gráfica.

Se programó el socket necesario para comunicar el PIC y la red local. La programación se realizó basándonos en un demo producido por el fabricante del microcontrolador.

Una vez funcionando las dos etapas anteriores, se modificaron y cambiaron algunos parámetros tales como: se le colocó seguridad WAP, la configuración de red LAN, etc. Se ocuparán 4 direcciones: una para el servidor (192.168.1.202) que se configuró estática, así como la dirección de la máquina que controla el comportamiento del servidor y del microcontrolador (192.168.1.102), la del PIC32 será estática (192.168.1.97), y los equipos que se llegaran a conectar se les asignará una dirección IP vía DHCP.

Así mismo, se montó un servidor Linux distribución Ubuntu 10.04LT, en el que se instaló un servidor Apache, en el que se instalaron mysql y php5, cada uno de ellos con un fin específico en el caso de mysql fue para realizar la base de datos, en la cual almacenaremos todas las temperaturas que vaya marcando el sensor cada minuto, además de que se crearon scripts en php para jalar la base de datos y por último el servidor cumple la función como medio para colocar y montar páginas y observarlas vía web.

#### 4.5 Librerías de stack Microchip

Microchip ofrece las librerías de la pila TCP/IP. Algunas de las cuales son: TCPIP stack (.h) y (.c), Wi-Fi, Common, estas a su vez se encuentran formadas por archivos que forman las librerías del stack de la pila TCP, en ellos están declarados los diferentes protocolos de comunicación cada uno de ellos con sus diversos parámetros, y declaraciones de funciones, extensiones, definiciones, condicionales, etc, esto con el rigor de ser utilizadas para fines en concreto [13].

Librerías stack (.c)	Librerías stack (.h)
Announce, ARP, ARCFOUR, AutoIP, BerkeleyAPI, Delay, BigInt, DHCP, DHCPs, DNS,DNSs, DynDNS, ENC28J60, ENCX24J600, ETHPIC32ExtPhy, ETHPIC32ExtPhyDP83848, etc.	Telnet, Tick, UART, TCP, TCPIP, SNTP, SSL, MAC, SMTP, StackTsk, LCDBlocking, ICMP, IP, MPFS2, NBSNS, HTTP2, etc.

Tabla 14. Ejemplos de las librerías del stack.

En los siguientes apartados se hará hincapié y se explicaran algunas de las librerías de Stack a las cuales se les hicieron modificaciones, esto con el fin de observar su comportamiento.

#### 4.5.1 HWP PIC32\_ETH\_ETH795.h

En esta librería están definidos algunos registros que serán ocupados por dispositivos externos, en este caso la LCD [14].

```
//DEFINICIÓN DE REGISTROS DE ACTIVACION DE LA LCD
#define LCD_DATA_TRIS      (*((volatile unsigned char*) &TRISE))
#define LCD_DATA_IO       (*((volatile unsigned char*) &LATE))
#define LCD_RD_WR_TRIS    (TRISDbits.TRISD5)
#define LCD_RD_WR_IO      (LATDbits.LATD5)
#define LCD_RS_TRIS       (TRISBbits.TRISB15)
#define LCD_RS_IO         (LATBbits.LATB15)
#define LCD_E_TRIS        (TRISDbits.TRISD4)
#define LCD_E_IO          (LATDbits.LATD4)
```

#### 4.5.2 TCPIP ETH795.h

En esta librería se encuentran los parámetros de configuración de red.

```
#define STACK_USE_GENERIC_TCP_CLIENT_EXAMPLE //HTTPClient example
in GenericTCPClient.c
```

Direccionamiento de red opciones: Al PIC se le coloco una dirección IP estática, configurando los parámetros que aparecen a continuación [16].

```
#define MY_DEFAULT_IP_ADDR_BYTE1    (192ul)
#define MY_DEFAULT_IP_ADDR_BYTE2    (168ul)
#define MY_DEFAULT_IP_ADDR_BYTE3    (1ul)
#define MY_DEFAULT_IP_ADDR_BYTE4    (97ul)
```

Para verificar si el PIC tiene conectividad con el servidor se realizó un ping y la respuesta es la que se aprecia en la figura 31, en ella se observa que el tiempo de espera y los paquetes recibidos.

```
root@yadhiraserver:~# ping 192.168.1.97
PING 192.168.1.97 (192.168.1.97) 56(84) bytes of data.
64 bytes from 192.168.1.97: icmp_seq=1 ttl=100 time=0.177 ms
64 bytes from 192.168.1.97: icmp_seq=2 ttl=100 time=0.160 ms
64 bytes from 192.168.1.97: icmp_seq=3 ttl=100 time=0.164 ms
64 bytes from 192.168.1.97: icmp_seq=4 ttl=100 time=0.149 ms
64 bytes from 192.168.1.97: icmp_seq=5 ttl=100 time=0.152 ms
64 bytes from 192.168.1.97: icmp_seq=6 ttl=100 time=0.156 ms
64 bytes from 192.168.1.97: icmp_seq=7 ttl=100 time=0.158 ms
64 bytes from 192.168.1.97: icmp_seq=8 ttl=100 time=0.161 ms
64 bytes from 192.168.1.97: icmp_seq=9 ttl=100 time=0.166 ms
^C
[1] + Stopped                  ping 192.168.1.97
root@yadhiraserver:~#
```

Figura 31. Respuesta de un ping para comprobar la asignación de IP.

### 4.5.3 **HardaweProfile.h**

En esta Liberia se selecciona el PIC, con el que se va a trabajar [16].

```
#if defined (YadhiraPIC)
```

```
#include "Configs/HWP YOUR_BOARD.h"
```

- #elif defined(CFG\_INCLUDE\_PIC32\_ETH\_SK\_ETH795)
- // DM320004 PIC32 Ethernet Starter Kit, PIC32MX795F512L, internal Ethernet (National DP83848 PHY)

```
#include "Configs/HWP PIC32_ETH_SK_ETH795.h"
```

### 4.5.4 **GenericTCPClient.c**

En esta librería se encuentra el ejemplo del socket TCP, el cual cuenta con funciones que permiten realizar una conexión con un servidor. Es decir esta función implementa un sencillo cliente de HTTP, que opera a través de TCP.

### 4.5.5 **Main.h**

En main.h se deshabilitaron todas las funciones que no se iban a ocupar tales como PingDemo, GenericTCPServer, SMTDemo, etc, solo se dejó la de GenericTCPClient ya que es la única que utilizara el proyecto.

Listado de funciones para los distintos protocolos y socket.

```
//void PingDemo(void);
```

```
void GenericTCPClient(void);
```

```
//void GenericTCPServer (void);
```

## 4.6 Realización del SOCKET TCP

En este apartado se hará mención de un ejemplo de socket cliente con el cual se hicieron pruebas, para después posteriormente adaptarlo a nuestras necesidades. Para realizar la programación de socket, como primer paso se definió quienes serían: el cliente, el servidor y el puerto. En el caso cliente es el Microcontrolador PIC 32, el puerto fue escogido de manera arbitraria fue el 3550, para implementar y configurar el socket fue necesario realizar intentos analizando las diferentes librerías con las que cuenta principalmente: GenericTCPCliente.c, Main.h y Main.c. Enseguida se explicara los cambios hechos en cada una de ellas, además de que se quitaron algunas librerías y deshabilitaron con la finalidad de lograr una conexión exitosa entre el microcontrolador y el servidor.

Lo primero fue realizar un proyecto con el nombre de socket cliente, se quitaron librerías (.c). Se ingresó a la librería GenericTCPCliente.c, se colocó IP de servidor el cual es 192.168.1.202, además del puerto.

En estas líneas de código se puede apreciar dicha información.

- `static BYTE ServerName[] = "192.168.1.202" // Configuración del servidor`
- `static WORD ServerPort =3550; // Configuración del Puerto`

Además se colocó texto mismo que mostrara el servidor una vez que se halla conectado algún dispositivo de la siguiente manera.

- `TCPPutROMString(MySocket, (ROM BYTE*)"Conectado al PIC32MX795F512F\n ");`
- `TCPPutROMString(MySocket, (ROM BYTE*)"Te enviare las temperaturas del sensor LM35 cada minuto \n");`

Además de que se utilizaron funciones de un socket mencionadas anteriormente tales como: TCPOpen, TCPDisconnect.

En la función main.c, se deshabilitaron, funciones de los diferentes protocolos tal como: BerkeleyTCPClientDemo, BerkeleyTCPServerDemo, GenericTCPServer,

etc., además de que se quitaron funciones del módulo Wi-Fi, y algunas definiciones y propiedades de otros PIC que se encontraban declarados.

#### **4.7 Diagrama de flujo para el proyecto**

Comencemos con la descripción en general de los procesos que se realizarán en el microcontrolador, lo primero que se hizo fue el planteamiento del problema en general por medio de bloques en el que se colocan los procesos de manera muy general, es decir no se describen con detalles cada uno de los procesos, mismos que deberán ser descritos, posteriormente, la explicación de este conlleva a hacer mención de los procesos inscritos dentro del diagrama. En la figura 32 se observa el diagrama flujo y a continuación se hace una breve descripción de los procesos.

1. Lo primero será inicializar los procesos tanto como el LCD, Timer y ADC.
2. Se deberá hacer la lectura del sensor de temperatura con la finalidad de registrarla.
3. Se deberá mostrar el registro de la temperatura en la LCD.
4. Se debe verificar si la temperatura excede los 55° [C], si es así se activará un led y una alarma.
5. Se hace la inicialización del Socket TCP en este, se verifica el puerto de conexión así como el servidor al que se conectará, que en este caso es el que tiene la dirección 192.168.1.202
6. Se deberá realizar la transmisión por medio del socket utilizando la tecnología Ethernet, y se deberán guardar en una base de datos, misma que se creará con mysql dentro del servidor y que se llamará por medio de los script de php.
7. Se pretende mostrar el estatus de la conexión del socket mismo que se deberá mostrar en la LCD, pues así se comprobará la conexión que existe en el mismo.
8. Después de hacer la transmisión, el sistema se esperará 1 minuto para volver hacer el registro de la siguiente muestra de temperatura del sensor y así volver a realizar todo el proceso.

9. Para que todo esto se lleve a cabo la condición sobre la que radica es que la planta este encendida, ya que de no ser así no se podría llevar a cabo nada de lo que se comentó y es ahí donde termina el diagrama de flujo.

