

**UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

**“PROTOTIPO DE SISTEMA DE MONITOREO PARA LA PLANTA DE  
TRATAMIENTO DE AGUAS RESIDUALES DE LA UNIVERSIDAD  
TECNOLÓGICA DE LA MIXTECA”**

**TESIS**

**PARA OBTENER EL TÍTULO DE  
INGENIERO EN ELECTRÓNICA**

**PRESENTA**

**JOSÉ YOVANY LUIS GARCÍA**

**DIRECTOR DE TESIS**

**M.C. JOSÉ ANTONIO MORENO ESPINOSA**

**HUAJUAPAN DE LEÓN, OAXACA, OCTUBRE DEL 2013**



Tesis presentada el 31 de Octubre de 2013  
ante los siguientes sinodales:

Ing. Heriberto I. Hernández Martínez  
Ing. Alejandro E. Ramírez González  
Dr. Vania S. Robles González

Director de tesis:  
M.C. José Antonio Moreno Espinosa





# Dedicatoria

*A mi mayor fuente de sabiduría e inspiración  
hoy y siempre, mis padres:*

*José Domingo Luis Rodríguez  
y  
María Concepción García Gazga*

*De todo corazón*

*Yovany*



# Agradecimientos

*¿Quién soy yo?,*

*soy parte de todas las personas que han interactuado conmigo a lo largo de este escenario llamado “vida”, soy las manos de mi abuela, el amor de mi abuelo, las lágrimas y sufrimientos de mi madre, la fuerza y el coraje de mi padre, el amor de mi querida novia, las bromas de mis amigos, primos, compañeros de clase y mi hermano, el conocimiento que los profesores han compartido conmigo a lo largo de mi preparación profesional, soy todos los consejos que todas las personas me han dado. Por lo tanto, no soy solo yo, sino, la suma de todo lo anterior. Soy el orgulloso resultado del trabajo de otros, de todos aquellos que han tocado mi vida de diversas maneras, por lo cual me siento afortunado, y ahora, es tiempo de dar y compartir lo que soy.*

A mis padres por la enorme paciencia, el apoyo moral, las palabras de aliento, por comprender las distintas etapas de mi vida y por el amor incondicional.

A mi madre María Concepción, por todo el amor, sabiduría, cariño y por ser un ejemplo a seguir en la lucha por la vida, hacer el bien y lo correcto.

A mi padre José Domingo, por ser un padre ejemplar, fuerte, sabio, con el don de ser líder y dar buenos consejos y disciplina cuando son necesarios.

A mi hermano Ahuizotl, para que este proyecto de investigación te sirva como ejemplo a seguir y despierte tu interés por el deseo de superarte, aprender y lograr tus objetivos, porque con tus ocurrencias la vida parece simple y porque has venido a ocupar mi lugar como “el de los mandados”.

A mi prometida Gabriela, por traer a mi vida estabilidad sentimental y emocional, por ser el complemento ideal y regalarme la dicha de formar parte de un nuevo comienzo, por la paciencia y el tiempo que no pasé contigo por terminar mi proyecto. *“Vi la magia en tus ojos, es caricia en mi piel, es locura el deso en tu boca de miel...”*

A mi director de tesis M.C. José Antonio Moreno Espinosa por haber confiado en mis capacidades y por haberme guiado a lo largo del desarrollo de este proyecto de tesis.

A los profesores asignados como sinodales, Ing. Alejandro E. Ramírez González, Ing. Heriberto I. Hernández Martínez y Dr. Vania S. Robles González. Que gracias a sus observaciones y revisiones han aportado en el mejoramiento de este trabajo.

A Heriberto, que de manera desinteresada apoyó en la estructuración y redacción de este documento.

A todos mis compañeros de licenciatura, que pasamos por muchas aventuras y festejos, en donde las bromas no hacían falta y hacían que la carga de trabajo no fuera tan pesada. En especial a mis compañeros de trabajo Eric (Saavedra), Miguel Ángel (Pequeña), Gildardo ('Z'), Edgar Espindola (Tiburón), Cornelio, Juanito (Chino), Lucio, Indira, Itzel (Bambucha), Walkiria, Omar, David Cruz (Oso), Edilberto (Perico), David Antonio (Pato), Emmanuel (Fito) y Diego.

A mis amigos del grupo de salsa, Tania, Norma, Adriana, Carlos y Jorge, por esas salidas de baile, por ofrecer su amistad sincera y porque fue en este grupo donde conocí a Gabriela.

A mis compañeros de licenciatura Eric, Miguel Ángel y Carlos que por su sinceridad nos hicimos grandes amigos.

A los amigos del DoTA que siempre pasamos momentos de diversión, tensión y decepción en las eternas partidas. A Herme, K, Negro, Pita, Cueto, Orlando, Fercho, Chiquilin, Eric, Oso, Ante, Oliver, Mars y los que no alcanzo a recordar.

A Julio Alfredo Cortés Rodríguez, por proporcionar la plantilla de LaTeX en la que se redactó este documento.

A los desarrolladores de las herramientas de *software* libre que se utilizaron en este proyecto de tesis, ya que ofrecen inspiración y herramientas para la innovación tecnológica, y porque ofrecen sus mejores aplicaciones al público en general de manera desinteresada. A los desarrolladores de TeXLive, TeXMaker, JabRef, Yii, Bootstrap, YiiBootstrap, YiiBooster, MySQL, Apache, Xampp, GitHub, Java, NetBeans, Eclipse, Gimp, Inkscape y Notepad++.

# Resumen

Actualmente, el monitoreo remoto es ampliamente utilizado en la industria para monitorizar y controlar procesos. El monitoreo y automatización remota se engloban en el concepto de SCADA. Los sistemas SCADA constan de elementos *hardware* y *software*, que permiten el acceso a datos remotos y el control de un proceso (industrial) mediante el uso de sistemas de comunicaciones.

Este proyecto de tesis tiene como objetivo diseñar y construir un prototipo de sistema de monitoreo remoto de temperatura, pH y Oxígeno Disuelto para la planta de tratamiento de aguas residuales de la Universidad Tecnológica de la Mixteca.

El sistema propuesto consta de sensores de propósito industrial, almacena información de mediciones en un servidor de bases de datos y ofrece acceso vía Web mediante una interfaz de usuario que permite visualizar el estado de los parámetros medidos de manera gráfica y tabular.

Debido a la magnitud del sistema SCADA, para su desarrollo se emplea la metodología de desarrollo para mejoramiento de procesos de producción y la metodología de sistemas empotrados. Con ello se establecen las fases de desarrollo que producen un proyecto completo y modular.

Al final, se obtuvo un prototipo de sistema SCADA capaz de monitorizar una planta o proceso industrial usando diferentes interfaces de comunicación e interfaces de usuario para la visualización de los registros de mediciones.



## Acrónimos

ADC	Convertidor Análogo-Digital ( <i>Analog-Digital Converter</i> ).
ADU	Unidad de Datos de Aplicación ( <i>Application Data Unit</i> ).
AR	<i>Active Record</i> .
ARP	<i>Address Resolution Protocol</i> .
CAS	<i>Circuito de Acondicionamiento de Señal</i> .
CPU	Unidad Central de Procesamiento ( <i>Central Processing Unit</i> ).
CI	Circuito Integrado.
CMMI	Modelo de Integración de Capacidad y Madurez ( <i>Capability Maturity Model Integration</i> ).
CRC	<i>Cyclical Redundancy Check</i> .
CSMA	<i>Carrier Sense Multiple Access</i> .
DAC	Convertidor Digital-Análogo ( <i>Digital-Analog Converter</i> ).
DBMS	Sistema de Gestión de Bases de Datos ( <i>Data Base Management Systems</i> ).
DCE	Equipo de Comunicaciones de Datos ( <i>Data Communication Equipment</i> ).
DLL	Librería de Enlace Dinámico ( <i>Dynamic Link Library</i> ).
DRY	<i>Don't Repeat Yourself</i> .
DTE	Equipo Terminal de Datos ( <i>Data Terminal Equipment</i> ).
EIA	Asociación de Industrias Electrónicas ( <i>Electronic Industrial Association</i> ).
E-R	Entidad-Relación ( <i>Entity Relationship</i> ).
FieldBus	Bus de Campo.
FTP	Protocolo de Transferencia de Ficheros ( <i>File Transfer Protocol</i> ).
GUI	Interfaz Gráfica de Usuario ( <i>Graphic User Interface</i> ).
HDLC	<i>High-level Data Link Control</i> .
HMI	Interfaz Humano Máquina ( <i>Human Machine Interface</i> ).
HTTP	Protocolo de Transferencia de Hipertexto ( <i>Hypertext Transfer Protocol</i> ).

---

IEC	Comisión Electrotécnica Internacional ( <i>International Electrotechnical Commision</i> ).
ICMP	<i>Internet Control Message Protocol</i> .
IP	Protocolo de Internet ( <i>Internet Protocol</i> ).
ISA	Sociedad de Instrumentación de América ( <i>Instrument Society of America</i> ).
ISO	Organización Internacional de Normalización ( <i>International Organization for Standarization</i> ).
JDBC	<i>Java Database Connectivity</i> .
LAN	Redes de Área Local ( <i>Local Area Network</i> ).
LRC	<i>Longitudinal Redundancy Check</i> .
MCU	Microcontrolador ( <i>Microcontroller Unit</i> ).
MTU	Unidad Terminal Maestra ( <i>Master Terminal Unit</i> ).
MVC	Modelo Vista Controlador ( <i>Model View Controller</i> ).
OD	Oxígeno Disuelto.
ODBC	<i>Open Data Base Conectivity</i> .
op-amp	Amplificador Operacional.
OSI	Interconexión de Sistemas Abiertos ( <i>Open System Interconnection</i> ).
O <sub>2</sub>	Oxígeno.
PCB	Tarjeta de Circuito Impreso ( <i>Printed Circuit Board</i> ).
PDU	Unidad de Datos de Protocolo ( <i>Protocol Data Unit</i> ).
PID	Proporcional Integral Diferencial.
ppm	Partes por Millón ( <i>Parts per Million</i> ).
PPP	<i>Point-to-Point Protocol</i> .
PTAR	Planta de Tratamiento de Aguas Residuales.
RTU	Unidad Terminal Remota ( <i>Remote Terminal Unit</i> ).
SCADA	Control con Supervisión y Adquisición de Datos ( <i>Supervisory Control and Data Acquisition</i> ).
SMTP	Protocolo de Transferencia de Correo Simple ( <i>Simple Mail Transfer Protocol</i> ).
SPICE	<i>Simulation Program with Integrated Circuits Emphasis</i> .
SQL	<i>Structured Query Language</i> .
STLA	Sistema de Tratamiento por Lodos Activados.
TCP	Protocolo de Control de Transmisión ( <i>Transmission Control Protocol</i> ).



---

TIA	Asociación de la Industria de Telecomunicaciones ( <i>Telecommunications Industry Association</i> ).
UDP	<i>User Datagram Protocol</i> .
UML	Lenguaje Unificado de Modelado ( <i>Unified Modeling Language</i> ).
USART	<i>Universal Synchronous and Asynchronous serie Receiver and Transmitter</i> .
UTM	Universidad Tecnológica de la Mixteca.
UTP	Cable de Par Trenzado ( <i>Unshielded Twisted Pair</i> ).
μP	Microprocesador ( <i>Microprocessor</i> ).
WAN	Redes de Área Amplia ( <i>Wide Area Network</i> ).
WWW	<i>World Wide Web</i> .
Yii	<i>Framework Yii (Yes, it is)</i> .



# Índice general

<b>Dedicatoria</b>	V
<b>Agradecimientos</b>	VII
<b>Resumen</b>	IX
<b>Acrónimos</b>	XI
<b>Índice general</b>	XVIII
<b>Índice de tablas</b>	XIX
<b>Índice de figuras</b>	XXI
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del Problema . . . . .	3
1.2. Justificación . . . . .	4
1.3. Hipótesis . . . . .	4
1.4. Objetivos . . . . .	4
1.4.1. Objetivo general . . . . .	4
1.4.2. Objetivos específicos . . . . .	5
1.5. Contenido del Documento de Tesis . . . . .	5
<b>2. Marco Teórico</b>	<b>7</b>
2.1. Sistemas SCADA . . . . .	7
2.1.1. Componentes <i>hardware</i> de un sistema SCADA distribuido . . . . .	9
2.1.1.1. Instrumentación . . . . .	10
2.1.1.2. Componentes de la RTU . . . . .	11
2.1.1.3. Componentes de la MTU . . . . .	13
2.1.2. Componentes <i>software</i> de un sistema SCADA distribuido . . . . .	14
2.1.2.1. Comunicación con dispositivos de campo . . . . .	15
2.1.2.2. Comunicación entre aplicaciones . . . . .	15
2.1.2.2.1. ODBC . . . . .	15
2.1.2.2.2. SQL . . . . .	16
2.1.2.3. Bases de datos relacionales . . . . .	16
2.1.2.4. Interfaces de usuario (UI) . . . . .	16
2.1.3. SCADA en la Internet . . . . .	17