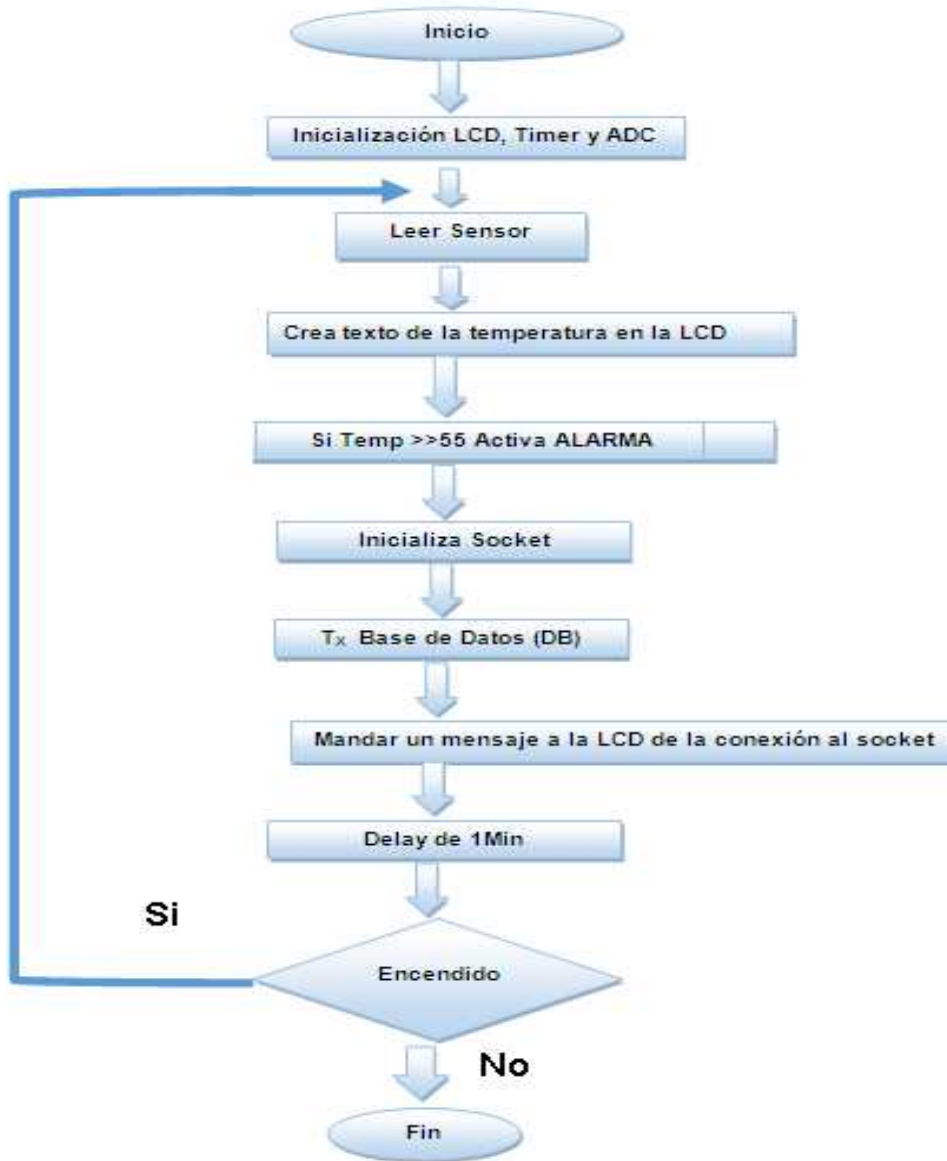


Figura 32. Diagrama de flujo del proyecto.

#### 4.8 Creación del proyecto

Se creó un proyecto llamado TCP/IP, el cual está compuesto por una serie de librerías, cada una de ellas se encuentran ligadas a la pila TCP, por tal motivo es necesario y de vital importancia conocer la composición de dicho programa ya que

cada una de ellas contiene funciones que en conjunto nos permiten una clara conexión con cada uno de los procesos involucrados.

Por tal motivo el stack de microchip nos permite y nos proporciona las herramientas necesarias para poder realizar de manera adecuada cada proceso, además de que cuenta con las funciones que deberán ser invocadas para establecer la conexión, ya que uno de los principales motivos de realizar esto es la implementación de un Socket TCP, ya que este nos permitirá establecer un punto de comunicación mismo que a su vez permitirá trasladar los datos de la temperatura de nuestro sensor por medio del Ethernet.

El proyecto estará compuesto por programas y librerías del stack pero solo se tendrá manipulación de algunas pues todas las demás se dejaran intactas y solo se harán la modificación de algunas de ellas que son mismas que son parte del stack de Microchip:

- Main.c
- Main.h
- GenericCliente.c

Estas son librerías son parte del stack de las cuales se hará modificaciones con la finalidad de poder cumplir con nuestro objetivo, que es la implementación del socket cliente, además que se incorporaran las correspondientes al ADC, Timer y LCD.

- ADCLib.c
- Explore.c
- LCDlib.c
- ADC.h
- LCD.h
- Explore.h

El microcontrolador lleva a cabo algunas tareas específicas, mismas que se irán describiendo conforme el desarrollo del trabajo, como primer instancia se realizó un diagrama de flujo, el cual nos permitió describir el algoritmo que se utilizara. Ya

que este consiste en una representación gráfica de distintos procedimientos lógicos, que tiene como finalidad brindar una simplificación y comprensión de éstos.

En el PIC 32, se implementaran diferentes módulos tales como: el ADC, el Timer y además que se levantó un socket TCP, mismos que llevan implícitos ciertos procesos, así como rutinas, registros, comandos etc. Para los cuales es importante considerar y analizar todos sus aspectos prácticos, su comportamiento, por lo cual para cada uno de ellos se hizo una descripción precisa con la finalidad de observar su uso, el comportamiento que realizaran junto con todo el sistema en conjunto. En el anexo del trabajo recepcional se muestra parte del código de la librería cliente socket (.c), en ella la mayor injerencia de acuerdo a nuestros objetivos además de que los cambios realizados fueron de acuerdo al diagrama de flujo planteado en la figura 32.

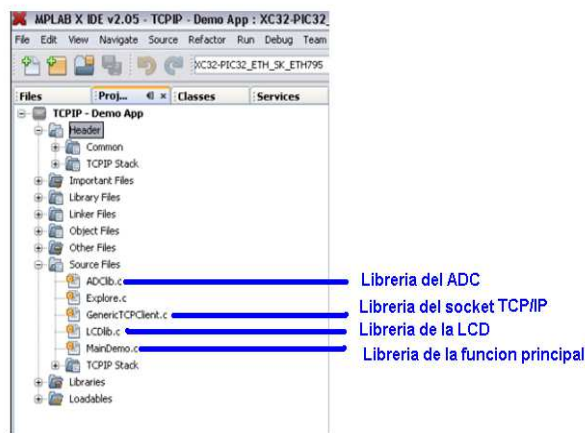


Figura 33. Creación del proyecto.

#### 4.9 Http 1.1 para la transferencia y captura en base de datos

Se abrió un socket TCP utilizando el protocolo http 1.1 con el método POST, ya que este método nos permite guardar elementos en un servidor y una de las razones es guardar los valores de temperatura monitoreados por el microcontrolador en una base de datos. Con el fin de tener registrados los cambios de temperatura del agua por cada minuto.

Se implementó el protocolo http 1.1, enseguida se mostraran las líneas de código que deberán colocarse en el request, todo ello con la finalidad de dar hincapié a las características que debe contener la sintaxis del protocolo los campos son: servidor, método POST, el tipo de conexión, etc.

Request

```
strcpy (mensaje, "POST /10sep/recibedatos.php HTTP/1.1\r\n");
strcat (mensaje, "Host: 192.168.1.202\r\n" );
strcat (mensaje, "Connection: keep-alive\r\n" );
strcat (mensaje, "Accept: text/html, */*\r\n");
strcat (mensaje, "Accept-Language: en-us\r\n" );
strcat (mensaje, "Content-Type: application/x-www-form-urlencoded\r\n" )
sprintf (cad1, "sensor=\"TEMP\"&temperatura=%04.1f\r\n",z);
sprintf (cad2, "Content-Length: %d\r\n", strlen(cad1) )
strcat (mensaje, cad2 );
strcat (mensaje, "\r\n");
strcat (mensaje, cad1 );
```

#### **4.10 Creación de la red LAN**

Se creó una red LAN con topología de estrella, que es una disposición física en la que se conectan los nodos de una red de computadoras o servidores, mediante la combinación de estándares y protocolos mismos que son establecidos por las personas que crean la red [8].

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Fácil inserción de nuevos elementos.</li> <li>• Alta seguridad.</li> <li>• Fácil detección de nodos con fallos.</li> <li>• Direccionamiento entre nodos sencillo y si el nodo central es activo.</li> <li>• Posibilidad de múltiples protocolos.</li> <li>• Posibilidad de introducir jerarquías en la prioridad de tramas.</li> </ul>	<ul style="list-style-type: none"> <li>• Un fallo en el nodo central bloquea las comunicaciones.</li> <li>• Si el nodo es activo se retrasa el tráfico.</li> <li>• Las ampliaciones están sujetas a la capacidad del nodo central, si la exceden aumentan mucho los costes.</li> </ul>

Tabla 15. Ventajas y Desventajas.

Como la arquitectura de la red nos permite establecer cómo es que vamos a colocar todos nuestros dispositivos. En la figura 34 se muestra la distribución de topología en estrella que se utilizó para la red y en la cual se puede observar: el servidor, la PC, router y el microcontrolador. Además de que se pueden conectar otros dispositivos al router por medio de cable o vía inalámbrica y si la IP se asignara por DHCP.



Figura 34. Esquema de la topología implementada.

Como se mencionó anteriormente se creó una red LAN misma que se configuro en el router, en la figura 35, se muestra en una tabla las entradas del enrutamiento, así como las características tales como: IP, mascara de red, puerta de enlace y por último la interfaz que se crea según él sea el caso. Además podemos corroborar que se dio de alta una red LAN e inalámbrica para todos aquellos dispositivos que se desean conectar.

Routing Table - Google Chrome  
192.168.1.1/RTable.htm

Lista de entradas de tabla de enrutamiento Actualizar

IP de LAN de destino	Máscara de subred	Puerta de enlace	Interfaz
0.0.0.0	0.0.0.0	172.17.151.254	WAN (Internet)
172.17.0.0	255.255.0.0	172.17.151.254	WAN (Internet)
192.168.1.0	255.255.255.0	192.168.1.1	Red LAN e inalámbrica

Cerrar

Figura 35. Tabla de enrutamiento de router.