2.2.	Planta de Tratamiento de Aguas Residuales de la UTM . . . . .	17
2.2.1.	El Agua Residual . . . . .	17
2.2.2.	Sistema de tratamiento de aguas residuales de la UTM . . . . .	18
2.2.2.1.	Sistema de tratamiento por lodos activados . . . . .	19
2.3.	Descripción de Sensores . . . . .	20
2.3.1.	Electrodo de pH . . . . .	20
2.3.1.1.	Principio de funcionamiento del electrodo de pH . . . . .	20
2.3.1.2.	Características del electrodo de pH . . . . .	21
2.3.2.	Electrodo de OD . . . . .	22
2.3.2.1.	Principio de funcionamiento del electrodo de OD . . . . .	23
2.3.2.2.	Características del electrodo de OD . . . . .	23
2.3.3.	Sensor de temperatura LM35 . . . . .	24
2.3.3.1.	Funcionamiento y características del sensor LM35 . . . . .	24
2.4.	Protocolos de Comunicaciones . . . . .	24
2.4.1.	El modelo de referencia OSI . . . . .	25
2.4.1.1.	Capa física . . . . .	25
2.4.1.2.	Capa de enlace de datos . . . . .	26
2.4.1.3.	Capa de red . . . . .	26
2.4.1.4.	Capa de transporte . . . . .	27
2.4.1.5.	Capa de sesión . . . . .	27
2.4.1.6.	Capa de presentación . . . . .	27
2.4.1.7.	Capa de aplicación . . . . .	27
2.4.2.	Interfaces serie . . . . .	27
2.4.2.1.	Comunicación de datos serie en el estándar RS-232 . . . . .	28
2.4.2.1.1.	Características de las señales eléctricas . . . . .	28
2.4.2.1.2.	Descripción funcional de los circuitos de intercambio . . . . .	29
2.4.2.2.	Comunicación de datos serie en el estándar RS-485 . . . . .	29
2.4.3.	Modbus . . . . .	30
2.4.3.1.	Capa de enlace de datos del protocolo Modbus . . . . .	31
2.4.3.2.	Descripción de los códigos de función del protocolo Modbus . . . . .	33
2.4.3.2.1.	0x01 Lectura de Salidas Digitales o Relevadores . . . . .	33
2.4.3.2.2.	0x02 Lectura de Entradas Digitales . . . . .	34
2.4.3.2.3.	0x04 Lectura de Registros de Entrada . . . . .	34
2.4.3.2.4.	0x05 Escritura de una Salida Digital o Relevador Individual . . . . .	35
2.4.3.2.5.	0x06 Escritura de un Registro Individual . . . . .	35
2.4.4.	Suite de protocolos TCP/IP . . . . .	36
2.4.4.1.	Capa de enlace . . . . .	36
2.4.4.2.	Capa de internet . . . . .	37
2.4.4.3.	Capa de transporte . . . . .	37
2.4.4.4.	Capa de aplicación . . . . .	37
2.5.	Herramientas de Desarrollo de Aplicaciones Web y de Escritorio . . . . .	37
2.5.1.	Java . . . . .	38
2.5.2.	PHP . . . . .	38
2.5.3.	MySQL . . . . .	39
2.5.4.	Framework Yii . . . . .	39

2.5.5. JavaScript	41
<b>3. Desarrollo del Sistema</b>	<b>43</b>
3.1. Iniciativa	44
3.1.1. Estado actual de la PTAR	44
3.1.2. Propuesta	44
3.2. Gestionamiento	45
3.2.1. Requerimientos funcionales de la propuesta	45
3.2.1.1. Clasificación y análisis de requerimientos	45
3.2.2. División <i>hardware</i> y <i>software</i>	47
3.2.2.1. Selección de hardware	47
3.2.2.2. Selección de herramientas <i>software</i> de diseño y desarrollo	49
3.3. Definición	50
3.3.1. Diseño hardware	50
3.3.1.1. Diseño de la etapa de acondicionamiento de los sensores	50
3.3.1.1.1. CAS del sensor de pH	50
3.3.1.1.2. CAS del sensor de OD	54
3.3.1.1.3. CAS del sensor de temperatura	57
3.3.1.2. Entradas y salidas de la RTU	59
3.3.1.3. Interfaz RS-485	59
3.3.1.4. Transceptor RS-232/RS-485	61
3.3.2. Diseño <i>software</i>	62
3.3.2.1. <i>software</i> de la RTU	62
3.3.2.1.1. Requerimientos formales	62
3.3.2.1.2. Clases	62
3.3.2.1.3. Diagramas de actividades	63
3.3.2.2. <i>software</i> de la MTU	66
3.3.2.2.1. Requerimientos formales	66
3.3.2.2.2. Clases	67
3.3.2.2.3. Casos de uso	68
3.3.2.2.4. Diagramas de actividades	70
3.3.2.3. <i>software</i> de la aplicación Web	72
3.3.2.3.1. Requerimientos formales	72
3.3.2.3.2. Clases	74
3.3.2.3.3. Modelo entidad-relación	75
3.3.2.3.4. Casos de uso	75
3.3.2.3.5. Diagramas de actividades	77
3.3.3. Integración de elementos <i>hardware</i> y <i>software</i>	78
3.3.3.1. Diseño del PCB de los módulos <i>hardware</i>	80
3.3.3.1.1. Diseño del PCB para el CAS del sensor de pH	80
3.3.3.1.2. Diseño PCB para el CAS del sensor de OD y temperatura	82
3.3.3.1.3. Diseño del PCB para el módulo del MCU	84
3.3.3.1.4. Diseño del PCB para el transceptor RS-232/RS-485	86
3.3.3.2. Diseño de interfaces gráficas	87

<b>4. Verificación y Validación del Producto Final: Pruebas y Resultados</b>	<b>89</b>
4.1. Pruebas de Caja Blanca . . . . .	90
4.1.1. Prueba de configuración inicial del sistema . . . . .	90
4.1.2. Prueba de generación de $V_{offset}$ y $V_{POD}$ . . . . .	92
4.1.3. Prueba de variaciones de voltaje en las entradas . . . . .	93
4.2. Pruebas de Caja Gris . . . . .	96
4.2.1. Prueba de comunicación serie entre la MTU y la RTU . . . . .	97
4.2.2. Prueba de comunicación entre la MTU y la base de datos . . . . .	102
4.2.3. Prueba de conexión de sensores a la RTU . . . . .	102
4.2.4. Prueba del <i>software</i> de la MTU . . . . .	104
4.2.5. Prueba de la aplicación Web . . . . .	110
4.3. Pruebas de Caja Negra . . . . .	114
4.3.1. Prueba de gestión de sesión de usuario . . . . .	114
4.3.2. Prueba de gestión de estados digitales . . . . .	116
<b>5. Conclusiones</b>	<b>119</b>
5.1. Líneas Futuras de Investigación . . . . .	121
<b>Bibliografía</b>	<b>123</b>
<b>A. Diagramas Esquemáticos</b>	<b>127</b>
<b>B. PCB</b>	<b>129</b>
<b>C. Diseño Software</b>	<b>133</b>
C.1. Clases . . . . .	133
C.2. Diagramas de actividades . . . . .	138

# Índice de tablas

2.1. Niveles de voltaje del receptor del estándar RS-232. . . . .	29
2.2. Niveles de voltaje del transmisor del estándar RS-232. . . . .	29
2.3. Señales del estándar RS-232. . . . .	29
2.4. Códigos de función MODBUS implementados en redes serie. . . . .	33
2.5. PDU de Petición de la función 0x01. . . . .	33
2.6. PDU de Respuesta de la función 0x01. . . . .	34
2.7. PDU de Petición de la función 0x02. . . . .	34
2.8. PDU de Respuesta de la función 0x02. . . . .	34
2.9. PDU de Petición de la función 0x04. . . . .	35
2.10. PDU de Respuesta de la función 0x04. . . . .	35
2.11. PDU de la función 0x05. . . . .	35
2.12. PDU de la función 0x06. . . . .	36
3.1. Tareas <i>hardware</i> y <i>software</i> . . . . .	47
3.2. Relación entre tareas y componentes hardware. . . . .	48
3.3. Parámetros usados en el CAS del sensor de pH. . . . .	52
3.4. Tabla de entradas y salidas de la simulación del CAS para el sensor de pH. . . . .	54
3.5. Parámetros usados en el CAS del sensor de OD. . . . .	56
3.6. Tabla de entradas y salidas de la simulación del CAS para el sensor de OD. . . . .	56
3.7. Parámetros usados en el CAS del sensor de temperatura. . . . .	58
3.8. Tabla de entradas y salidas de la simulación del CAS para el sensor de temperatura. . . . .	59
3.9. Requerimientos formales del <i>software</i> de la RTU. . . . .	62
3.10. Operaciones de la clase RTU. . . . .	64
3.11. Configuración de dispositivos del MCU ATmega8. . . . .	65
3.12. Configuración de terminales del MCU ATmega8. . . . .	65
3.13. Requerimientos formales del <i>software</i> de la MTU . . . . .	66
3.14. Casos de uso del monitoreo principal. . . . .	68
3.15. Casos de uso de la calibración de sensores. . . . .	69
3.16. Requerimientos formales de la aplicación Web para sesión de usuario. . . . .	72
3.17. Requerimientos formales de la aplicación Web para la presentación de mediciones y estados digitales. . . . .	72
3.18. Requerimientos formales de la aplicación Web para gestionar las salidas digitales. . . . .	72
3.19. Descripción de clases de modelos (MVC) de la aplicación Web. . . . .	74
3.20. Descripción de clases de controladores (MVC) de la aplicación Web. . . . .	75
3.21. Casos de uso para sesión de usuarios. . . . .	76
3.22. Casos de uso para funcionalidades Web. . . . .	76

3.23. Descripción de materiales empleados en el diseño PCB del CAS para el sensor pH. . . . .	81
3.24. Descripción de materiales del diseño PCB para el CAS del sensor de OD y temperatura. . . . .	82
3.25. Descripción de materiales empleados en el diseño PCB de la RTU. . . . .	85
3.26. Descripción de materiales empleados en el diseño PCB de la RTU. . . . .	86
4.1. Mediciones $V_{CASpH}$ con variaciones en el voltaje de entrada del sensor de pH. . . . .	94
4.2. Mediciones $V_{CASOD}$ con variaciones en la corriente de entrada del sensor de OD. . . . .	95
4.3. Mediciones $V_{CASTemp}$ con variaciones en el voltaje de entrada del sensor de temperatura. . . . .	96
4.4. Resultados en la prueba de conexión del sensor de pH. . . . .	103
4.5. Resultados en la prueba de conexión del sensor de temperatura. . . . .	103



# Índice de figuras

1.1. Procesos de la PTAR de la UTM. . . . .	2
1.2. Elementos <i>hardware</i> del sistema SCADA. . . . .	3
1.3. Elementos <i>software</i> del sistema SCADA. . . . .	4
2.1. Configuración de un sistema SCADA centralizado. . . . .	8
2.2. Configuración de un sistema SCADA distribuido. . . . .	9
2.3. Niveles de jerarquía de <i>hardware</i> en un sistema SCADA distribuido. . . . .	10
2.4. Configuración de una RTU basada en MCU. . . . .	11
2.5. Funcionamiento de un ADC y un DAC. . . . .	13
2.6. Arquitectura general del <i>software</i> de un sistema SCADA distribuido. . . . .	14
2.7. Niveles de tratamiento de la PTAR. . . . .	19
2.8. Sistema de aireadores. . . . .	20
2.9. Electrodo de pH de la marca <i>Cole Parmer</i> . . . . .	20
2.10. Características del electrodo de pH. . . . .	21
2.11. Funciones de transferencia de entrada y salida del sensor de pH. . . . .	22
2.12. Electrodo de OD de la marca <i>Mettler Toledo</i> . . . . .	22
2.13. Características del electrodo de OD. . . . .	23
2.14. Configuración del sensor LM35. . . . .	24
2.15. Arquitectura de protocolos del modelo de referencia OSI. . . . .	25
2.16. Asignación de terminales y señales del conector DB-9 para el estándar RS-232. . . . .	28
2.17. Interfaz multipunto del estándar RS-485. . . . .	30
2.18. Protocolo Modbus relacionado con el Modelo OSI. . . . .	31
2.19. Unidad de datos de protocolo de MODBUS. . . . .	32
2.20. Unidad de datos de aplicación de MODBUS. . . . .	32
2.21. Modelo OSI relacionado con el modelo TCP/IP. . . . .	36
2.22. Arquitectura MVC del framework Yii. . . . .	40
2.23. Flujo de trabajo en una aplicación Yii. . . . .	41
3.1. Niveles de la metodología CMMI. . . . .	43
3.2. Diagrama general del sistema SCADA. . . . .	44
3.3. Diagrama completo del sistema SCADA propuesto. . . . .	45
3.4. Comunicación de la MTU con la RTU y con el servidor de la aplicación Web y la base de datos. . . . .	47
3.5. Etapas del CAS para el sensor de pH y de temperatura. . . . .	50
3.6. Configuración y simulación de la etapa de adición de $V_{offset}$ y etapa de aislamiento para el CAS de pH. . . . .	51

3.7. Configuración del filtro <i>pasa-bajas</i> con ganancia ajustable. . . . .	52
3.8. Simulación Completa del CAS para el sensor de pH. . . . .	53
3.9. Gráfica de entradas y salidas de la simulación del CAS para el sensor de pH. . . . .	53
3.10. Etapas del CAS para el sensor de OD. . . . .	54
3.11. Configuración de la etapa de conversión-amplificación . . . . .	54
3.12. Configuración del filtro <i>pasa-bajas</i> inversor de ganancia unitaria. . . . .	55
3.13. Simulación del CAS para el sensor de OD. . . . .	56
3.14. Gráfica de entradas y salidas de la simulación del CAS para el sensor de OD. . . . .	57
3.15. Configuración del CAS para el sensor de temperatura. . . . .	57
3.16. Simulación del CAS para el sensor de temperatura. . . . .	58
3.17. Gráfica de entradas y salidas de la simulación del CAS para el sensor de temperatura. . . . .	58
3.18. Configuración de la RTU. . . . .	59
3.19. Distribución de terminales del ATmega8. . . . .	60
3.20. Configuración del DAC0808. . . . .	60
3.21. Configuración del dispositivo MAX489. . . . .	60
3.22. Comunicación serie de la RTU con la MTU. . . . .	61
3.23. Transceptor RS-232/RS-485. . . . .	61
3.24. Diagrama de requerimientos formales del <i>software</i> de la RTU. . . . .	63
3.25. Estructura <i>software</i> de la RTU. . . . .	63
3.26. Diagramas de actividades del <i>software</i> de la RTU. . . . .	64
3.27. Diagrama de requerimientos formales del <i>software</i> de la MTU. . . . .	66
3.28. División de subprocesos del <i>software</i> de la MTU. . . . .	67
3.29. Casos de uso para monitoreo principal. . . . .	69
3.30. Casos de uso para la calibración de sensores. . . . .	70
3.31. Diagrama de actividades de monitoreo automático. . . . .	71
3.32. Diagrama de requerimientos formales de la aplicación Web para sesión de usuario. . . . .	73
3.33. Diagrama de requerimientos formales de la aplicación Web para la presentación de mediciones y estados digitales. . . . .	73
3.34. Diagrama de requerimientos formales de la aplicación Web para gestionar las salidas digitales. . . . .	73
3.35. Clases de modelos (MVC) de la aplicación Web. . . . .	74
3.36. Clases de controladores (MVC) de la aplicación Web. . . . .	75
3.37. Modelo E-R de la aplicación Web. . . . .	76
3.38. Diagrama de casos de uso para sesión de usuarios. . . . .	77
3.39. Diagrama de casos de uso para funcionalidades Web. . . . .	77
3.40. Diagrama de actividades para la presentación de mediciones analógicas. . . . .	78
3.41. Diagrama de actividades para establecer salidas digitales. . . . .	79
3.42. Integración del sistema SCADA propuesto. . . . .	79
3.43. PCB del CAS para el sensor de pH, las imágenes no están a escala. . . . .	81
3.44. Vista 3D del dispositivo CAS para el sensor de pH. . . . .	82
3.45. PCB del CAS para el sensor de OD y temperatura, las imágenes no están a escala. . . . .	83
3.46. Vista 3D del dispositivo CAS para el sensor de OD y temperatura. . . . .	83
3.47. PCB para el módulo del MCU, las imágenes no están a escala. . . . .	84
3.48. Vista 3D del módulo del MCU. . . . .	85
3.49. PCB del transceptor RS-232/RS-485, las imágenes no están a escala. . . . .	86