#### 4.10.1 Configuración del router Cisco

Para el envío de datos se empleó la tecnología Ethernet, ya que la tarjeta cuenta con el puerto Ethernet, lo que nos permite que exista la conexión alámbrica.



Figura 36. Router físico.

En la figura 37 se pueden apreciar los parámetros de configuración de internet mostrando: tipo de conexión, dirección IP, máscara de red, puerta de enlace y DNS, también se muestra la información del enrutador como: Versión de firmware, verificación del firmware, dirección MAC y nombre del enrutador, las cuales son algunas de las características del router.

Información del enrutador	
Versión del Firmware:	v8.00.7 build 011, 27 Jul, 2009
Verificación del firmware:	37A3ED5F448F7BF59CDDABA7E44C5597
Hora actual:	No disponible
Dirección MAC:	00:22:19:1F:CD:62
Nombre del enrutador:	WRT54G
Nombre de host:	
Nombre de dominio:	
Internet	
Tipo de configuración	Tipo de conexión: Static
	Dirección IP: 172.17.151.21
	Máscara de subred: 255.255.0.0
	Puerta de enlace predeterminada: 172.17.151.254
	DNS 1: 172.17.102.14
	DNS 2: 172.20.5.253
	DNS 3:
	MTU: 1500

Figura 37. Muestra el estado del enrutador

La configuración avanzada nos permite, escoger el modo de enrutamiento que se le asignara al router y que en este caso se colocó como puerta de enlace, En la figura 38 se puede observar detalles tales como IP de LAN de destino, mascara de red, puerta de enlace predeterminado así como la interfaz.

Figura 38. Configuración avanzada.

La configuración de internet también radica en elementos que deben ser asignados de manera correcta, para así poder establecer la conexión a internet a través de toda la red local que se creó. En la figura 39 se pueden observar parámetros de clasificación de la red interna de la UACM.

Figura 39. Parámetros de la red.

Por último se puede observar el estado de la red local que se creó para todos aquellos dispositivos que se quieran conectar, en esta se pueden observar los parámetros tales como: Dirección MAC, Dirección IP, Mascara de red, Servidor DHCP, Dirección IP inicial y Dirección IP final.



Figura 40. Parámetros de la red.

Se configuro la red inalámbrica, con la finalidad de que aquellos dispositivos que cuenten con la tecnología de WI-FI puedan conectarse a la red LAN, en este caso teléfonos móviles. Se establecieron normas de seguridad para el acceso a la red en la figura 41, Se puede observar dicha configuración para ello se debe acceder a Inalámbrica—Seguridad Inalámbrica.

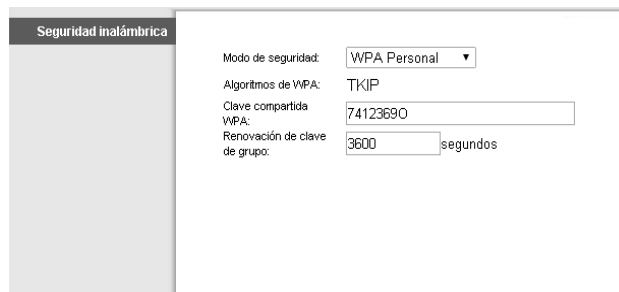


Figura 41. Seguridad WAP.

#### 4.11 Montaje del Servidor distribución Linux (Ubuntu 10.04 LT)

Se montó un servidor Ubuntu el cual estará alojados los resultados de las muestras de la temperatura por minuto, es decir se crea una base de datos con la finalidad de que se almacén los valores. Para ello fue sustancial colocar todos los elementos que nos lleven a administrar el servidor, para ellos instalamos LAMP, el cual está compuesto por: Apache, Php, mysql [18].

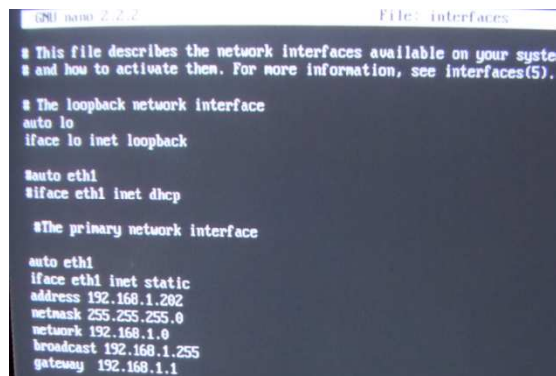
#### 4.11.1 Configuración del servidor

Al servidor, se le asignó una IP estática, la cual fue 192.168.1.202. Para configurar esto se hizo la edición del archivo que contiene los parámetros de la tarjeta de red /etc/network/interfaces, para poder colocar los parámetros que a parecen a continuación se editó dicho archivo [15].

auto eth1----- Aquí hacemos alusión a la tarjeta que se va a configurar.

```
iface eth1 inet static
address "192.168.1.202"
netmask "255.255.255.0"
gateway " 192.168.1.1"
```

En la figura 42 se el archivo modificado y en el cual se encuentran los parámetros de red previamente descritos.



```
GNU nano 2.2.7 File: interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

#auto eth1
#iface eth1 inet dhcp

#The primary network interface
auto eth1
iface eth1 inet static
address 192.168.1.202
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

Figura 42. Descripción de la tarjeta de red.

Se creó una página de bienvenida al servidor misma que se encuentra alojada en el directorio /var/www, para poder ver su respuesta, se deberá colocar en el browser la dirección IP 192.168.1.202 la repuesta a ello se puede observar en la figura 43.

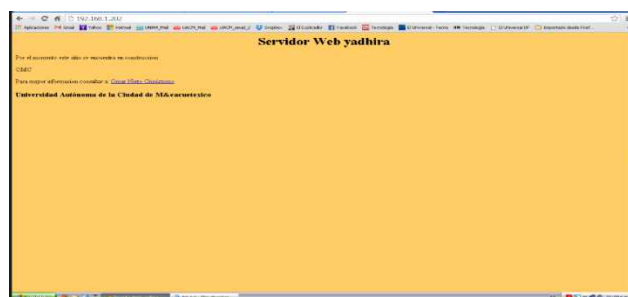


Figura 43. Respuesta del servidor.

Es importante observar los puertos que tenemos abiertos en nuestro servidor, además de saber cuáles son los que ocupan algunas aplicaciones y protocolos para ello se coloca la línea de comando.

➤ nmap 192.168.1.202

Se obtiene el resumen de los puertos abiertos mostrados en la figura 44, en la tabla 16 se observan más claramente.

PUERTO	Estado	Servicio
22/tcp	Abierto	ssh
25/tcp	Abierto	smtp
80/tcp	Abierto	http
110/tcp	Abierto	Pop3
139/tcp	Abierto	Net-BIOS-ssn
143/tcp	Abierto	Imap
445/tcp	abierto	Microsoft-ds
3306/tcp	Abierto	Mysql
5432/tcp	Abierto	Postgresql
8080/tcp	Abierto	http-proxy

Tabla 16. Muestra los puertos abiertos.

```

root@yadhiraserver:~# nmap 192.168.1.202
Starting Nmap 5.00 ( http://nmap.org ) at 2014-07-25 13:27 PDT
Interesting ports on localhost (192.168.1.202):
Not shown: 990 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
5432/tcp  open  postgresql
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
root@yadhiraserver:~#

```

Figura 44. Respuesta del servidor.

#### 4.11.2 Mysql

Para la instalación de MYSQL fue necesario instalar algunos de sus paquetes dentro del servidor como: mysql-server, mysql-client, mysql-common y mysql-admin.

**root@yadhiraserver:** apt-get install mysql-server

Una vez instalado mysql lo que prosigue es conectarse a la base de datos y esto se hizo de la siguiente manera, además de que se creó un usuario con privilegios esto con la finalidad de poder administrar la base de datos y que el acceso sea solamente limitado es decir; Una buena práctica para no evitar problemas de seguridad es, al mismo tiempo, crear un usuario que tendrá un acceso total pero limitado.

Lo primero que se tendrá que hacer será acceder como root:

```
root@yadhiraserver$ mysql -u root -p
```

```
root@yadhiraserver$ mysql -h localhost -u yadhira
```

#### 4.11.3 Base de datos

Una vez instalado y configurado Mysql se creó una base de datos llamada "10 sep", en ella se implementó una tabla llamada datos la cual cuenta con 3 campos que son: sensor, temperatura y fecha.

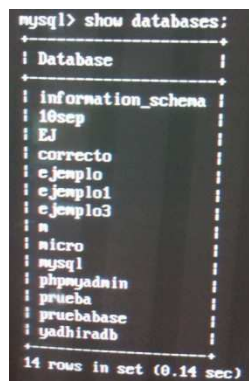
Para poder observar las bases de datos dentro de mysql se ingresara mediante usuario privilegiado de la siguiente manera:

```
root@yadhiraserver> mysql -h localhost -u yadhira
```

Para verificar la base de datos que se creó introducimos la siguiente instrucción.

En la figura 45 se pueden observar la lista de las bases de datos existentes en mysql para poder visualizarlo se colocó la siguiente instrucción.

```
mysql> show databases;
```



```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| 10sep          |
| EJ             |
| correcto       |
| ejemplo        |
| ejemplo1       |
| ejemplo3       |
| m              |
| micro          |
| mysql          |
| phpmyadmin     |
| prueba         |
| pruebabase     |
| yadhiradb      |
+-----+
14 rows in set (0.14 sec)
```

Figura 45. Bases de datos en Mysql.

Para ser uso de la base de datos llamada "10sep" se colocó la siguiente instrucción, esto se hizo con la finalidad de poder utilizarla y crear una tabla. En la figura 46 se observa que la selección fue exitosa y que puede hacer uso ella.

```
mysql> use 10 sep;
```

```
mysql> use 10sep;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Figura 46. Comando use.

Para poder observar la descripción de los campos de la tabla realizada se coloca la siguiente instrucción, esto nos da como respuesta lo que aparece en la figura 47 en la cual se puede observar los siguientes parámetros: field, tipo, Null, key, default y extra, de cada uno de nuestros campos declarados que en este caso fueron: sensor, temperatura y fecha.

```
mysql> describe datos;
```

```
mysql> describe datos;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| sensor | varchar(30) | YES | | NULL | |
| temperatura | double | YES | | NULL | |
| fecha | timestamp | NO | | CURRENT_TIMESTAMP | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figura 47. Comando describe.

Para observar los datos almacenados se colocara la siguiente instrucción esta nos dará como respuesta lo que aparece en la figura 48.