3.50. Vista 3D del Transceptor RS-232/RS-485. . . . .	87
3.51. Ventana principal de la GUI del <i>software</i> de la MTU. . . . .	87
3.52. Vista principal de la aplicación Web. . . . .	88
4.1. Diagrama de actividades para las pruebas de caja blanca. . . . .	90
4.2. Instrumentos de medición empleados en las pruebas. . . . .	91
4.3. Presentación final de la RTU. . . . .	91
4.4. Configuración de terminales en el gabinete de la RTU. . . . .	92
4.5. Prueba de conexiones de la RTU. . . . .	92
4.6. Mediciones de las salidas en los dispositivos para generar $V_{offset}$ y $V_{POD}$ . . . . .	93
4.7. Configuración para pruebas de variaciones de voltaje. . . . .	93
4.8. Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de pH. . . . .	94
4.9. Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de OD. . . . .	95
4.10. Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de temperatura. . . . .	96
4.11. Diagramas de actividades para las pruebas de caja gris. . . . .	97
4.12. Presentación final del transceptor RS-232/RS-485. . . . .	97
4.13. Elementos requeridos para la comunicación serie. . . . .	98
4.14. Configuración de conexiones para la comunicación entre la RTU y la MTU. . . . .	98
4.15. Configuración del <i>software</i> Docklight. . . . .	99
4.16. Resultados de las pruebas de comunicación serie con cables UTP de diferente longitud. . . . .	100
4.17. Comportamiento del <i>software</i> ante fallos en la comunicación serie. . . . .	101
4.18. Comportamiento del <i>software</i> de la MTU ante fallos de comunicación con la base de datos. . . . .	102
4.19. Conexión del sensor de pH a la RTU. . . . .	103
4.20. Conexión del sensor de temperatura a la RTU. . . . .	104
4.21. Presentación de mediciones de pH en el <i>software</i> de la MTU. . . . .	105
4.22. Pruebas de calibración del sensor de pH. . . . .	106
4.23. Mediciones en el <i>software</i> de la MTU después de realizar la calibración del sensor de pH. . . . .	107
4.24. Presentación de mediciones de temperatura en el <i>software</i> de la MTU. . . . .	108
4.25. Presentación de mediciones de OD en el <i>software</i> de la MTU . . . . .	109
4.26. Pruebas de calibración del sensor de OD. . . . .	110
4.27. Presentación de mediciones de pH en la aplicación Web. . . . .	111
4.28. Presentación de mediciones de temperatura y OD en la aplicación Web. . . . .	112
4.29. Presentación de los registros de mediciones en forma tabular. . . . .	113
4.30. Diagrama de actividades para las pruebas de caja negra. . . . .	114
4.31. Gestión de usuarios en la aplicación Web. . . . .	115
4.32. Presentación y gestión de salidas digitales en la aplicación Web. . . . .	117
4.33. Recepción de la ADU emitida desde la aplicación Web y gestión de salidas digitales desde el <i>software</i> de la MTU. . . . .	118

A.1. Diagrama esquemático del dispositivo CAS para el sensor de pH. . . . .	127
A.2. Diagrama esquemático del dispositivo CAS para el sensor de OD. . . . .	127
A.3. Diagrama esquemático del dispositivo CAS para el sensor de temperatura. . . . .	128
A.4. Diagrama esquemático para las entradas y salidas del MCU. . . . .	128
B.1. PCB del CAS para el sensor de pH. . . . .	129
B.2. PCB del CAS para el sensor de OD y temperatura. . . . .	130
B.3. PCB para el módulo del MCU. . . . .	131
B.4. PCB del transceptor RS-232/RS-485. . . . .	131
C.1. Diagrama de clases para monitoreo de entradas analógicas. . . . .	134
C.2. Diagrama de clases para monitoreo de entradas digitales. . . . .	135
C.3. Diagrama de clases para monitoreo y control de salidas digitales. . . . .	136
C.4. Diagrama de clases para calibración de sensores. . . . .	137
C.5. Diagrama de actividades de la gestión de la interfaz serial. . . . .	138
C.6. Diagrama de actividades del procesamiento de ADUs. . . . .	138
C.7. Diagrama de actividades para establecer el tiempo de muestreo. . . . .	139
C.8. Diagrama de actividades para calibrar el sensor de pH. . . . .	139
C.9. Diagrama de actividades para calibrar el sensor de OD. . . . .	140

# 1

## Introducción

El presente trabajo de tesis aborda el área de monitoreo remoto. Generalmente un sistema de monitoreo remoto se divide en las siguientes etapas: “la adquisición de datos de los sensores, un sistema donde se almacenan registros de mediciones y una infraestructura de comunicaciones para conectar el sistema de almacenamiento de datos con una red de comunicaciones pública como es la Internet” [11, 25]. En esta infraestructura se ejecuta una aplicación Web que presenta dicha información al usuario final a través de una interfaz gráfica de usuario (GUI, *Graphic User Interface*) [13, 29].

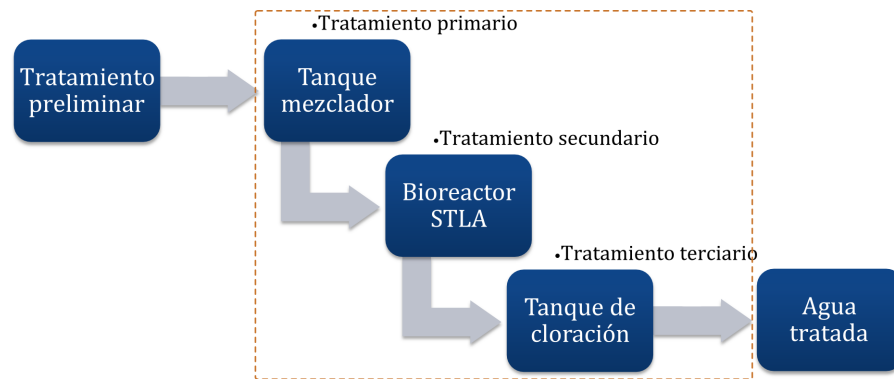
El monitoreo remoto en plantas industriales, a menudo consiste de una compleja red de diversos equipos comunicándose a través de varios protocolos. Dichos equipos están constituidos por *hardware* y *software* (sistemas empotrados, *Embedded Systems*) [11, 25].

El monitoreo y automatización remota se engloba en el concepto de SCADA (Control con Supervisión y Adquisición de Datos, *Supervisory Control and Data Acquisition*) [4]. Se da el nombre de SCADA a cualquier sistema que consta de componentes *hardware* y *software*, el cual permite el acceso a datos remotos y el control de un proceso (industrial) mediante el uso de sistemas de comunicaciones. En un sistema SCADA, una unidad terminal maestra (MTU, *Master Terminal Unit*) generalmente consiste en una computadora que se encarga de monitorizar una o varias unidades terminales remotas (RTU, *Remote Terminal Unit*). La comunicación entre la MTU y la RTU se realiza a través de protocolos industriales de control (por ejemplo Modbus).

Actualmente, los sistemas SCADA son usados en aplicaciones como: plantas de energía, telecomunicaciones, transporte, tratamiento de agua y desechos, y otras. En un sistema SCADA, los componentes *hardware* obtienen datos y los emiten a una computadora, la cual cuenta con *software* para procesamiento y presentación de información por medio de GUIs.

La Universidad Tecnológica de la Mixteca (UTM) cuenta con una planta de tratamiento de aguas residuales (PTAR), la cual consta de una etapa preliminar (sistema de rejillas o cribas) y tres procesos principales (Figura 1.1)[6]:

- *Tratamiento primario*: En este proceso se utiliza un tanque mezclador para homogenizar el efluente (agua residual) y controlar el flujo de entrada al siguiente tratamiento.
- *Tratamiento secundario*: Este tratamiento emplea un Sistema de Tratamiento por Lodos Activados (STLA), en donde la biodegradación de la materia orgánica se lleva a cabo en un bioreactor.
- *Tratamiento terciario*: En este proceso se utiliza un tanque en el que se dosifica una solución de hipoclorito de sodio para la eliminación de bacterias.



**Figura 1.1.** Procesos de la PTAR de la UTM.

Los parámetros importantes en el proceso de tratamiento son [17]:

- *pH*: La medición de pH es importante para el tratamiento primario y secundario. Este parámetro determina si el agua es ácida o alcalina, por lo que el valor del pH debe estar en un rango de 7 a 7.5 unidades [37].
- *Oxígeno Disuelto (OD)*: Para el tratamiento secundario este parámetro es fundamental, ya que determina la cantidad de oxígeno ( $O_2$ ) disponible por los microorganismos presentes en el tratamiento secundario. Se recomienda que el valor del OD esté por arriba de  $3 \text{ a } 4 \frac{\text{mg}O_2}{\text{L}}$  [37].
- *Temperatura*: La temperatura es importante para asegurar el crecimiento óptimo de los microorganismos en el tratamiento secundario, se recomienda que la temperatura esté entre  $28 \text{ y } 30^\circ\text{C}$  [37].

El monitoreo de los parámetros anteriormente mencionados, es un inicio para una futura automatización de la PTAR de la UTM, con lo cual se podría garantizar una minimización significativa en los costos de operación, que son generados por los equipos de la planta que se mantienen en operación las 24 horas del día, lo cual significa un gran consumo de energía eléctrica [30].

Por lo anterior, este proyecto de tesis plantea el diseño y construcción de un sistema de monitoreo remoto para la PTAR de la UTM. El sistema que se desarrolló tiene como objetivo implementar un sistema SCADA para monitorizar de manera remota y automatizada el estado de pH, OD y temperatura de la PTAR de la UTM. El sistema almacenará registros de mediciones y proporcionará una GUI a través de una aplicación Web.

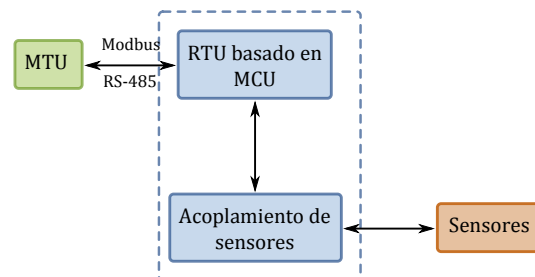
## 1.1. Planteamiento del Problema

La necesidad inicial, que generó este trabajo de tesis, es la automatización de la PTAR. El propósito de la automatización es hacer eficiente la operación del sistema de aireación, el cual actualmente están en operación las 24 horas del día y generan un gran consumo de energía eléctrica. La automatización permite que el consumo de energía sea sustancialmente bajo, lo cual tiene un impacto directo en los costos de operación [30].

Dicho lo anterior, el primer paso para la automatización de la PTAR es la caracterización de la planta. Ésto se logra con un sistema de monitoreo de los parámetros que se han mencionado previamente. Por lo que, este trabajo de tesis propone el diseño y construcción de un prototipo de sistema de monitoreo para la PTAR de la UTM, dicho sistema será implementado bajo el concepto de sistemas SCADA.

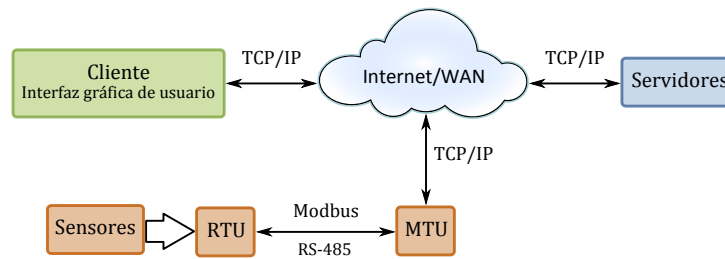
El sistema SCADA servirá para monitorizar de manera remota y automatizada el estado de pH, OD y temperatura de la PTAR de la UTM. Dicho sistema constará de sensores de propósito industrial, se almacenará la información de las mediciones en un servidor de bases de datos y ofrecerá acceso Web mediante una GUI que permitirá visualizar el estado de los parámetros medidos de manera gráfica y tabular.

La etapa de elementos *hardware* del sistema SCADA tiene como entradas las órdenes y peticiones provenientes de una MTU y como salidas la interacción con los sensores, como se ilustra en la figura 1.2. El *hardware* estará constituido por los siguientes elementos: una etapa de acoplamiento de sensores, en la cual se realiza el acondicionamiento de señal de los sensores, esta etapa es necesaria porque los sensores de pH, OD y temperatura requieren el uso de amplificadores operacionales de instrumentación para poder operar las señales de dichos sensores; un sistema basado en un microcontrolador (MCU, *Microcontroller Unit*) como unidad de procesamiento remota (RTU); y una interfaz de comunicación serie RS-485 para establecer la comunicación de la RTU con la MTU mediante el protocolo Modbus.



**Figura 1.2.** Elementos *hardware* del sistema SCADA.

Los elementos *software* tendrán como entradas la interacción de un usuario del sistema (Administrador) y como salidas las solicitudes de monitoreo hacia la RTU, como se ilustra en la figura 1.3. El software estará constituido por los siguientes elementos: el *software* de la MTU, que se encargará de enviar solicitudes de monitoreo a la RTU y almacenar registros de mediciones en el servidor de base de datos; una aplicación Web que se ejecutará en un servidor, que a través de una GUI permitirá la visualización de los registros de mediciones de manera gráfica y tabular y permitirá la interacción del usuario final con el sistema mediante el uso de navegadores Web como *Chrome*, *Mozilla*, *Opera*, *Internet Explorer*, y otros.



**Figura 1.3.** Elementos *software* del sistema SCADA.

## 1.2. Justificación

En la planta de tratamiento de aguas residuales de la Universidad Tecnológica de la Mixteca aún no se cuenta con un sistema de monitoreo automatizado, por lo que este proceso se realiza de manera manual por los encargados.

Debido a la situación anterior, es necesario diseñar y construir un prototipo de sistema de monitoreo. Este sistema debe permitir la adquisición de datos de los parámetros involucrados en los procesos de tratamiento de la planta, almacenar dicha información en forma de registros en una base de datos, los cuales puedan ser visualizados por los usuarios finales a través de una GUI en una aplicación Web.

## 1.3. Hipótesis

Del prototipo de sistema de monitoreo para la planta de tratamiento de aguas residuales de la Universidad Tecnológica de la Mixteca, se obtendrá una herramienta para almacenar y acceder a la información del estado actual y estados previos de temperatura, pH y Oxígeno Disuelto, así como realizar un análisis en base a un historial de registros de mediciones y conocer el comportamiento de la planta.

## 1.4. Objetivos

### 1.4.1. Objetivo general

Diseñar y construir un prototipo de Sistema de Monitoreo para la Planta de Tratamiento de Aguas Residuales de la Universidad Tecnológica de la Mixteca.



### 1.4.2. Objetivos específicos

- Diseñar un sistema de acondicionamiento de señal para cada uno de los sensores haciendo uso de amplificadores operacionales de instrumentación.
- Usar un MCU de la marca ATMEL para implementar la RTU.
- Hacer uso del protocolo Modbus para establecer la comunicación entre la RTU y la MTU.
- Realizar la comunicación entre la RTU y la MTU a través de una interfaz serie RS-485.
- Almacenar registros de mediciones en un servidor de base de datos MySQL.
- Diseñar una aplicación Web usando el *framework Yii* para PHP.

## 1.5. Contenido del Documento de Tesis

La estructura de este documento de tesis es la siguiente:

En el capítulo 1 se presenta una breve introducción acerca de los sistemas de monitoreo, el planteamiento del problema, la justificación, hipótesis y los objetivos de este proyecto de investigación.

El capítulo 2 presenta las bases teóricas que sustentan este proyecto de investigación. Se describen las principales características de los sistemas SCADA, sus elementos *hardware* y *software*, los protocolos de comunicación, se da una descripción de la PTAR de la UTM, los sensores con los que cuenta la UTM y las herramientas tecnológicas para desarrollar el sistema.

En el capítulo 3 se detalla el desarrollo del sistema empleando la metodología de desarrollo para mejoramiento de procesos de producción y la metodología de sistemas empotrados, por lo que se expone de manera detallada las fases de desarrollo del sistema SCADA.

El capítulo 4 presenta la verificación y validación del sistema SCADA en base a los requerimientos.

En el capítulo 5 se plantean las conclusiones y los futuros trabajos de investigación.

Por último, se presentan las referencias bibliográficas utilizadas, los diagramas esquemáticos, el diseño completo del circuito impreso y diagramas del diseño software.



# 2

## Marco Teórico

### 2.1. Sistemas SCADA

La telemetría, tele-medida o medida a distancia, es requerida para conectar equipos y sistemas separados por distancias muy grandes, principalmente en procesos industriales, procesos de manufactura moderna, industria minera, industrias de seguridad y otras. El rango de distancia puede ser desde unos cuantos metros a miles de kilómetros. La telemetría es usada para enviar órdenes, programas y recibir información de monitoreo desde ubicaciones remotas (Monitoreo Remoto) [4, 29].

Un sistema de monitoreo remoto (*Remote Monitoring System*) se define como un sistema que permite realizar rutinariamente monitoreo y supervisión a través de una infraestructura de red [11, 25].

El concepto de Control con Supervisión y Adquisición de Datos (SCADA) se refiere a la combinación de telemetría y adquisición de datos. Se da el nombre de SCADA a cualquier sistema que consta de componentes *hardware* y *software*, el cual permite el acceso a datos remotos y el control de un proceso mediante el uso de sistemas de comunicaciones.

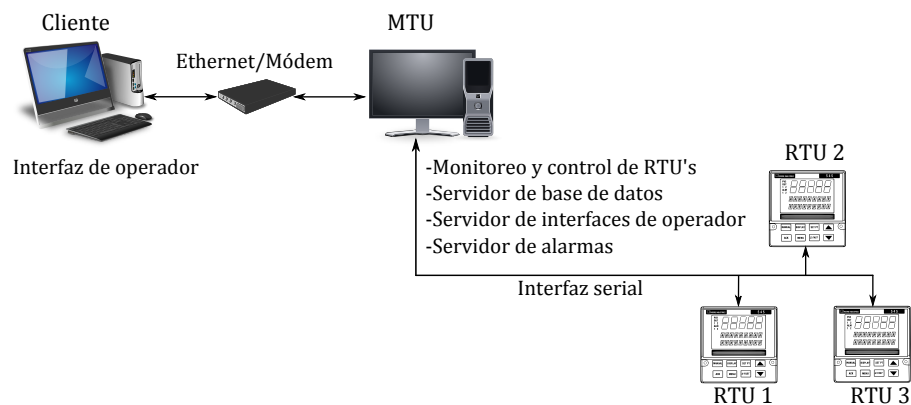
Actualmente, los sistemas SCADA son usados por las empresas industriales modernas en aplicaciones como: plantas de energía, telecomunicaciones, transporte, tratamiento de agua y desechos, etc., en donde los componentes *hardware* obtienen los datos y los emiten a una computadora, la cual cuenta con *software* para procesamiento de la información y para mostrarla en pantallas personalizadas (GUIs). De esta manera, los sistemas SCADA permiten a los operadores monitorizar y controlar los equipos desde una estación central.

Los sistemas SCADA han evolucionado en paralelo con las tecnologías modernas, y con ello la mayoría de dichos sistemas presentan dos topologías:

- **Sistemas SCADA centralizados:** En los sistemas centralizados o monolíticos, una computadora central es la encargada de realizar el monitoreo completo de la planta (industrial) y toda la información es almacenada en una base de datos dentro de la misma computadora [4]. Este tipo de red SCADA se implementa para comunicarse con RTUs en el área de campo y a través de buses de campo (*FieldBus*) [46], tal como se ilustra en la figura 2.1.

Las desventajas comunes de los sistemas SCADA centralizados o monolíticos son: para un sistema pequeño los costos iniciales son bastante altos y debido al tamaño fijo del sistema, una expansión gradual para la mejora de la planta no es posible.

- **Sistemas SCADA distribuidos:** Un sistema SCADA distribuido es soportado por un conjunto de computadoras, las cuales están interconectadas a través de una red y cuyos componentes *software* se encuentran dispersos entre dichas computadoras para realizar tareas de manera colaborativa, lo cual permite la escalabilidad del sistema [4, 5, 44]. Un sistema SCADA distribuido generalmente hace uso de la arquitectura Cliente/Servidor.



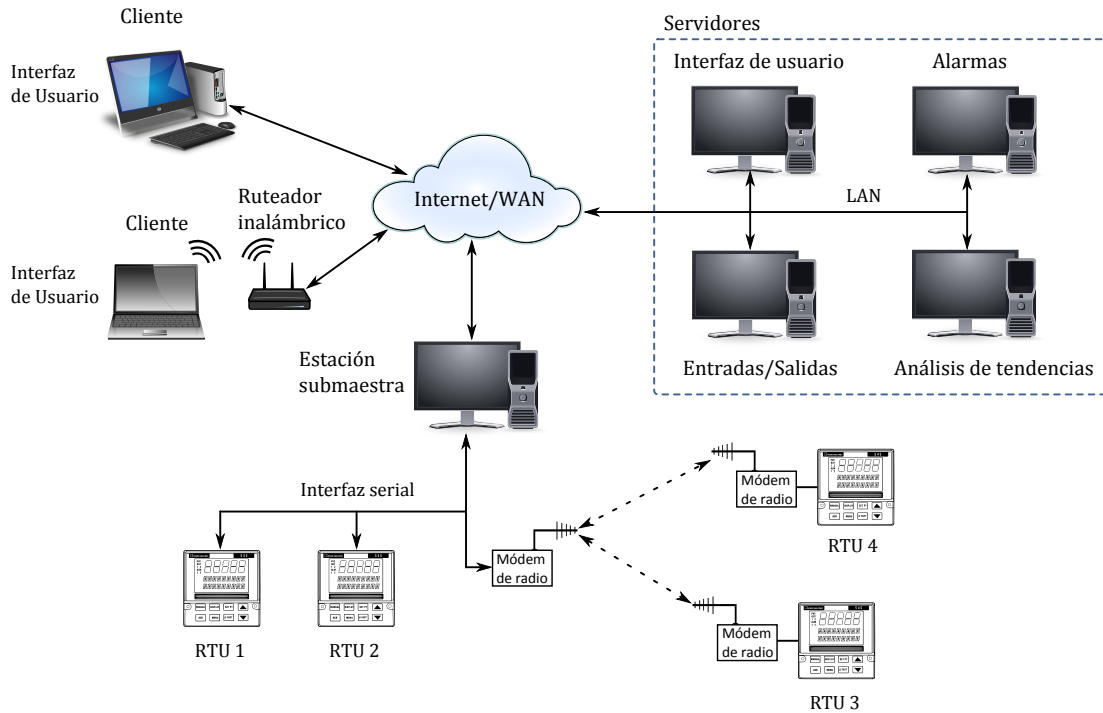
**Figura 2.1.** Configuración de un sistema SCADA centralizado.

Debido a las desventajas que presentan los sistemas SCADA centralizados, en este proyecto de investigación se implementará un sistema SCADA distribuido, por lo que en esta sección se hará énfasis a la teoría relacionada con sistemas SCADA distribuidos.

Ahora, cada computadora dentro de un sistema SCADA distribuido tiene una función específica, por ejemplo: algunas computadoras sirven como procesadores de comunicación con dispositivos de campo o RTUs; otras sirven como interfaces de operador que proporcionan interfaces humano máquina (HMI, *Human Machine Interface*); y algunas computadoras se utilizan como servidores de bases de datos para procesamiento de información [46]. En la figura 2.2 se ilustra la configuración típica de un sistema SCADA distribuido.

La distribución de funciones provee una mayor capacidad de procesamiento comparado con un sistema centralizado. Las tareas esenciales de un sistema SCADA distribuido, que requieren procesamiento de manera individual, son las siguientes [4, 46]:

- **Entradas y salidas:** Son la interfaz entre el sistema de monitoreo y las instalaciones de la planta.



**Figura 2.2.** Configuración de un sistema SCADA distribuido.

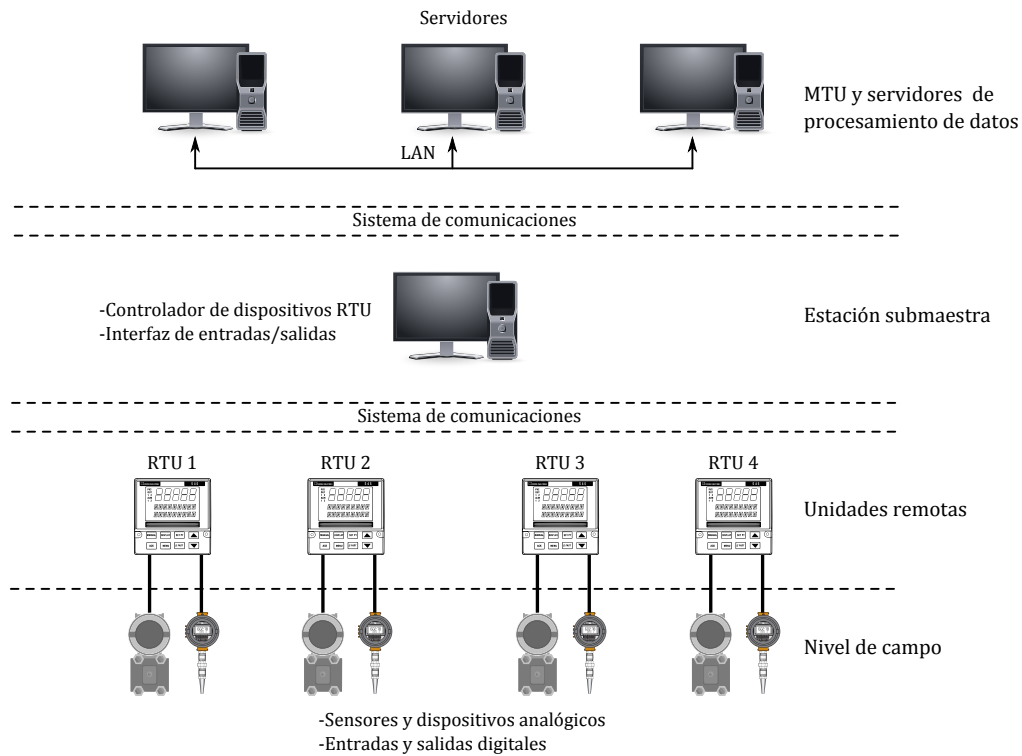
- **Bases de datos:** Se encarga de almacenar todos los datos provenientes de las entradas y salidas, al mismo tiempo permite el acceso a esta información, de tal manera que las demás tareas puedan realizar sus funciones.
- **Alarmas:** Esta tarea administra todas las alarmas del sistema a través de la comparación de las entradas con un valor de umbral predeterminado.
- **Tendencias:** Captura todos los datos de las variables que se desean monitorizar con el propósito de mostrar su evolución con respecto al tiempo.
- **Reportes:** Los reportes son generados a partir de los datos de la planta y pueden ser creados a petición del usuario u operador.
- **Presentación:** A través de una GUI gestiona y presenta la información al operador [13].