```
mysql> select*from datos;
```

```
TEMP | 22.3 | 2015-02-06 16:21:06 |
TEMP | 22.2 | 2015-02-06 16:21:34 |
TEMP | 22.2 | 2015-02-06 16:22:02 |
TEMP | 22.2 | 2015-02-06 16:22:30 |
TEMP | 22.2 | 2015-02-06 16:22:58 |
TEMP | 22.3 | 2015-02-06 16:23:26 |
TEMP | 22.1 | 2015-02-06 16:23:54 |
TEMP | 22 | 2015-02-06 16:24:22 |
TEMP | 21.7 | 2015-02-06 16:24:50 |
TEMP | 21.8 | 2015-02-06 16:25:18 |
TEMP | 21.7 | 2015-02-06 16:25:46 |
TEMP | 21.9 | 2015-02-06 16:26:14 |
TEMP | 22 | 2015-02-06 16:26:42 |
TEMP | 22 | 2015-02-06 16:27:10 |
TEMP | 22 | 2015-02-06 16:27:38 |
TEMP | 22.1 | 2015-02-06 16:28:06 |
TEMP | 22.1 | 2015-02-06 16:28:34 |
TEMP | 22.1 | 2015-02-06 16:29:02 |
TEMP | 21.9 | 2015-02-06 16:29:30 |
TEMP | 22 | 2015-02-06 16:29:58 |
458 rows in set (0.01 sec)
```

Figura 48. Comando select\*from.

Para solo seleccionar una temperatura muestra se colocó la siguiente instrucción, la cual nos facilita la búsqueda pues se especifica la temperatura a buscar, sin necesidad de estar buscando dentro de cada una de las muestras. En la figura 49 se hace constancia de lo anterior y puede observar que solo nos muestra la temperatura igual a 91° [C].

```
mysql> select * from datos where temperatura="91";
```

```
mysql> select * from datos where temperatura="91";
+-----+-----+-----+
| sensor | temperatura | fecha                |
+-----+-----+-----+
| TEMP   | 91          | 2014-11-14 15:50:11 |
| TEMP   | 91          | 2014-11-19 15:30:20 |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Figura 49. Comando donde se selecciona.

## 4.12 Script en el servidor

Conexión.php.- En este script se realizó la configuración del botón mostrar que aparece en la página WEB, este nos permitirá observar las temperaturas almacenadas en la base de datos de nuestro servidor captadas a través del sensor de temperatura, en la figura 50 se observa la sintaxis que lleva el archivo.

```
$con=conectar();  
if(isset($_REQUEST['mostrar'])){  
    $query="select * from datos";  
    $resultado=mysql_query($query,$con);  
    $total=mysql_num_rows($resultado);  
  
    echo "<table border '2'><tr><td>Sensor</td> <td> Temperatura </td> <td> Fecha</td ></tr>";  
  
    while ($dato=mysql_fetch_array($resultado)){  
        echo "<tr>";  
        echo "<td>" . $dato['sensor'] . "</td>";  
        echo "<td>" . $dato['temperatura'] . "</td>";  
        echo "<td>" . $dato['fecha'] . "</td>";  
        echo "</tr>";  
    }  
    echo "</table>";  
    echo "total de personas registradas:$total";  
}  
  
?>  
</body>  
</html>
```

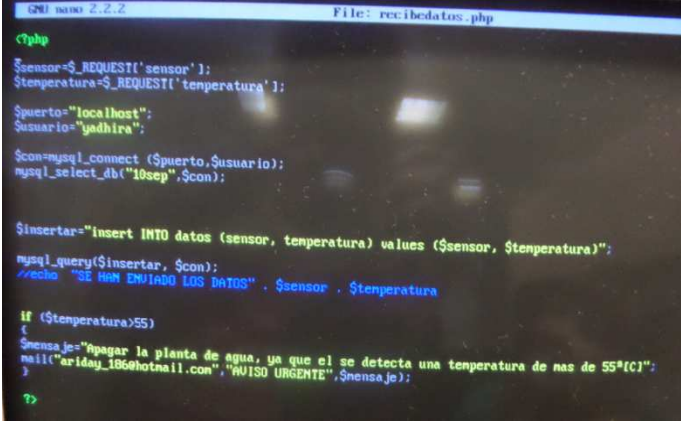
Figura 50. Script para la parte de conexión.

Leerbd.php.- En este script se especifica a que base de datos se conectara en nuestro caso es a la base de datos llamada "10sep" al usuario yadhira, en la figura 51 se muestran las líneas que contiene el dicho archivo.

```
<?php  
function conectar () {  
    $puerto="localhost";  
    $usuario="yadhira";  
  
    $con=mysql_connect ($puerto,$usuario);  
    mysql_select_db("10sep",$con);  
  
    return $con;  
}  
?>
```

Figura 51. Script para la parte de recibe datos.

Recibedatos.php.- En este archivo se hace la declaración de la parte del correo electrónico es decir ahí se colocaron los parámetros que permiten que se envíe un correo electrónico cuando se alcanza una temperatura de 55° [C], en la figura 52 se observa parte de la programación del archivo



```
GNU nano 2.2.2 File: recibedatos.php
<?php
$sensor=$_REQUEST['sensor'];
$temperatura=$_REQUEST['temperatura'];

$puerto="localhost";
$usuario="yadhira";

$con=mysql_connect ($puerto,$usuario);
mysql_select_db("10sep",$con);

$insertar="insert INTO datos (sensor, temperatura) values ($sensor, $temperatura)";
mysql_query($insertar, $con);
//echo "SE HAN ENVIADO LOS DATOS" . $sensor . $temperatura

if ($temperatura>55)
{
    $mensaje="Apagar la planta de agua, ya que se detecta una temperatura de mas de 55°C";
    mail("ariday_18@hotmail.com","AVISO URGENTE",$mensaje);
}
?>
```

Figura 52. Script para la parte de recibe datos.

#### 4.13 Configuración de correo electrónico

Para la implementación del correo electrónico se utilizó la función mail bajo el entorno de php, dicha invocación recibe tres parámetros de manera obligada y otros dos de manera opcional. A continuación se hará una breve descripción:

**Destinatario:** La dirección de correo o direcciones de correo que han de recibir el mensaje. En caso de que se desee colocar más de una dirección electrónica se podrá hacer separándolas por comas.

**Asunto:** Para indicar una cadena de caracteres que queremos que sea el asunto del correo electrónico a enviar. En este caso se deberá colocar el mensaje que aparecerá en el remitente del correo electrónico.

**Cuerpo:** El cuerpo del mensaje, lo que queremos que tenga escrito el correo. En este apartado se colocara el aviso de urgencia, donde se colocara el texto que hace alusión a que la planta está sufriendo un sobrecalentamiento.

\$mensaje="Apagar la planta de agua, ya que el sensor ha excedido una temperatura de 55 grados".

mail ("ariday\_186@hotmail.com","AVISO URGENTE", \$mensaje).

# CAPÍTULO 5

*Diseño de la interfaz del sensor de temperatura.*

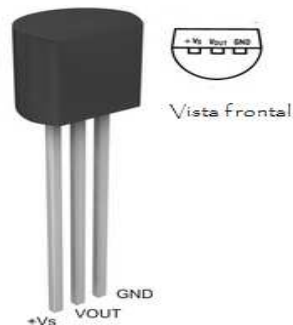
## 5 Diseño de la interfaz del sensor de temperatura

El diseño y elaboración de la interface debe ser precisa, todo ello con la finalidad de que se obtenga un sistema eficiente energéticamente y esto no ayudara a que no se presenten fallos con el tiempo, se adapten y sean de bajo costo.

### 5.1 Características del sensor LM35

El LM35 es un sensor de temperatura, el cual tiene diferentes características y propiedades que de alguna manera nos ayudan a resolver la parte del censado del nivel de temperatura que posee el agua cuando se está filtrado. Este dispositivo tiene como tensión de salida es linealmente proporcional a la Celsius (centígrados) la temperatura. Es decir se comporta de manera lineal, para ello se hicieron diferentes pruebas para constatar este hecho que más adelante se mostraran en una tabla es decir se constatará la teoría con respecto a las pruebas físicas realizadas. El rango de este sensor actúa entre  $-55$  a  $150$  ° C, a continuación se muestra algunas características.

- Linealidad  $10.0$  [mV]/ por [ ° C]
- Exactitud  $0.5$  [° C]
- Rango de temperatura  $-55$  ° a  $150$  [° C]
- Opera entre  $4$  a  $30$  [V]
- Menos de  $60$  [ $\mu$ A] corriente de drenaje
- Baja auto-calentamiento  $0.08$  [° C] en aire
- No linealidad sólo  $\pm 1/4$  [° C] típico
- Salida de baja impedancia  $0.1$  [W] por  $1$  [mA] de carga



#### 5.1.1 Comportamiento lineal

Esto involucra o hace referencia al

comportamiento

Figura 53. Representación del LM35.

de la temperatura con relación al voltaje, es decir que el resultado de estas dos variables implica un conducta lineal, se cuenta con el hecho de que 10 [mV] son equivalentes a 1° [C].

### 5.1.2 Diseño del amplificador no inversor

La señal de salida del sensor alcanza una tensión máxima de 1.5 [V] ya que el nivel de temperatura máxima es de 150 [°C], y el comportamiento es de 10 [mV] por cada 1 [°C], Para aprovechar mejor el rango dinámico del ADC con la finalidad de poder aumentar el nivel máximo de tensión a 5 [V] y generar una mejor resolución la cual se utilizara en el sistema de ADC. En la figura 54 se observa diagrama esquemático de un diseño de un amplificador No-inversor [9].

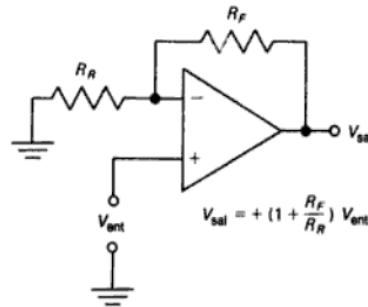


Figura 54. Diagrama esquemático del amplificador NO-inversor.