### 2.1.1. Componentes *hardware* de un sistema SCADA distribuido

En esta sección se describen los componentes *hardware* (figura 2.3) de un sistema SCADA distribuido, los cuales se clasifican en los siguientes niveles de jerarquía [4, 29, 46]:

- Dispositivos de instrumentación y control a nivel de campo<sup>1</sup>.
- Uno o más dispositivos de interfaces de datos de campo. Dichos dispositivos son las RTUs que desempeñan el papel de interfaz con los dispositivos de instrumentación y control a nivel campo.

<sup>1</sup>Los dispositivos de campo son los que están en contacto directo con la planta industrial.



**Figura 2.3.** Niveles de jerarquía de *hardware* en un sistema SCADA distribuido.

- Un servidor o conjunto de servidores como sistema central, los cuales usualmente son denominados como centro del sistema SCADA, estaciones maestras o MTUs. La MTU de un sistema distribuido consta de un conjunto de servidores que realizan las siguientes tareas: gestión de entradas y salidas, bases de datos, alarmas, tendencias, reportes y presentación.
- Un sistema de comunicaciones para el intercambio de datos entre RTUs y MTUs.

#### 2.1.1.1. Instrumentación

La instrumentación electrónica se aplica en el sensado y procesamiento de la información proveniente de variables físicas<sup>2</sup> y de otro tipo<sup>3</sup>, a partir de las cuales se realiza el monitoreo y control de procesos, empleando dispositivos y tecnologías electrónicas.

Para su funcionamiento la instrumentación se divide en tres etapas:

- *Los sensores:* Son aquellos que transforman la magnitud de una variable física que se desea medir a una señal eléctrica.
- *Acondicionamiento:* Como su nombre lo indica, acondiciona los niveles de la señal de salida de un sensor. Esto se debe a que en la mayoría de los casos la señal de los sensores suele no ser adecuada para su procesamiento. En general los sensores entregan señales muy

<sup>2</sup>Temperatura, presión, volumen, etc.

<sup>3</sup>Variables de referencia en dispositivos de control o variables de umbral en sistemas de alarmas.

pequeñas y vulnerables a ruido, por lo que el acondicionamiento generalmente consiste de dos subetapas: la etapa de amplificación y la etapa de filtrado. Dichas subetapas se realizan a través de un amplificador operacional de instrumentación.

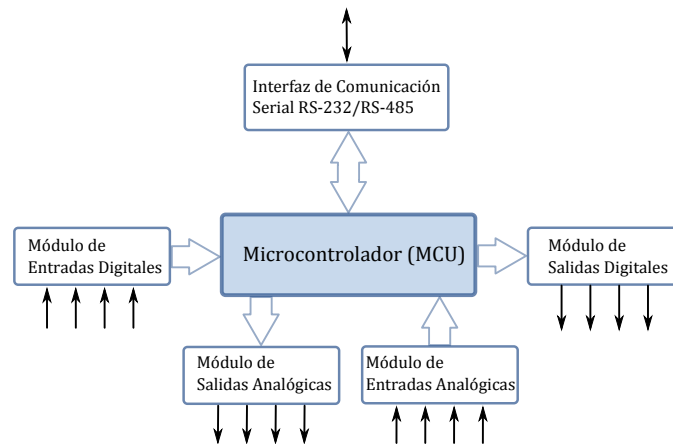
- **Digitalización:** La digitalización proporciona un código digital (binario) equivalente a la señal de entrada proveniente de la etapa de acondicionamiento. Dicho código permite el procesamiento de las señales a través de sistemas basados en MCU, procesadores o computadoras.

### 2.1.1.2. Componentes de la RTU

Una RTU es una unidad independiente cuyas tareas son: adquisición de datos, gestión de equipos de control en un proceso industrial (ubicado en un lugar remoto) y transferir dichos datos a la MTU [4].

Generalmente una RTU consiste en un sistema empotrado (*Embedded System*), el cual tiene como unidad principal un procesador [25]. La complejidad de la unidad central de procesamiento determina la clasificación del sistema empotrado, microprocesador ( $\mu P$ ) o MCU.

La ventaja de un diseño basado en MCU sobre uno basado en  $\mu P$  radica en que el MCU incluye internamente tres unidades funcionales: unidad central de procesamiento (CPU, *Central Processing Unit*), memoria de código/datos y periféricos de entrada y salida. En cambio, un  $\mu P$  necesita de las mismas unidades funcionales pero de manera externa. Un diseño basado en MCU reduce significativamente el tiempo de desarrollo, el tamaño del dispositivo y el costo [25]. La figura 2.4 muestra la configuración típica de una RTU basada en MCU.



**Figura 2.4.** Configuración de una RTU basada en MCU.

La configuración de una RTU basada en MCU está formada por los siguientes módulos *hardware* [4, 29]:

- **Unidad central de procesamiento:** La CPU es la encargada de procesar las instrucciones almacenadas en la memoria de código. Por lo tanto, la CPU tiene acceso a las memorias

(de datos y de código), realiza los cálculos asociados o requeridos, controla periféricos de entrada y salida, y periféricos de interfaz de comunicación.

Las instrucciones almacenadas en la memoria de código forman la configuración y el programa de control propias de la RTU. La configuración y el programa de control establecen los parámetros para: periféricos de entrada y salida (analógicas y digitales); la comunicación entre la RTU y la MTU, la cual generalmente consiste en un protocolo industrial de control y monitoreo [25]; y, los programas de aplicación de la RTU, tal es el caso de un control analógico de lazo cerrado PID (*Proporcional Integral Diferencial*).

- *Entradas analógicas*: Estos periféricos permiten la adquisición de datos de las señales que proporcionan los módulos de instrumentación (sensores y acondicionamiento).

El elemento principal de este módulo es un convertidor análogo-digital (ADC, *Analog-Digital Converter*), cuya función es tomar un voltaje analógico como entrada y entregar como salida un código digital (código binario) correspondiente a dicho voltaje de entrada (figura 2.5a). Este código digital es adecuado para su procesamiento a través de un MCU.

- *Salidas analógicas*: Un periférico de salida analógica en general consiste de dos módulos: un registro de salidas digitales (usualmente 8 bits) y un convertidor digital-análogo (DAC, *Digital-Analog Converter*). El DAC es un dispositivo que recibe un código digital (binario) y genera un nivel de voltaje de salida (analógico), como se ilustra en la figura 2.5b.

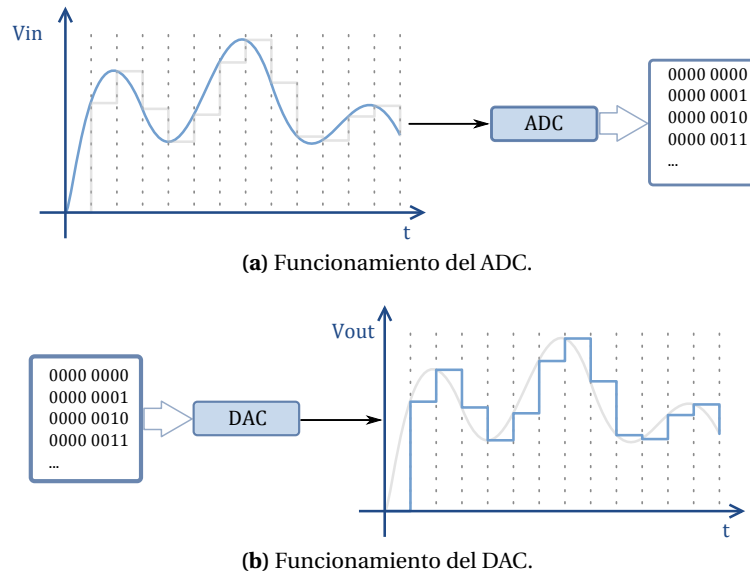
En una RTU, una salida analógica frecuentemente se usa como voltaje de referencia para el controlador de algún dispositivo analógico, como puede ser el controlador de un motor, una bomba, un sistema de enfriamiento, un calefactor, etc. Una salida analógica también puede ser usada como fuente de alimentación para dispositivos de bajo consumo de energía, por ejemplo, la alimentación o polarización de un sensor de OD y otros.

- *Entradas y salidas digitales*: Las entradas y salidas digitales se usan para indicar información como señales de estados lógicos de alarmas y otros dispositivos. Por ejemplo el estado (encendido o apagado) de un relevador que controla una válvula, un actuador o cualquier dispositivo analógico o de las alarmas que indican el estado crítico de alguna variable en el proceso de una RTU. Dichas alarmas son procesadas por la MTU para tomar medidas de control o simplemente mostrarlas en una GUI.

Es importante considerar un aislamiento adecuado para las entradas y salidas digitales, ya que los periféricos de un MCU normalmente operan con niveles de voltaje TTL. Una buena opción es emplear dispositivos de aislamiento óptico, relevadores o transformadores.

- *Interfaces de comunicación*: En la comunicación de la RTU con la MTU se pueden manejar diferentes interfaces de comunicación según sean las necesidades. Actualmente, las interfaces de comunicación que una RTU debe soportar abarcan: interfaces serie RS-232/RS-485, línea telefónica, fibra óptica, microondas, satélite o radio vía portadora (VHF, UHF o 900 MHz).





**Figura 2.5.** Funcionamiento de un ADC y un DAC.

### 2.1.1.3. Componentes de la MTU

Una MTU es el centro de un sistema SCADA y está formada por un servidor o conjunto de servidores (usualmente computadoras) interconectados a través de una red para realizar tareas de manera colaborativa. La MTU es la que inicia toda la comunicación con las RTUs, reúne datos, almacena información, realiza procesamiento y proporciona GUIs [4, 29, 46].

Las funciones principales de una MTU son obtener periódicamente datos de campo de las RTUs, monitorizar y controlar remotamente los dispositivos a través de las estaciones de operador. Los componentes y características básicas de una MTU son:

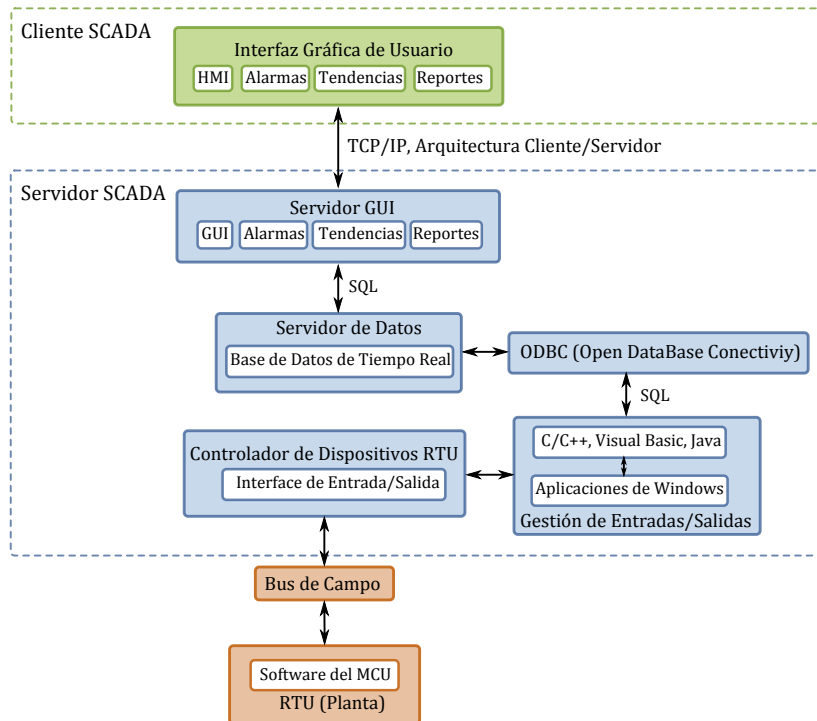
- *Servidores e interfaz de operador:* Una MTU consta de una o más estaciones de operador, las cuales cuentan con una interfaz de usuario para mostrar el estado de las RTUs y permiten el control al operador; un servidor para el manejo de bases de datos; el servidor de alarmas; un servidor de presentación, el cual muestra la evolución de las variables monitoreadas a través de gráficas de tendencias; un servidor para generar reportes; un servidor de interfaces de usuario; y un servidor de entradas y salidas, el cual tiene bajo su control a los dispositivos de campo (RTU), este servidor también es conocido como el controlador de las RTUs.
- *Software de la MTU:* El *software* en una MTU se divide en dos categorías: el sistema operativo, el cual puede ser: un sistema *MS-DOS*, *Windows*, *Windows NT* y los distintos sistemas *Unix* (*Linux* y *Mac OS*) y el *software* del sistema SCADA, generalmente consiste en la adquisición de datos, control, archivado o almacenamiento en base de datos y la GUI o HMI.

- *Interfaces de comunicación:* La MTU establece la comunicación con las RTUs a través de una interfaz de comunicación. Esta comunicación generalmente se realiza haciendo uso de buses de campo<sup>4</sup>.

Para distribuir el procesamiento a través de los múltiples servidores, los sistemas SCADA distribuidos hacen uso de las siguientes tecnologías[8]: redes de área local (LAN, *Local Area Network*), tal es el caso de la familia de redes Ethernet o estándar IEEE 802.3; las redes de área amplia (WAN, *Wide Area Network*); y la suite de protocolos de Internet, de los cuales los más conocidos globalmente son el Protocolo de Control de Transmisión (TCP, *Transmission Control Protocol*) y el Protocolo de Internet (IP, *Internet Protocol*), estos protocolos forman parte de un conjunto de protocolos conocidos como TCP/IP.

### 2.1.2. Componentes *software* de un sistema SCADA distribuido

El *software* de un sistema SCADA distribuido es multitareas y está basado principalmente en bases de datos de tiempo real localizados en uno o más servidores. La arquitectura general del *software* de un sistema SCADA se ilustra en la figura 2.6 y sus componentes son [4, 29, 46]:



**Figura 2.6.** Arquitectura general del *software* de un sistema SCADA distribuido.

- Interfaz de usuario.