El diseño del amplificador No-inversor requiere de un arreglo de resistores, el cual requiere de una serie de cálculos que deben ser propiamente determinados a través de la ecuación (3) [9].

$$A = \frac{V_o}{V_i} = 1 + \frac{R_1}{R_2} \dots \dots \dots (3)$$

A continuación se hace el desarrollo de la obtención de los valores de los resistores. Como primer paso se tomaran en cuenta las tensiones de entrada  $V_i$  y salida  $V_o$  con la finalidad de obtener la ganancia.

$$A = \frac{V_o}{V_i} = \frac{5 [V]}{1.5 [V]} = 3.3\bar{3} [V] \dots \dots \dots (4)$$

Lo que prosiguió fue proponer un valor de resistencia de  $R_2$  y así poder obtener el valor de  $R_1$  de la siguiente manera.

$$R_1 = R_2(A - 1) = (1[K\Omega]) * (3.3\bar{3}) = 2.2[K\Omega] \dots \dots \dots (5)$$

### 5.1.3 Comportamiento lineal del sensor en el osciloscopio

En la tabla 17 se puede observar, dos de las mediciones que se realizaron con el osciloscopio

Entrada	Voltaje	Temperatura
1	640m[V]	64° [C]
2	200m[V]	20° [C]

Tabla 17. Mediciones de temperatura para comprobar linealidad.

Se hizo la demostración del comportamiento de sensor de temperatura con el osciloscopio “LG”, modelo: 05-5020G y frecuencia=20[MHz], con la punta de 10x, así poder verificar el comportamiento lineal que se menciona en el apartado anterior. En la tabla 17, se exponen dos muestras en ellas, podemos observar tanto voltaje como temperatura; además de poder verificar la contribución del amplificador no-inversor y esto se puede apreciar en la ecuación (6).

$$\frac{640m[V]}{200m[V]} = 3.2 \dots \dots \dots (6)$$



Figura 55. Respuesta en el osciloscopio.

## 5.2 Realización de las placas de experimentación (Press-N-Pee)

En esta parte de trabajo, se realizó la fuente de voltaje variable DC y el sensor de temperatura LM35 con la etapa del amplificador no-inversor. La impresión de circuitos impresos evita que los dispositivos electrónicos se enreden entre cables y pueda existir corto circuito o que no exista la conexión lo que podría ocasionar que no funcione todo el sistema en conjunto debido al mal cableado.

**Sensor de temperatura.-** Se hizo la construcción del sistema esquemático del sensor de temperatura junto con el amplificador no-inversor, mismo que se puede apreciar en la figura 56 de lado izquierdo y de lado derecho se encuentra el de las pistas.

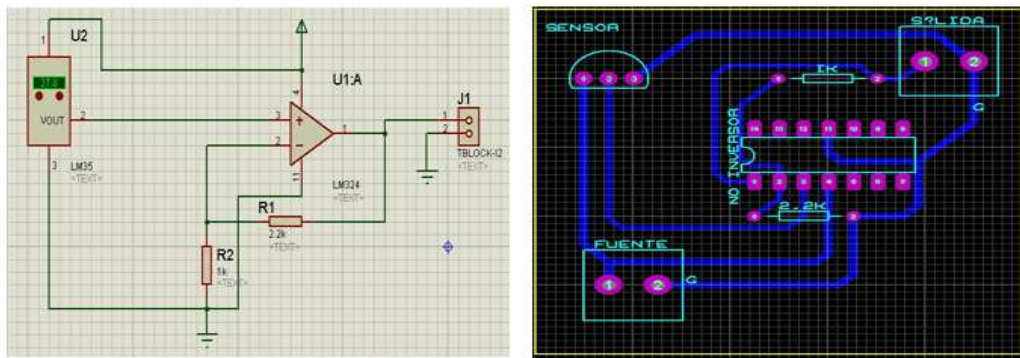


Figura 56. Diagramas con Proteus.

En la figura 57, al final del proceso se obtuvo la placa que contendrá el sensor y la etapa de amplificación tal como aparece en la figura 58.

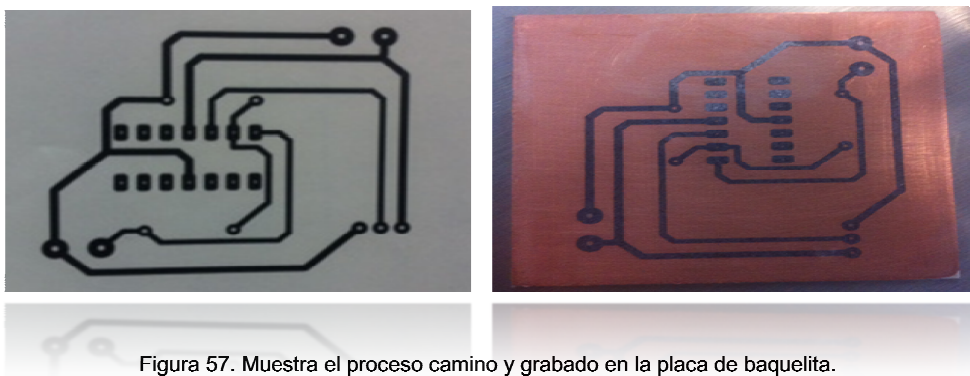


Figura 57. Muestra el proceso camino y grabado en la placa de baquelita.



Figura 58. Circuito final.

### 5.2.1 Creación de la fuente de voltaje DC

Se realizó una fuente de voltaje variable DC en placa, todo ello con la finalidad de poder alimentar el PIC. A continuación se enuncia la lista de elementos que se necesitan para su realización.

#### Lista de elementos para la fuente de alimentación.

- Transformador 30 [V] a 1[A].
- Disipador de calor.
- Puentes de diodos.
- 2 Capacitores electrolíticos 220[ $\mu$ F].
- 1 Capacitor cerámico 100[nF].
- 1 Resistores 220[ $\Omega$ ].
- 2 Borneras.
- Regulador de voltaje (LM317T).
- Potenciómetro.

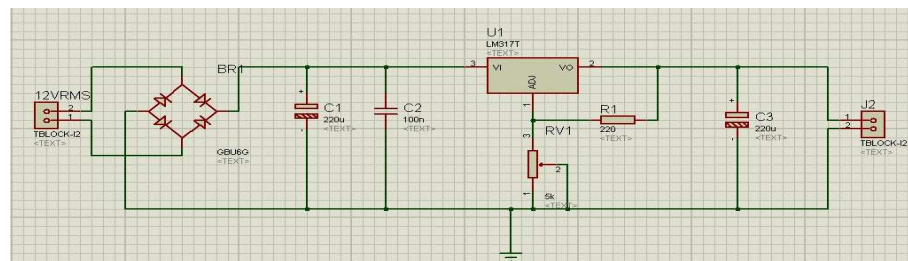


Figura 59. Fuente de voltaje alimentación.

El esquema que aparece en la figura 60 es el que se implementó en la tarjeta, para ello primero se realizó el bosquejo esquemático, así como el diagrama de pistas, y después llevo a la placa.

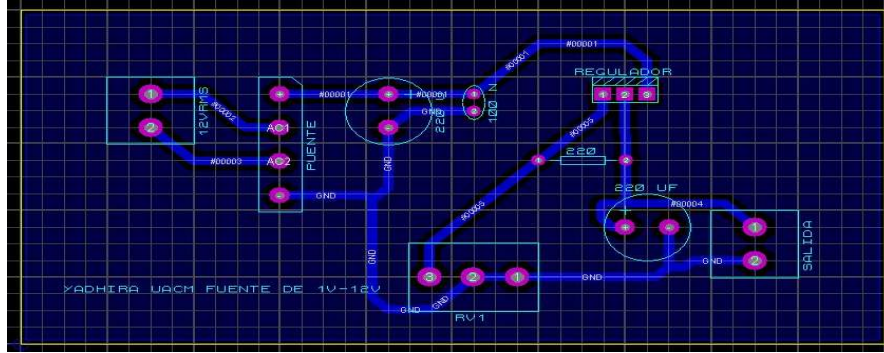


Figura 60. Diagrama de pistas de la fuente

Con dicho esquema se obtiene la placa que se muestra en la figura 61 del lado izquierdo. Al final se logra la placa que se muestra en la figura 61 del lado derecho.

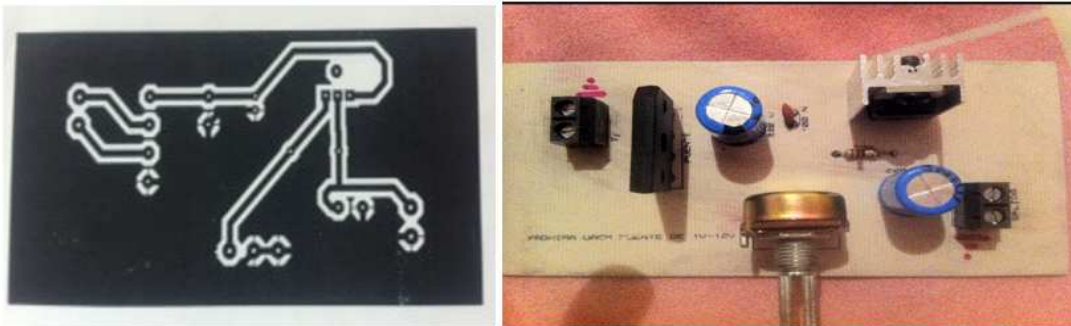


Figura 61. Placa y circuitos finales para la fuente.

Se hizo la medición con el multímetro de los valores mínimo y máximo de la fuente de voltaje, con la finalidad de observar los cambios de voltaje y así calibrar para obtener el de 5 [V].



Figura 62. Respuesta de la fuente de voltaje.

En la tabla 18 se muestran los valores prácticos de la implementación de la fuente de la figura 62, Se hicieron las pruebas para poder observar el comportamiento

real de la fuente, ya que nuestra principal preocupación y lo que cabe resaltar es que se necesitara para alimentar el PIC, este dentro de su tarjeta de expansión de puertos tiene varias entradas que pueden ser utilizadas para otros dispositivos en dado caso que se requiera o sea necesario.

Voltaje Mínimo	Voltaje Máximo
1.2 [V]	18.2 [V]

Tabla 18. Valores máximos y mínimos de la fuente.

# CAPÍTULO 6

*Resultados.*

## 6 Resultados

El sistema se compone de diferentes interfaces, en esta sección se observaran el funcionamiento en conjunto de cada una de las interfaces, cada una de ellas desempeña una función en específico por lo que a continuación se observa su respuesta.

El sistema propuesto deberá ser colocado en la planta purificadora de la UACM-SLT, pero debido a cuestiones de infraestructura de la planta es decir que faltan un nodo de comunicación, así como algunos ajustes en la etapa de purificación, no se pudo llevar a cabo el montaje y para observar que está funcionando sistema de manera adecuada se realizaron pruebas todo ellos con la finalidad que se cumpla con los requerimientos del proyecto.

Como el sensor no se colocó dentro del tinaco la toma de temperaturas se tomaron de la temperatura ambiente, mismas que se almacenaron en la base de datos además de observarse en la pantalla y para observar lo que sucede cuando se tiene una temperatura mayor 55 el sensor se sometió a calor esto con la finalidad de poder comprobar si efectivamente se prende el LED, el buzzer y como se manda el correo electrónico en los siguientes apartados se puede observar que es lo que pasa en cada caso en particular.

Implementar el sistema en el laboratorio y el sensor dentro del dispositivo experimental, son objetivos que quedaron fuera del alcance de este trabajo.

### 6.1 Sistema completo

En la figura 63 se puede observar el sistema completo con cada una de sus interfaces como son: el servidor encargado de recibir las temperaturas y almacenarlos base de datos además de permitir la conexión a la página web, y la computadora en la cual el usuario puede observar la tabla que contiene la toma de mediciones de temperatura del sensor mediante una página WEB.

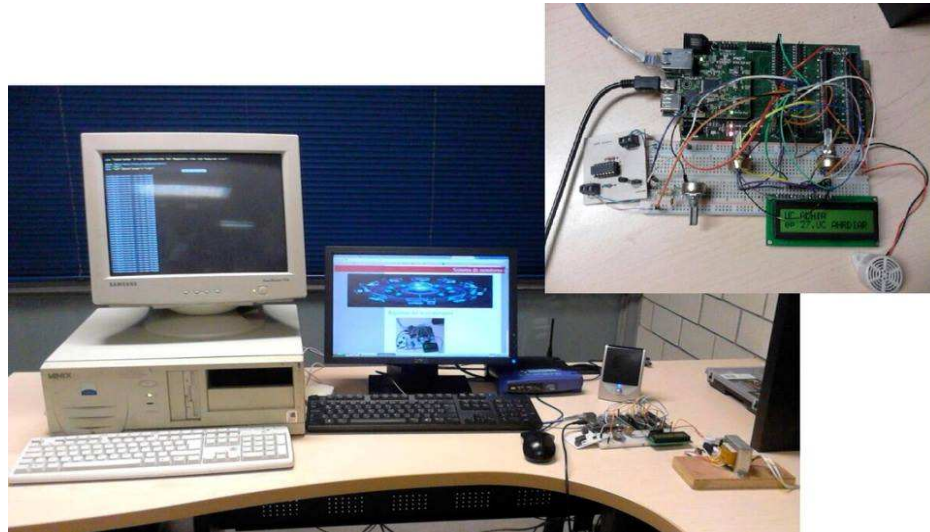


Figura 63. Circuito Final.

## 6.2 Interface con el Microcontrolador

En esta parte se puede observar el microcontrolador conectado a la tarjeta PCB la cual tiene el sensor de temperatura junto con la etapa de amplificación, además de la pantalla, ya que en ella el usuario puede observar el valor con el que actual del sensor.

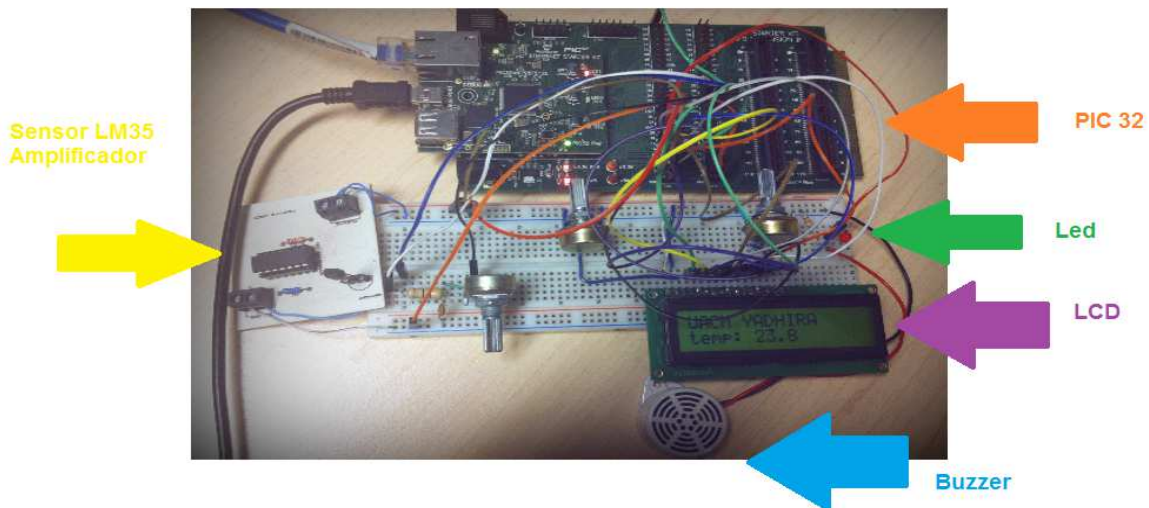


Figura 64. Etapas del microcontrolador.

Cuando el sensor percibe un temperatura de mayor igual a 55° [C], se puede observar que emitirá el sonido se la alarma así como el encendido del LED, para ello se sometió a una prueba y la respuesta de dicho comportamiento se puede observar en la figura 65.

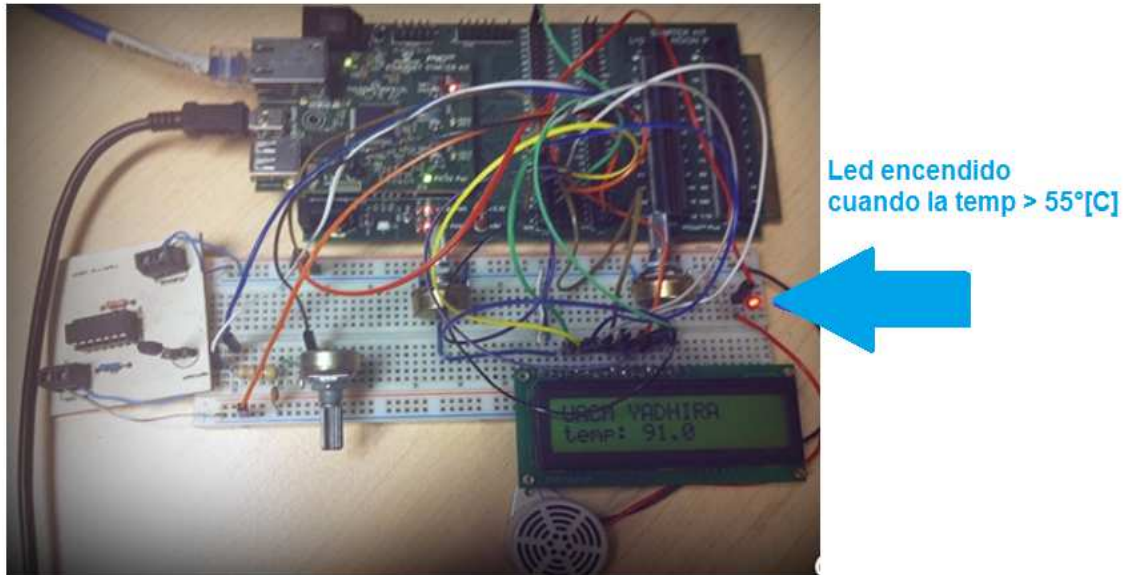


Figura 65. Respuesta del sensor.

### 6.3 Tabla de resultados de las temperaturas

En el servidor se pueden observar el monitoreo del sensor de temperatura, es decir las tomas de temperaturas que se generan por minuto son almacenadas en la base de datos que se creó, todo ello con la finalidad que tanto los encargados de la planta purificadora puedan observar el comportamiento del sensor así como los usuarios que deseen conocer el estado del dicho dispositivo. En la figura 66 se puede observar la tabla que contiene los campos de: sensor, temperatura y hora, dado que la toma de temperatura fue tomada del medio ambiente se puede observar que las mediciones de temperatura se mantienen constantes ya que no hay un cambio drástico ya que el sistema toma y guarda las mediciones cada minuto.

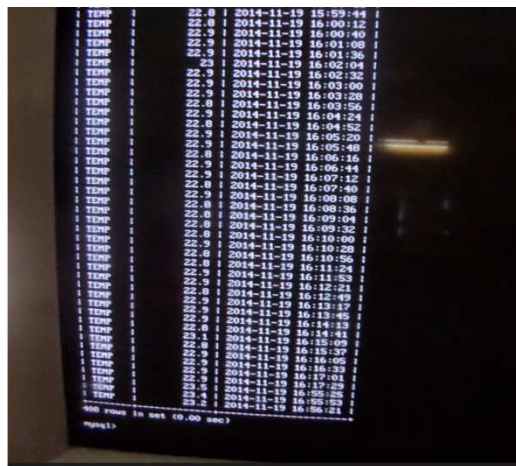


Figura 66. Resultado de la base de datos.

### 6.3.1 Página WEB

La página web fue diseñada de forma que el usuario pudiera interactuar de manera muy sencilla para poder acceder a ella se deberá ingresar en la barra del navegador de su preferencia la IP 192.168.1.202/10sep/conexion.php, se podrá observar una página como la que aparecen la figura 67, en la cual aparece un botón que dice mostrar al darle clic aparecerá la tabla que contiene las mediciones de sensor.