<sup>4</sup>El término “bus de campo” engloba varios protocolos de control industrial, por ejemplo Modbus, PROFIBUS, WorldFIP y CAN

- Servidor de representaciones gráficas de tendencias.
- Servidor de alarmas y eventos.
- Servidor para generación de reportes.
- Bases de datos.
- Acceso a los datos.
- Manejo de redes.
- Interfaces RTU.

Generalmente, los componentes de interfaz de usuario, representaciones gráficas, servidor de alarmas y eventos, servidor de reportes y servidor de bases de datos, consisten en aplicaciones distribuidas con arquitectura Cliente/Servidor, de las cuales las aplicaciones Web son las más usadas globalmente. En cambio, las aplicaciones para interfaces RTU se desarrollan empleando lenguajes de programación como C/C++, Visual Basic y Java.

#### 2.1.2.1. Comunicación con dispositivos de campo

Los servidores de datos realizan solicitudes periódicamente a los controladores de dispositivos de campo (RTU). La comunicación entre la MTU y la RTU globalmente se realiza a través de buses de campo, por ejemplo Modbus, PROFIBUS, WorldFIP y CAN. Por lo tanto el *software* SCADA debe proporcionar las herramientas necesarias para llevar a cabo dicha comunicación [4, 46].

#### 2.1.2.2. Comunicación entre aplicaciones

La comunicación interna entre el conjunto de servidores para procesamiento de datos, se realiza usando la arquitectura distribuida Cliente/Servidor. En general esta comunicación se realiza mediante la suite de protocolos TCP/IP [4, 46].

Los métodos más conocidos para el intercambio de información entre las aplicaciones de los servidores son ODBC (*Open Data Base Connectivity*) y SQL (*Structured Query Language*) [29].

##### 2.1.2.2.1. ODBC

ODBC es una tecnología de *Microsoft Windows* y es un estándar que permite a las aplicaciones el acceso a datos en sistemas de gestión de bases de datos (DBMS, *Data Base Management Systems*) utilizando SQL como método estándar de acceso [29].

Una aplicación desarrollada en lenguajes estándares de programación (C, C++, Visual Basic o Java), necesita la inclusión del controlador ODBC correspondiente para tener acceso a distintas bases de datos, dicho controlador es una librería de enlace dinámico (DLL, *Dynamic Link Library*).

La interfaz ODBC define los siguientes aspectos: una librería de llamadas a funciones ODBC, la sintaxis SQL necesaria, códigos de error estándar, el método de conexión a un DBMS y el formato de presentación de los datos.

### 2.1.2.2.2. SQL

SQL es un estándar para la comunicación con bases de datos, que permite una interfaz común para el acceso a los datos por parte de cualquier aplicación que se apegue a dicho estándar [29].

Las posibilidades de esta tecnología incluyen:

- *Procedimientos*: Son bibliotecas de órdenes almacenadas en la base de datos que permiten reducir el tráfico en la red y simplifican los procedimientos de acceso a los usuarios de las bases de datos.
- *Eventos*: Son órdenes que se activan de forma automática bajo unas ciertas condiciones, facilitando el mantenimiento de integridad de datos.
- *Replicación*: Permite la duplicación y sincronización de bases de datos; la accesibilidad, permite el intercambio o envío de información basándose en eventos.

### 2.1.2.3. Bases de datos relacionales

Las bases de datos relacionales (*Relational Data Base*) proporcionan estructuras de datos, independientemente del tipo de aplicaciones que acceden a éstos o de su estructura [29].

Una base de datos relacional es un conjunto de tablas de datos que contienen campos. Éstos sirven como nexos de unión a través de los cuales se pueden establecer múltiples combinaciones. Las combinaciones posibles prácticamente son ilimitadas, todo depende de la configuración del método de búsqueda (consulta, *Query*), o el tipo de datos que se requiera consultar.

Este tipo de organización permite el uso de la arquitectura Cliente/Servidor y con ello se simplifican la administración de los datos y las aplicaciones que los usan, se disminuye el espacio de almacenamiento y se reducen los problemas asociados a las bases de datos redundantes.

Los usuarios y las aplicaciones pueden acceder a los datos de forma rápida y sencilla, ya que pueden personalizar sus propias consultas y obtener los datos de acuerdo a sus necesidades.

### 2.1.2.4. Interfaces de usuario (UI)

El término “interfaz gráfica de usuario” (GUI) se refiere a los métodos y dispositivos usados en la interacción entre máquinas y los seres humanos que las usan. La interfaz de usuario de un sistema mecánico, un vehículo o las instalaciones de una industria, frecuentemente es llamada HMI [29].

En un sistema de control industrial, una HMI se usa para proporcionar a los operadores información de una máquina o planta y permite monitorizar, controlar y archivar dicho sistema. Por lo tanto, las tareas fundamentales que una HMI debe realizar son: la comunicación de información de la máquina al usuario y la comunicación de información del usuario a la máquina.

Las HMI de tipo supervisoras son usadas en sistemas en donde la distancia, entre la estación de control y la estación remota (o planta), es considerablemente grande. Las HMI de este tipo

son propias de un sistema SCADA, en el cual se encuentran dispositivos tales como bombas, ventiladores o indicadores de purificación de aguas.

Una HMI de un sistema SCADA permite que desde una interfaz de operador se pueda monitorizar completamente una planta y muestra una representación gráfica del estado actual de cada componente, el estado de las alarmas que indican cuando algún parámetro está fuera del rango de operación adecuado. Estas tareas se realizan gracias a la retroalimentación por parte de los sensores que miden los parámetros de los procesos en la planta.

### 2.1.3. SCADA en la Internet

La Internet es simplemente la red virtual en donde todas las estaciones se interconectan fácilmente sin tener que preocuparse por las conexiones físicas.

Actualmente la Internet hace uso de la suite de protocolos abiertos TCP/IP [4, 48], en donde el protocolo IP ofrece la capacidad de realizar el ruteo permitiendo que los paquetes sean enviados a través de redes interconectadas con topologías diferentes y el protocolo TCP permite que los paquetes sean enviados desde un punto a otro garantizando que lleguen a su destino.

El protocolo de aplicación más usado en Internet es el protocolo de transferencia de hipertexto (HTTP, *Hypertext Transfer Protocol*), el cual es la base para la WWW (*World Wide Web*). El protocolo HTTP es de arquitectura Cliente/Servidor, por lo tanto, el uso más frecuente que se le da es en un navegador (cliente) que accede a un servidor Web. Los datos que pueden ser transferidos por este protocolo son: texto nativo, hipertexto, audio, imágenes, etc.

El protocolo HTTP permite el desarrollo de aplicaciones de interfaces gráficas de usuario para sistemas SCADA, ya que es posible transferir representaciones gráficas y estados de los dispositivos de campo. Además, la arquitectura Cliente/Servidor permite que más de un usuario u operador acceda al sistema SCADA a través de un navegador Web ejecutado en una computadora desde cualquier lugar, permitiendo el monitoreo y control de la planta como si el operador estuviese en las instalaciones de la misma.

## 2.2. Planta de Tratamiento de Aguas Residuales de la UTM

### 2.2.1. El Agua Residual

Esencialmente el agua residual, es el agua usada por una comunidad en una gran variedad de aplicaciones. Según la Norma Oficial Mexicana *NOM-001-ECOL-1996* emitida por la Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT) [32], se definen las aguas residuales como:

*“Las aguas de composición variada provenientes de las descargas de usos municipales, industriales, comerciales, de servicios, agrícolas, pecuarios, domésticos, incluyendo fraccionamientos y en general cualquier otro uso, así como la mezcla de ellas”.*

Las fuentes principales de aguas residuales son [6, 37]:

- *Las aguas residuales domésticas:* Engloban las aguas residuales generadas por los hogares, instalaciones comerciales e instituciones públicas. La contaminación es orgánica en su mayoría, compuesta de materia fecal, papel, jabón, suciedad, restos de alimentos (basura) y otras sustancias.
- *Las aguas residuales industriales:* En donde predominan los desechos industriales, están en función del tipo y tamaño de la industria. Algunas son aguas de enjuague; otras se encuentran muy cargadas de materia orgánica o mineral, o con sustancias corrosivas, venenosas, inflamables o explosivas.

### 2.2.2. Sistema de tratamiento de aguas residuales de la UTM

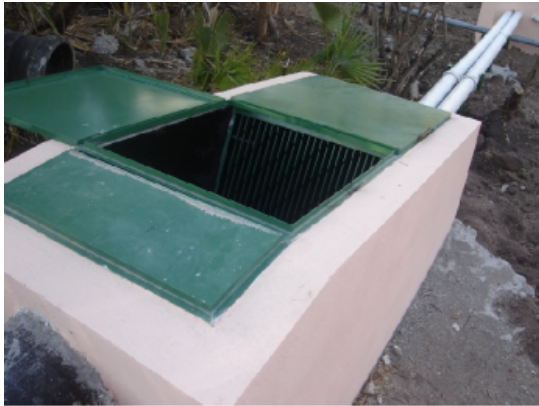
Los métodos de tratamiento de aguas residuales en donde predomina la aplicación de fuerzas de contacto físico son conocidos como unidades de operación. Los métodos de tratamiento de aguas residuales en los cuales la eliminación de contaminantes se realiza a través de reacciones químicas o biológicas son conocidos como unidades de procesamiento. Ambos métodos son agrupados para permitir varios niveles de tratamiento conocidos como: preliminar, primario, secundario y avanzado (o terciario) [37].

Los niveles de tratamiento (Figura 2.7) con los que cuenta la PTAR de la UTM son los siguientes [6]:

- *Tratamiento preliminar:* Consiste en un sistema de rejillas o cribas (figura 2.7a) para separar los sólidos mayores o flotantes y con ello proteger los equipos de los siguientes niveles de tratamiento.
- *Tratamiento primario:* Emplea el proceso físico de asentamiento en un tanque de sedimentación, para eliminar la mayoría de los sólidos suspendidos y sedimentales que se encuentran en el agua residual. Este proceso utiliza un tanque mezclador (figura 2.7b) para homogeneizar el efluente (agua residual) y controlar el flujo de entrada al siguiente tratamiento.
- *Tratamiento secundario:* Emplea el proceso biológico conocido como Sistema de Tratamiento por Lodos Activados (STLA), para remover la mayor parte de la materia orgánica del agua residual. La biodegradación de la materia orgánica se lleva a cabo en un bioreactor (figura 2.7c).
- *Tratamiento terciario:* En donde se utiliza un tanque (figura 2.7d) para dosificar una solución de hipoclorito de sodio para la eliminación de bacterias.

El pH es la expresión de la concentración de los iones  $H^+$  en el agua [37]. Elevadas concentraciones de este ión confieren al agua condiciones de pH ácido mientras que bajas concentraciones del ión significan que el agua tiene condiciones básicas. El pH del agua afecta diversos procesos como puede ser la solubilidad y el estado químico de muchos compuestos y elementos (por ejemplo, de los metales pesados), la actividad de algunas enzimas y la sobrevivencia de numerosas especies.

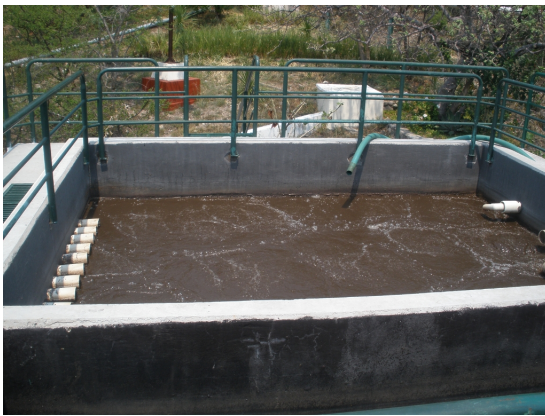
Desde el punto de vista biológico, la mayoría de las especies prefieren valores de pH cercanos a la neutralidad (aproximadamente 7). Sin embargo existen especies adaptadas a vivir en condiciones extremas de pH, ya sea hacia la parte ácida o básica de la escala.



(a) Tratamiento preliminar.



(b) Tratamiento primario.



(c) Tratamiento secundario.



(d) Tratamiento terciario.

**Figura 2.7.** Niveles de tratamiento de la PTAR.

### 2.2.2.1. Sistema de tratamiento por lodos activados

El principio básico de este proceso consiste en que las aguas residuales se pongan en contacto con una población microbiana mixta en forma de suspensión floculenta (o lodo) en un sistema aireado y agitado [37].

El funcionamiento de procesos aeróbicos, tal es el caso de el sistema de lodos activados, depende de la disponibilidad de cantidades suficientes de  $O_2$  en el sistema. La transferencia de  $O_2$  es el proceso mediante el cual el  $O_2$  es transferido de un medio gaseoso a un medio líquido.

Una vez que se alcanza el grado de tratamiento que se desea, el lodo se separa del agua residual por asentamiento, esta etapa es también conocida como sedimentación. El “sobrenadante” de la etapa de separación es el agua residual tratada y libre de lodos. La mayor parte del lodo asentado en la etapa de separación se regresa a la etapa de aireación para mantener la concentración de los lodos en el tanque de aireación al nivel necesario para un tratamiento efectivo.

Las etapas esenciales del proceso de lodos activados son: aireación, separación y reciclaje de lodos. Los sistemas de aireación disponibles para usar en la etapa de aireación, generalmente se pueden dividir en sistemas de aireación por burbujas, sistemas de aireadores y sistemas mecánicos de aireación. El sistema de aireación de la PTAR de la UTM consiste en un sistema de aireadores como se muestra en la figura 2.8 [6].



**Figura 2.8.** Sistema de aireadores.

## 2.3. Descripción de Sensores

### 2.3.1. Electrodo de pH

Los electrodos de pH se usan en una gran variedad de aplicaciones incluyendo el tratamiento de aguas, procesos químicos, instrumentación médica y sistemas de evaluación ambiental [33].