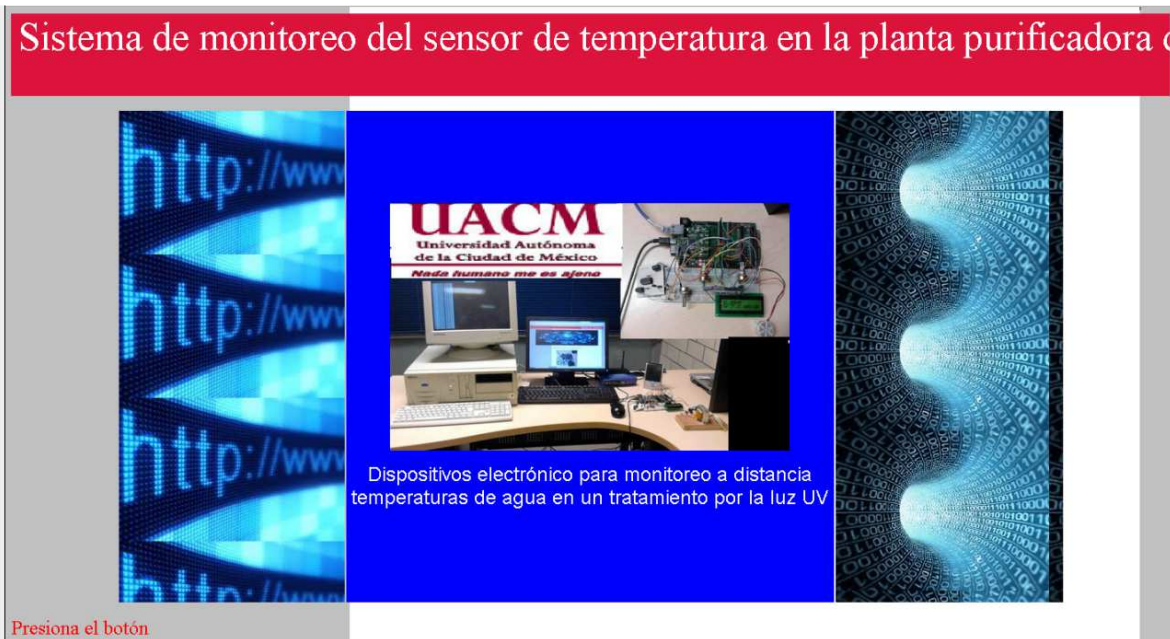


Figura 67. Página Web que observara el usuario.

## 6.4 Correo electrónico

El resultado del correo electrónico se muestra en la figura 68 en ella se puede observar que en la cuenta de hotmail en la bandeja de entrada el correo informando que se está detectando una temperatura mayor de 55° [C], con esto se comprueba que el sistema realizando a acciones que se le programaron.

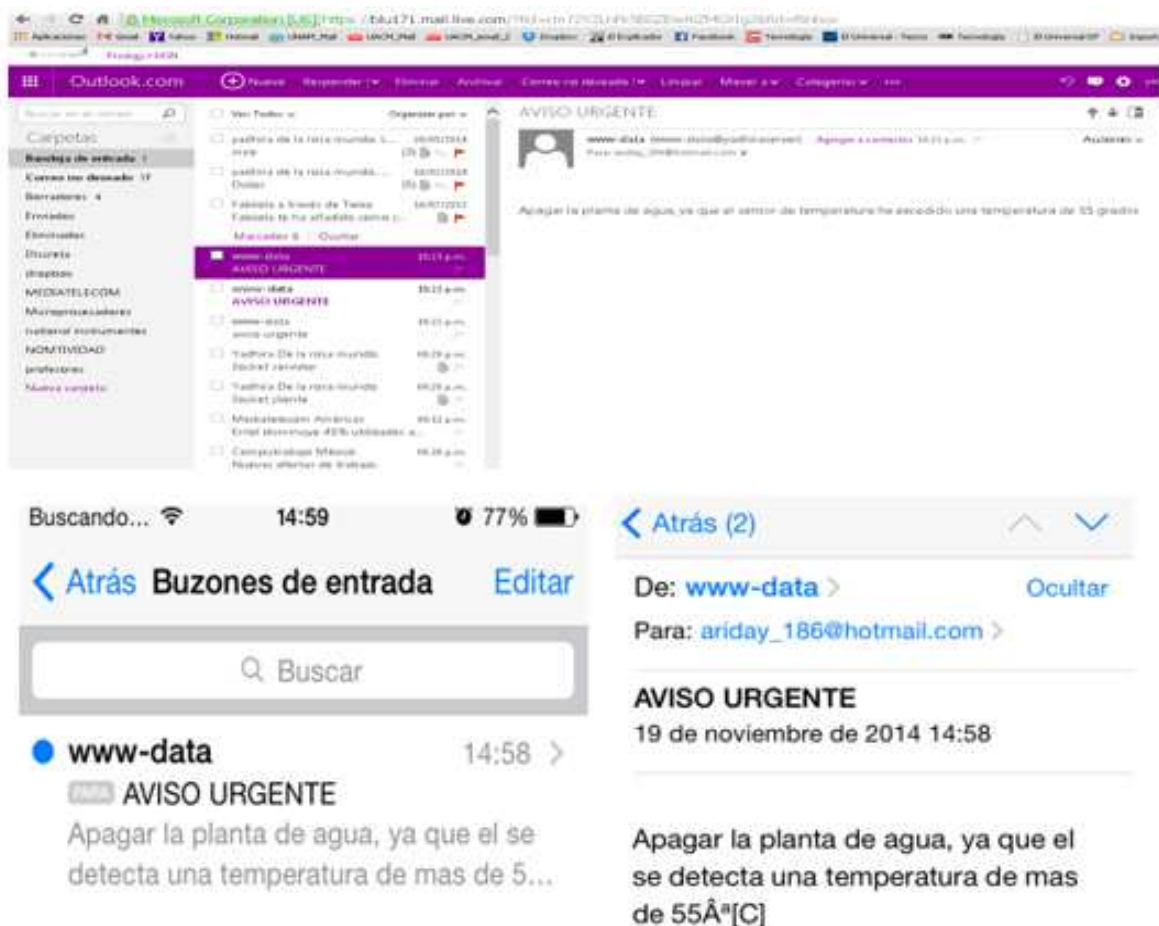


Figura 68. Respuesta en el correo.

## 6.5 Costo del proyecto

El proyecto utilizó diversos dispositivos electrónicos mismos que tiene un costo en el mercado, es trascendental hacer hincapié del costo de proyecto generado, por lo que en la tabla 19 se muestra el costo de cada uno de ellos, esto se realizó con la finalidad de poder observar el costo del sistema. El monto total es de \$2608.79 sin tomar en cuenta los costos de envío de la tarjeta starter y expansión de puertos.

<b>Elemento</b>	<b>costo</b>
<b>Tarjeta Starter kit</b>	\$1126.34
<b>Tarjeta de expansión</b>	\$1081.44
<b>Tarjetas PCB</b>	\$300
<b>LCD</b>	\$30
<b>Cables hembra macho</b>	\$50
<b>Buzzer</b>	\$19
<b>Sensor LM35</b>	\$29
<b>Led</b>	\$3

Tabla 19. Precios de los elementos del sistema.

## Conclusiones

La tendencia por mejorar la calidad de vida del ser humano nos lleva a investigar, diseñar e implementar sistemas, que ayuden a solucionar problemáticas de la vida cotidiana. Tal es el caso del sistema implementado en este trabajo, logrando al final un sistema funcional con capacidad de conexión a internet e interfaces que logran que se cumpla con uno de los objetivos del Laboratorio de Agua de la UACM, el usuario.

El objetivo central de esta tesis, descrito en la sección 1.2, fue alcanzado. El usuario cuenta ahora con una plataforma de comunicación para el monitoreo de parámetros experimentales a distancia. El sistema implementado cumple con las características planteadas a partir de las necesidades de dicho laboratorio y fue entregado operando en el edificio de profesores.

Cabe destacar que los objetivos radican en crear una plataforma digital de comunicación Ethernet vía remota, en el monitoreo del sensor de temperatura LM35 colocado en el fotoreactor de la planta purificadora de agua.

Utilizamos el software y hardware para realizar este proyecto, en el caso del hardware se cuenta con los circuitos electrónicos, así como un microcontrolador de Microchip que proporciona la comunicación Ethernet y diseño de circuito del sensor de temperatura LM35. Para la programación del PIC32 se utilizó el compilador MPLABX en el que se elaboró el programa de comunicación Ethernet, en lenguaje C. De igual manera la etapa del sensor de temperatura, el socket TCP y la LCD se programaron en C.

Cabe constatar que los resultados obtenidos a lo largo del trabajo permitieron probar que los objetivos planteados inicialmente se lograron de manera satisfactoria dando como resultado un producto final, el cual se encargó de realizar el monitoreo del sensor de temperatura cada minuto mostrándolo tanto en la pantalla LCD, como en base de datos creada en el servidor para el público en general además de que existe el ambiente grafico es decir la página WEB con la cual el usuario puede interactuar de manera sencilla y poder observar el

rendimiento del sensor, y como parte sustancial lo del correo electrónico ya que será recibido por el encargado de la planta purificadora de la UACM-SLT se obtendrá el mensaje de alerta esto con la finalidad de que se atienda la emergencia de inmediato y no haya pérdidas o se afecte el sistema. De esta forma, se puede decir que también los objetivos específicos, descritos en la sección 1.2.1, fueron alcanzados plenamente.

El sistema, además, es escalable pues puede tener mayores alcances al adaptar más interfaces compatibles y realizando los cambios del software, todo esto se puede realizar en trabajos posteriores, ya que se cuenta con los protocolos de comunicación usados y los circuitos integrados comunes y de bajo costo, todo dependerá de las necesidades que se requieran.

## Anexos

```
#define __GENERICTCPCLIENT_C
#include "TCPIPConfig.h"
#define USE_DBOUTPUT
#define UN_MINUTO 270000000
#include <plib.h>
#include <stdio.h>
#if defined(STACK_USE_GENERIC_TCP_CLIENT_EXAMPLE)
#include <p32xxxx.h>
#include "Header/HardwareProfile.h"
#include "TCPIP Stack/TCPIP.h"
#include "Explore.h"
#include "LCD.h"
#include "ADC.h"
static BYTE ServerName[] = "192.168.1.202";
//static IP_ADDR IP_dir;
static WORD ServerPort =80;
unsigned char data[1024];
int a;
double z;
char s[20],d[20];
char cad1[50],cad2[50], mensaje[512];
void GenericTCPClient(void)
{
    BYTE i;
    WORD w;
    BYTE vBuffer[20];
    static DWORD Timer;
    static TCP_SOCKET MySocket = INVALID_SOCKET;
    static enum _GenericTCPExampleState
    {
        INICIO=0,
        ESPERA_1MIN,
```

```

LEER_TEMP,
MOSTRAR_LCD,
SM_HOME,
    SM_SOCKET_OBTAINED,
SM_PROCESS_RESPONSE,
    SM_DISCONNECT,
    SM_DONE
} GenericTCPEExampleState = INICIO;
switch(GenericTCPEExampleState)
{
case INICIO:
    TMR4 = 0;
    GenericTCPEExampleState++;
    break;
case ESPERA_1MIN:
    if (TMR4>UN_MINUTO)
    {
        GenericTCPEExampleState++;
    }
    break;
case LEER_TEMP:
    a =readADC(TSENS);
    z=0.08895214*a; //Obtención de temperatura
        //Alarma cuando la temperatura se mayor a 50
        if (z>50){
            PORTC = 0x02;
        }
        else{
            PORTC=0;
        }
        GenericTCPEExampleState++;

    break;