En el presente trabajo de tesis se utilizó un electrodo para medición de pH de la marca *Cole Parmer*, cuyo modelo es *EW-05992-10*. Esta marca maneja dos tipos de electrodos, uno de unión simple y otro de unión doble. En este caso se ha usado el de unión doble, como el que se ilustra en la figura 2.9.



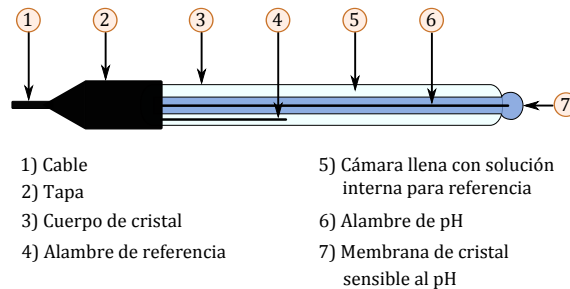
**Figura 2.9.** Electrodo de pH de la marca *Cole Parmer*.

#### 2.3.1.1. Principio de funcionamiento del electrodo de pH

Un electrodo de pH mide la actividad de iones de hidrógeno ( $H^+$ ) y produce un potencial eléctrico o voltaje. La operación del electrodo de pH está basada en el principio de que un potencial eléctrico se genera cuando dos líquidos con diferente pH entran en contacto en los lados opuestos de una membrana delgada de cristal [33].



Los electrodos de pH modernos son una combinación de electrodos compuestos de dos partes principales, un electrodo cristal y un electrodo de referencia como se ilustra en la figura 2.10. El pH es determinado esencialmente a través de la medición de la diferencia de voltaje entre estos dos electrodos.



**Figura 2.10.** Características del electrodo de pH.

En la punta del electrodo se encuentra la membrana pequeña, la cual es un tipo de cristal específico capaz de intercambiar iones. Éste es el elemento que sensa la concentración de hidrógeno de la solución de prueba. El potencial del electrodo de referencia es constante y es producido por el elemento interno del mismo, ya que se encuentra en contacto con la solución de relleno de referencia, la cual tiene un pH constante con un valor de siete.

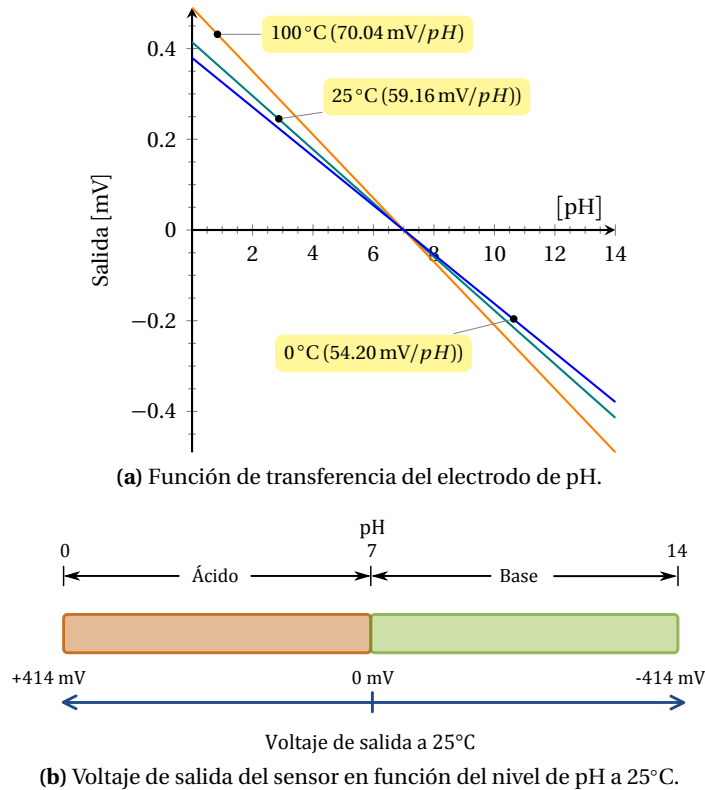
### 2.3.1.2. Características del electrodo de pH

Los electrodos de pH son sensores activos, lo que significa que no requieren de una fuente de excitación o polarización (voltaje o corriente) externa. Este electrodo se clasifica como un sensor bipolar debido a que la salida del electrodo puede oscilar por arriba o por abajo del punto de referencia. El electrodo produce un voltaje de salida, el cual es linealmente proporcional al valor de pH de la solución que está siendo medida. El rango de valores de pH en que opera el electrodo es de 0 a 14 unidades [33].

La impedancia de salida en este electrodo de pH es muy alta, esto se debe a que el bulbo de cristal tiene una resistencia que típicamente se encuentra en el rango de  $10 - 1000 M\Omega$ . Esto significa que el electrodo solamente puede ser monitoreado por un dispositivo de medición con una impedancia de entrada muy alta.

Las funciones de transferencia mostradas en la figura 2.11a y 2.11b indican que cuando el pH de la solución incrementa, el voltaje producido por el electrodo de pH decrementa. Como se observa en la figura 2.11a, la sensibilidad del electrodo de pH varía con respecto a la temperatura.

El electrodo ideal a una temperatura de  $25^{\circ}C$  tiene una sensibilidad de  $59.16 mV/pH$ , lo que significa que la señal de salida del electrodo oscila de  $+414.12 mV$  a  $-414.12 mV$  para el rango de pH entre 0 y 14 respectivamente.



**Figura 2.11.** Funciones de transferencia de entrada y salida del sensor de pH.

### 2.3.2. Electrodo de OD

Principalmente existen dos tipos de electrodos de OD: electrodos sin membrana y electrodos con membrana permeable (*Principio de Clark*) [21].

Actualmente los electrodos con membrana que funcionan con el principio de Clark son los más usados globalmente, los cuales permiten mediciones de  $O_2$  en gases y soluciones, otra ventaja es que el electrodo y la solución no se contaminan mutuamente.

En la UTM se cuenta con un electrodo de OD marca *Mettler-Toledo*, cuyo modelo es *InPro 6800* como el que se muestra en la figura 2.12. Este electrodo está destinado a mediciones en línea (*InLine Measurement*) de la presión parcial de  $O_2$  en líquidos y gases .

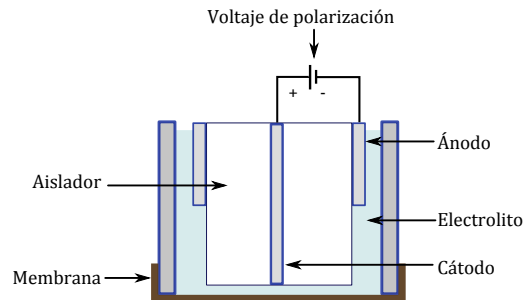


**Figura 2.12.** Electrodo de OD de la marca *Mettler Toledo*.

Algunas de las aplicaciones de estos electrodos de OD son: fermentación, propagación de levadura, aireación de mosto, acondicionamiento de agua de manantial y almacenamiento y procesamiento de jugo de frutas.

### 2.3.2.1. Principio de funcionamiento del electrodo de OD

El electrodo *InPro 6800* está basado en el principio de medición polarográfica (*Clark Polarographic Sensor*), el cual consiste de un electrodo de operación (*cátodo*), un electrodo de contra-referencia (*ánodo*) y una membrana permeable al  $O_2$ , dicha membrana separa los electrodos de la muestra. En la figura 2.13 se ilustran los componentes del electrodo de OD [21].



**Figura 2.13.** Características del electrodo de OD.

El electrodo de OD es un sensor pasivo, lo que quiere decir que requiere una fuente de excitación, en este caso se le llama fuente de polarización. Esta fuente de polarización debe proporcionar un voltaje constante al cátodo para reducir el  $O_2$ .

Las moléculas de  $O_2$  que migran a través de la membrana permeable son reducidas en el cátodo. Al mismo tiempo se origina una oxidación en el ánodo, por lo que el metal (*Plata*) oxidado del ánodo es liberado en forma de iones de plata dentro de un electrolito. El electrolito provoca que se cierre el circuito eléctrico entre el ánodo y el cátodo (conductividad iónica).

La corriente producida por las reacciones descritas anteriormente es proporcional a la presión parcial del  $O_2$  en la muestra.

### 2.3.2.2. Características del electrodo de OD

Para un funcionamiento correcto, el electrodo *InPro 6800* requiere de una polarización completa, lo cual se logra cuando se mantiene conectado el electrodo al voltaje de polarización constantemente por 6 horas. Si el sensor es desconectado por más de 5 minutos, se debe volver a realizar el proceso de polarización [21].

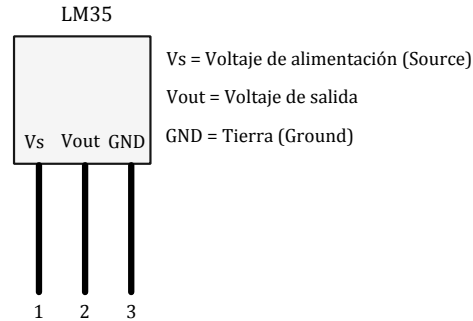
En aplicaciones estándares, como en este caso, el voltaje de polarización debe ser de 675mV y debe permanecer constante. La señal de salida del sensor de OD oscila de 50nA a 110nA al aire ambiente.

El rango de temperatura para una correcta operación es de 0 – 80°C. El rango de operación del electrodo es de 0ppm a 60ppm (ppm, *partes por millón*). La precisión que proporciona es menor

o igual a  $1\% + 6\text{ppm}$  para lecturas en líquidos. El material con el que está construido el cátodo es platino (*Pt*) y el ánodo es de plata (*Ag*).

### 2.3.3. Sensor de temperatura LM35

El sensor de circuito integrado *LM35* es un sensor de temperatura de precisión, el cual es un sensor pasivo cuyo voltaje de salida es linealmente proporcional a la temperatura en grados centígrados ( $^{\circ}\text{C}$ ). En la figura 2.14 se ilustra el encapsulado y la distribución de terminales del sensor [26].



**Figura 2.14.** Configuración del sensor LM35.

#### 2.3.3.1. Funcionamiento y características del sensor LM35

El voltaje de operación del sensor *LM35* es de 4 a 30V, tiene un rango de operación de  $-55$  a  $150^{\circ}\text{C}$  y tiene una sensibilidad de  $10\text{mV}/^{\circ}\text{C}$ . El *LM35* no requiere de una calibración externa y proporciona una precisión de  $\pm 0.25\text{mV}$  [26].

La impedancia de salida del *LM35* es baja, además permite el uso de una fuente individual de excitación, lo que facilita el diseño del circuito electrónico de interfaz.

## 2.4. Protocolos de Comunicaciones

Un protocolo de comunicaciones es un conjunto de reglas y convenciones que deben seguir dos equipos cualesquiera (computadoras o dispositivos) para intercambiar información a través de una red. Un protocolo es una regla o estándar que controla o permite la comunicación en su forma más simple. Puede ser definido como las reglas que dominan la sintaxis, semántica y sincronización de la comunicación. Puede ser implementado por *hardware*, *software* o una combinación de ambos [8, 16, 41].

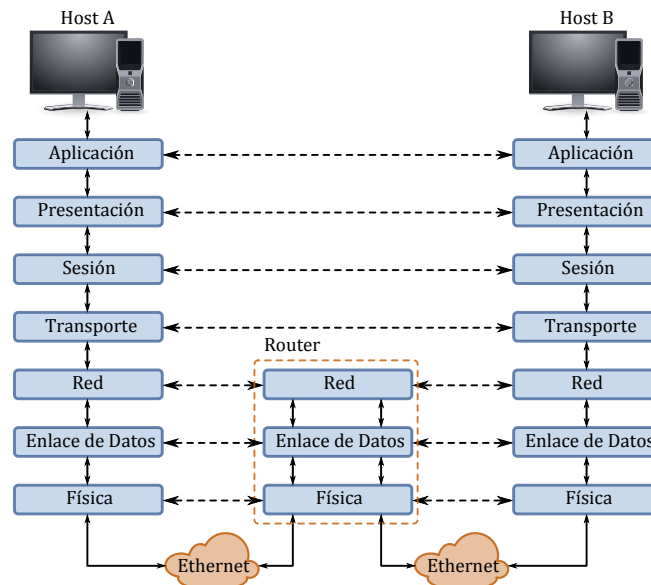
Cualquier red de comunicaciones de datos requiere un sistema para organizar el flujo de datos. Para ordenar los protocolos de comunicaciones, la Organización Internacional de Normalización (ISO, *International Organization for Standardization*), propuso en el año de 1984 el modelo de referencia para la Interconexión de Sistemas Abiertos (OSI, *Open System Interconnection*) [8, 16, 41].

El modelo OSI describe las reglas que los equipos de comunicaciones deben seguir para que el intercambio de datos sea posible dentro de una infraestructura que está compuesta por una gran variedad de productos de diferentes fabricantes [8].

### 2.4.1. El modelo de referencia OSI

Las recomendaciones del modelo OSI no son normas concretas, sino reglas genéricas para separar en siete niveles, o capas, el proceso de comunicación entre dos sistemas. Las capas del modelo OSI son (figura 2.15): física, enlace de datos, red, transporte, sesión, presentación y aplicación [8, 16, 41].

El modelo OSI permite que cada nivel se encargue de tareas específicas y utilice los servicios de los niveles inferiores sin preocuparse de cómo funcionan. Por lo que resulta fácil realizar cambios en una parte sin tener que alterar el resto de las especificaciones. La comunicación entre dos sistemas solo puede realizarse entre capas del mismo nivel, en la figura 2.15 se muestra el flujo de datos entre capas de dos sistemas que se comunican empleando el modelo de referencia OSI.



**Figura 2.15.** Arquitectura de protocolos del modelo de referencia OSI.

#### 2.4.1.1. Capa física

En la capa física se especifican las características mecánicas y eléctricas del sistema físico de transporte (par trenzado, coaxial, fibra óptica, etc.) y de las interfaces que permiten la conexión física de los equipos a dichos sistemas de transporte [8, 16, 41].

En este nivel se definen las topologías aceptadas, el modo de emisión o forma de la señal y el soporte de transmisión (banda base o señal portadora).

Las características de las señales y conexiones que se definen son: características físicas de los conectores, características eléctricas de las señales (voltaje e impedancia), características eléctricas del *hardware*, implementación de las señales y la codificación.