```

```

case MOSTRAR_LCD:
    putsLCD("UACM YADHIRA");
    sprintf(s, "\ntemp: %5.1f", z);
    putsLCD(s);
GenericTCPEXampleState++;
break;

case SM_HOME:// Conecte un socket para el servidor TCP remoto
    MySocket = TCPOpen((DWORD)(PTR_BASE)&ServerName[0],
TCP_OPEN_RAM_HOST, ServerPort, TCP_PURPOSE_GENERIC_TCP_CLIENT);
// Abort operation if no TCP socket of type TCP_PURPOSE_GENERIC_TCP_CLIENT is available
    // If this ever happens, you need to go add one to TCPIPConfig.h
    if(MySocket == INVALID_SOCKET)
        break;

    #if defined(USE_DBOUTPUT)
DBPRINTF("\n-----\n");
DBPRINTF("\nConnecting ... \n");
#endif

    GenericTCPEXampleState++;
    Timer = TickGet();
    break;

case SM_SOCKET_OBTAINED:
    // Wait for the remote server to accept our connection request
    if(!TCPIsConnected(MySocket))
    {
        // Time out if too much time is spent in this state
        if(TickGet()-Timer > 5*TICK_SECOND)
        {
            // Close the socket so it can be used by other modules
            TCPDisconnect(MySocket);
            MySocket = INVALID_SOCKET;
            GenericTCPEXampleState--;
        }
    }

```

```

        }
        break;
    }
    Timer = TickGet();
//strcpypgm2ram((char*)LCDText, "Connecting socket...\n");
    DBPRINTF("Connecting socket...\n");
        // Make certain the socket can be written to
        if(TCPIsPutReady(MySocket) < 125u) {
            // DBPRINTF("Error!!\n");
            break;
        }
        DBPRINTF("Connected!\n");

        /*TCPPutROMString(MySocket, (ROM BYTE*)"POST /ejemplo/recibedatos.php
        HTTP/1.1\n\r");
            TCPPutROMString(MySocket, (ROM BYTE*)"Host: 192.168.1.202:80 \n\r");
            TCPPutROMString(MySocket, (ROM BYTE*)"\n\r");
*/
//sprintf(cad, "TEMP=%f", z);
strcpy (mensaje, "POST /10sep/recibedatos.php HTTP/1.1\n\r");
strcat (mensaje, "Host: 192.168.1.202\n\r" );
    strcat (mensaje, "Connection: keep-alive\n\r" );
    strcat (mensaje, "Accept: text/html, */*\n\r");
    strcat (mensaje, "Accept-Language: en-us\n\r" );
    strcat (mensaje, "Content-Type: application/x-www-form-urlencoded\n\r" );
    sprintf (cad1,"sensor=\"TEMP\"&temperatura=%04.1f\n\r",z);
    sprintf (cad2, "Content-Length: %d\n\r", strlen(cad1) );
    strcat (mensaje, cad2 );
    strcat (mensaje," \n\r");
    strcat (mensaje, cad1 );

    TCPPutROMString(MySocket, (ROM BYTE*) mensaje);
        // Send the packet
        TCPFlush(MySocket);

```

```

        GenericTCPExampleState++;
        break;

case SM_PROCESS_RESPONSE:
    if(!TCPIsConnected(MySocket))
    {
        GenericTCPExampleState = SM_DISCONNECT;
    }

    w = TCPIsGetReady(MySocket);
    i = sizeof(vBuffer)-1;
    vBuffer[i] = '\0';
    while(w)
    {
        if(w < i)
        {
            i = w;
            vBuffer[i] = '\0';
        }
        w -= TCPGetArray(MySocket, vBuffer, i);

        DBPRINTF(vBuffer);

        TCPPutROMString(MySocket, (ROM BYTE*)"data from the TCP Server running on the
PIC32!\n");
        TCPFlush(MySocket);

        if(GenericTCPExampleState == SM_PROCESS_RESPONSE)
            break;
    }
    //DBPRINTF("\n");
    break;
case SM_DISCONNECT:
    DBPRINTF("Closing socket...\n");
    TCPDisconnect(MySocket);
    MySocket = INVALID_SOCKET;
    GenericTCPExampleState = SM_DONE;

```

```
        //DBPRINTF("Closed.\n");
        break;

    case SM_DONE:
        // Do nothing unless the user pushes BUTTON1 and wants to restart the
        whole connection/download process
        //if(BUTTON1_IO == 0u)
            GenericTCPEXampleState = INICIO;
        break;
    }
}

#endif //if defined(STACK_USE_GENERIC_TCP_CLIENT_EXAMPLE)
```

## Glosario

**Periféricos.** Dispositivos externos que intercambian datos con el procesador. La comunicación entre el procesador y el periférico está regulada por el procesador de acuerdo con los métodos.

**Interrupciones.** El periférico que está listo para ser atendido por el procesador solicita una “interrupción” de la ejecución del programa para que el procesador lo atienda.

**RJ-45.** Un conector similar a un conector telefónico que admite hasta ocho cables, utilizado para conectar dispositivos Ethernet.

**PCB.** Un circuito impreso (Printed Circuit Board) es una tarjeta o placa utilizada para realizar el montaje de los distintos elementos electrónicos que conforman el circuito y las interconexiones eléctricas entre ellos.

**Stack.** La pila de protocolos, es una colección ordenada de protocolos organizados en capas que se ponen unas encima de otras y en donde cada protocolo implementa una abstracción encuadrada en la abstracción que proporciona la capa sobre la que está encuadrada. Los protocolos encuadrados en la capa inferior proporcionan sus servicios a los protocolos de la capa superior para que estos puedan realizar su propia funcionalidad.

**ADC.** Convertidor analógico-digital, este se encarga de llevar a cabo la conversión de una señal analógica a un equivalente binario.

**RS-232.** Es un protocolo o estándar mundial que rige los parámetros de uno de los nodos de comunicación serial. Por medio de este protocolo se estandarizan la velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, la distancias entre equipos, los conectores, etc.

Además de las líneas de transmisión ( $T_x$ ) y recepción ( $R_x$ ), las comunicaciones seriales poseen otras líneas de control de flujo (Handshake), donde su uso es opcional dependiendo del dispositivo a conectar.

## Bibliografía

- [1] M. M. Bloomfield, Química de los organismos vivos, Mexico: Limusa, 2000.
- [2] J. I. C. Velázquez, REDES DE DATOS CONTEXTO Y EVOLUCION, México: Samara, 2014.
- [3] D. E. Comer, REDES GLOOBALES DE INFORMACION CON INTERNET Y TCP/IP PRINCIPIOS BASICOS PROTOCOLOS Y ARQUITECTURA, México: Pearson Prentice Hall, 1996.
- [4] Microchip, «TCP/IP Stack for PIC18, PIC24, dsPIC & PIC32,» 1998-2012. [En línea]. Available: [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2680&dDocName=en537041](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537041). [Último acceso: 1 Octubre 2014].
- [5] H. P. Enrique, Introducción a las telecomunicaciones modernas, México: Limusa, 2006.
- [6] ANYELGUTI, «Curso de PHP Básico,» [En línea]. Available: [http://phpbasico.freevar.com/temas/php14\\_1.php](http://phpbasico.freevar.com/temas/php14_1.php). [Último acceso: 9 Octubre 2014].
- [7] Intef, «Instalacion y configuración de PHPMyAdmin,» [En línea]. Available: [http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/instalacin\\_y\\_configuracin\\_de\\_phpmyadmin.html](http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/instalacin_y_configuracin_de_phpmyadmin.html). [Último acceso: 10 Octubre 2014].
- [8] L. E. Frenzel, Sistemas electrónicos de comunicación, Mexico: Alfaomega, 2003.
- [9] L. N. Robert L.Boylestad, Electrónica: Teoría de circuitos y dispositivos electronicos, 10a ed., México: PerasonPerason, 2009.
- [10] R. P. Areny, Sensores y acondicionadores de señal, 4a ed., México: Alfaomega Macombo, 2007.
- [11] R. Blake, Sistemas electrónicos de comunicaciones, Mexico: Thomson, 2004.
- [12] Microhip, «TCP/IP Stack for PIC18, PIC24, dsPIC & PIC32,» 2011. [En línea]. Available: [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2680&dDocName=en537041](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537041).
- [13] «Demo App,» Michochip, [En línea]. Available: <http://ww1.microchip.com/downloads/en/softwarelibrary/microchip-application-libraries-v2012-07-18-windows-installer.exe>.
- [14] «Probando el Stack TCP/IP de Microchip con MPLABX,» AquiHayApuntes.com, 2007. [En línea]. Available: <http://www.aquihayapuntes.com/curso-pic32/probando-el-stack-tcpip-de-microchip-con-mplabx.html>. [Último acceso: 7 Enero 2015].
- [15] F. Informatica, «Utiliza LINUX,» 2015. [En línea]. Available: <http://www.utilizalinux.com/2013/02/configurar-un-servidor-como-router-con.html>. [Último acceso: 2014 Noviembre 7 ].

- [16] «PIC32MX for DV Node Adapter,» Japonesa, [En línea]. Available: <http://sourceforge.jp/projects/pic32nodeadap/scm/git/nodeadapter/commits/6baf72d5b21f19b900e3e0a81ae2f349f167e875>. [Último acceso: 2014 Noviembre 7].
- [17] Microchip, «Microchip TCP/IP Stack Help,» 2011. [En línea]. Available: <http://www.egr.msu.edu/classes/ece480/capstone/fall11/group03/TCPIP%20Stack%20Help.pdf>. [Último acceso: 13 Noviembre 2014].
- [18] Ubuntu. [En línea]. Available: <https://help.ubuntu.com/10.04/index.html>. [Último acceso: 8 Septiembre 2014].
- [19] «El protocolo DHCP y su funcionamiento,» 2004. [En línea]. Available: <http://www.ordenadores-y-portatiles.com/dhcp.html>. [Último acceso: 10 Enero 2015].
- [20] Microchip, «PIC32MX795F512L In Production,» Microchip, 1998-2014. [En línea]. Available: <http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC32MX795F512L>. [Último acceso: 18 Diciembre 2014].