El objetivo primordial de la capa física consiste en transmitir bits por un canal de comunicación, de manera que todo lo que envíe el transmisor llegue al receptor sin sufrir alteraciones.

Globalmente las interfaces de capa física para comunicaciones de datos incluyen interfaces serie como *EIA/TIA-232* y *EIA/TIA-485*, interfaces paralelas, y las especificaciones físicas para sistemas LAN tales como *Ethernet* (figura 2.15) y *Token Ring*. Los sistemas inalámbricos cuentan con interfaces aéreas que definen la transmisión de datos usando radio, microondas o señales infrarrojas.

#### 2.4.1.2. Capa de enlace de datos

La capa de enlace de datos define las reglas para enviar y recibir información a través de una conexión física entre dos sistemas. Establece la forma de agrupar los datos en paquetes o tramas de longitud adecuada y añade los mecanismos necesarios para controlar la transmisión de información y detectar y corregir los errores que puedan aparecer [8, 16, 41].

El acceso al medio físico puede ser de tres maneras:

- *Controlado por un equipo único*: Este sistema emplea la arquitectura Maestro/Esclavo (por ejemplo el protocolo Modbus).
- *Condicionado por un derecho*: En este sistema el derecho de acceso es proporcionado por un testigo, el equipo que tiene dicho testigo puede emitir un mensaje y transmitir el testigo al siguiente equipo (por ejemplo el protocolo Profibus).
- *Aleatorio o descentralizado*: El equipo que quiere emitir verifica que la línea de transmisión esté libre. Si dos equipos emiten de forma simultánea, se origina una colisión y ambos mensajes se destruyen. Tal es el caso de los protocolos del tipo CSMA (*Carrier Sense Multiple Access*) empleados principalmente en redes Ethernet.

#### 2.4.1.3. Capa de red

Esta capa provee servicios de interconexión de redes para la distribución de datos a través de múltiples redes. La capa de red se ocupa del direccionamiento a través de sistemas mediante técnicas de encaminamiento (ruteo, *Routing*) y el control de flujo. El esquema de direccionamiento de redes asigna una dirección única a cada red y a cada equipo conectado [8, 16, 41].

El protocolo IP dentro de la suite de protocolos TCP/IP, es un ejemplo de protocolo de capa de red.

#### 2.4.1.4. Capa de transporte

La capa de transporte provee servicios de comunicaciones entre dos sistemas. Tiene la función de garantizar un enlace fiable asegurando que los datos sean entregados. Ambos sistemas establecen una conexión y mantienen un diálogo para llevar el control de la entrega de paquetes a través de la red [8, 16, 41].

Un protocolo de la capa de transporte divide la información en paquetes manejables por el sistema de transmisión y controla la gestión de paquetes (orden de envío y recepción, formatos de transmisión, peticiones de reenvío en caso de error, etc.).

#### 2.4.1.5. Capa de sesión

La capa de sesión establece, administra y termina las sesiones de comunicación entre equipos. Las sesiones de comunicación consisten de peticiones y respuestas de servicio, las cuales se llevan a cabo entre aplicaciones de diferentes equipos en la red [8, 16, 41].

#### 2.4.1.6. Capa de presentación

La capa de presentación provee una variedad de funciones de codificación y conversión que son aplicados a los datos de la capa de aplicación. Estas funciones aseguran que la información enviada desde la capa de aplicación de un sistema sea entendible por la capa de aplicación de otro sistema [8, 16, 41].

#### 2.4.1.7. Capa de aplicación

La capa de aplicación es la más cercana al usuario final, por lo tanto, el usuario final y la capa de aplicación interactúan directamente con el *software* de aplicación [8, 16, 41].

Esta capa es un espacio de libre utilización para fabricantes y usuarios. Presta servicios al usuario, los cuales comprenden la interacción directa con los procesos de aplicación. Algunos ejemplos de implementaciones en la capa de aplicación incluyen Telnet, el protocolo de transferencia de ficheros (FTP, *File Transfer Protocol*), bases de datos y el protocolo de transferencia de correo simple (SMTP, *Simple Mail Transfer Protocol*).

### 2.4.2. Interfaces serie

En las comunicaciones del entorno industrial, las conexiones físicas se realizan mediante interfaces serie, las cuales son normalizadas por la Asociación de Industrias Electrónicas (EIA, *Electronic Industrial Association*) y la Asociación de la Industria de Telecomunicaciones (TIA, *Telecommunications Industry Association*) [4, 12, 28].

Los estándares de interfaces serie más conocidos son: *EIA/TIA-232* o *RS-232*, y *EIA/TIA-485* o *RS-485*. Los cuales forman el elemento principal para la transferencia de información digital entre una RTU y un módem (modulador/demodulador). El módem es aquel que convierte las señales digitales (TTL), generadas por una computadora o un sistema basado en MCU, en una forma

analógica adecuada para la transmisión a distancias grandes a través de cables o un sistema de radio.

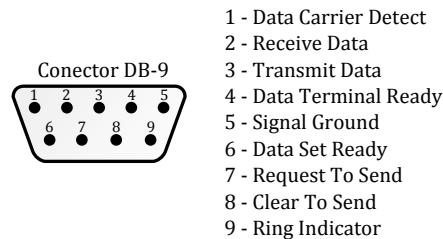
En los estándares RS-232 y RS-485 se definen las características eléctricas y los detalles mecánicos que permiten a equipos de comunicaciones, de diferentes fabricantes, conectarse y funcionar eficientemente.

#### 2.4.2.1. Comunicación de datos serie en el estándar RS-232

El estándar de interfaz RS-232 define la interfaz entre un Equipo Terminal de Datos (DTE, *Data Terminal Equipment*) y un Equipo de Comunicaciones de Datos (DCE, *Data Communication Equipment*) empleando el intercambio de datos binarios serie, en donde el DCE es el dispositivo que emite datos y el DTE es que recibe dichos datos [4, 12, 28].

El estándar RS-232 consiste principalmente de tres partes, las cuales definen las características de las señales eléctricas, las características de la interfaz mecánica y la descripción funcional de los circuitos de intercambio.

Los conectores comúnmente asociados al estándar RS-232 son el DB-25 (tipo D de 25 pines) y el DB-9 (tipo D de 9 pines) [12], de los cuales el conector DB-9 es el más usado globalmente en computadoras y dispositivos en el ámbito industrial. En la figura 2.16 se muestra la asignación de terminales y las señales del conector DB-9 del estándar RS-232.



**Figura 2.16.** Asignación de terminales y señales del conector DB-9 para el estándar RS-232.

##### 2.4.2.1.1. Características de las señales eléctricas

El estándar de interfaz RS-232 está diseñado para la comunicación de dispositivos DTE y DCE, en donde un dispositivo DTE es aquel que transmite los datos y un dispositivo DCE es aquel que los recibe [4, 12, 28].

Para un receptor RS-232, las especificaciones de los niveles de voltaje se describen en la tabla 2.1.

De manera similar al receptor, el transmisor RS-232 debe producir niveles de voltaje ligeramente más elevados para contrarrestar las caídas de voltaje a lo largo de la línea de transmisión (tabla 2.2).



**Tabla 2.1.** Niveles de voltaje del receptor del estándar RS-232.

Estados lógicos	Niveles de voltaje
Estado Lógico 0	+3V a +25V
Estado Lógico 1	-3V a -25V
Estado Indefinido	-3V a +3V

**Tabla 2.2.** Niveles de voltaje del transmisor del estándar RS-232.

Estados lógicos	Niveles de voltaje
Estado Lógico 0	+5V a +25V
Estado Lógico 1	-5V a -25V

#### 2.4.2.1.2. Descripción funcional de los circuitos de intercambio

Las funciones de los circuitos de intercambio definidos en el estándar RS-232, son las conexiones o señales eléctricas involucradas en la comunicación, las cuales se describen en la tabla 2.3 [4, 12, 28].

**Tabla 2.3.** Señales del estándar RS-232.

Señal	Nombre completo	Descripción
Common	Signal Ground	Es la línea común de referencia.
TXD	Transmit Data	Es la portadora de datos serie de un DTE a un DCE.
RXD	Received Data	Es la portadora de datos serie de un DCE a un DTE.
RTS	Request to Send	Esta línea habilita la transmisión.
CTS	Clear to Send	Es la señal de inicialización para enviar.
DSR	Data Set Ready	Indica que el DCE está listo para recibir.
DCD	Data Carrier Detect	Detecta la señal de la línea de entrada.
DTR	Data Terminal Ready	Indica que el DTE está listo para enviar.

#### 2.4.2.2. Comunicación de datos serie en el estándar RS-485

El estándar RS-485 es una expansión del estándar RS-422, es un método de comunicación usado en aplicaciones industriales. El estándar RS-485 define una interfaz de comunicación de datos balanceado o diferencial, haciendo uso de dos cables separados para cada señal. Esto permite velocidades de transferencia de datos muy elevadas y minimiza los problemas de las variaciones de potencial en la tierra (ground, GND), a diferencia del estándar RS-232, la tierra no es usada como voltaje de referencia [4, 12, 28].

El estándar RS-485 permite una conexión de red multipuntos en dos alambres y una comunicación de datos serie confiable con las siguientes características: una distancia máxima de alcance de 1200m, una tasa máxima de transferencia de datos de 10 Mbps (megabits por segundo), permite la conexión de 32 transmisores de línea (line driver) y 32 receptores de línea (line receiver) como máximo en la misma línea.

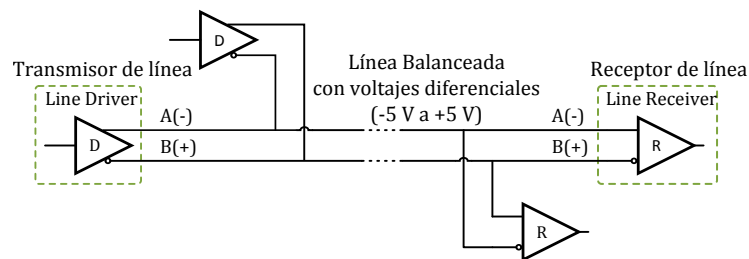
Los rangos de voltaje en la línea son: -1.5V a -6.0V para el estado lógico "1", y +1.5V a +6.0V para el estado lógico "0". Los puntos de referencia usados para estos voltajes son las terminales A y B

de los transmisores/receptores de línea mostrados en la figura 2.17. El transmisor de línea, para una interfaz RS-485, produce un voltaje diferencial de  $\pm 5V$  en los dos alambres.

La mejora más importante del estándar RS-485 con respecto al estándar RS-232 es que el modo de operación del transmisor de línea consiste de 3 estados: estado lógico “0”, estado lógico “1” y “alta impedancia”. Dichos estados se definen de la siguiente manera:

- Cuando la terminal A del driver es negativa, con respecto a la terminal B, la línea está en el estado lógico 1.
- Cuando la terminal A del driver es positiva, con respecto a la terminal B, la línea está en el estado lógico 0.
- El estado de alta impedancia, también conocido como estado deshabilitado, puede ser iniciado por una señal en una terminal de control en el circuito integrado del transmisor de línea.

Para sistemas *full-duplex* se requiere de cinco alambres, en cambio para sistemas *half-duplex*, solo se requieren tres alambres. En la figura 2.17 se ilustra un ejemplo de configuración del estándar de interfaz RS-485 multipunto [12].



**Figura 2.17.** Interfaz multipunto del estándar RS-485.

Los receptores de línea, solamente son sensitivos a la diferencia entre dos señales de entrada, por lo que, las señales comunes de ruido se presentan en ambos alambres y solamente tendrán un pequeño efecto en la operación del receptor.

El estándar de interfaz RS-485 es útil en sistemas que requieren la conexión de múltiples dispositivos separados por distancias grandes haciendo uso del mismo par de líneas. En dichos sistemas se requiere un *software* de control para establecer qué dispositivo de la red puede estar activo. En la mayoría de los casos una MTU controla qué transmisor/receptor estará activo en algún momento.

### 2.4.3. Modbus

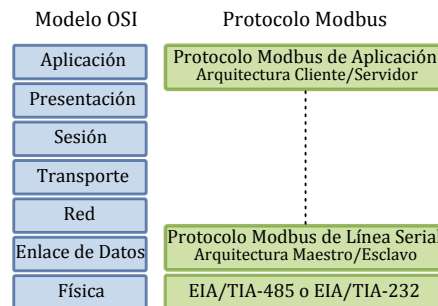
Modbus es un protocolo de comunicaciones clasificado como bus de campo. Un bus de campo es un bus de transferencia de información en serie utilizado en la industria. Los buses de campo están orientados a procesos discretos (automóviles, electrodomésticos, etc.) y procesos continuos (petroquímicas, electricidad, agua, etc.) [12, 28, 46].

Según la definición de la Comisión Electrotécnica Internacional (IEC, *International Electrotechnical Commission*) y la Sociedad de Instrumentación de América (ISA, *Instrument Society of America*), un bus de campo es una conexión serie digital que permite la transferencia de datos entre elementos primarios de automatización (instrumentos de campo), que realizan funciones de medida y control, y elementos de automatización y control de más alto nivel.

Un bus de campo, al igual que una red LAN, satisface los dos primeros niveles del modelo OSI y el último (figura 2.18). No obstante presenta mensajes más cortos (órdenes, eventos, medidas, etc.), con tiempos de respuesta entre 5ms y 100ms y alta seguridad en la comunicación, sobre distancias entre 200m y 2km, a velocidades que suelen ser inferiores a 1Mbps.

Normalmente en el medio físico de un bus de campo de bajo costo, consiste de un par de hilos con una interfaz serie RS-485, aunque se encuentran aplicaciones con coaxial, fibra óptica, radio e infrarrojos.

El estándar Modbus define un protocolo de mensajes de la capa de aplicación, posicionado en el nivel siete del modelo OSI, que provee comunicaciones con arquitectura Cliente/Servidor entre dispositivos conectados en diferentes tipos de buses o redes [22, 23, 24]. El estándar Modbus define un protocolo específico para interfaz serie (Modbus sobre una línea serie) para el intercambio de peticiones Modbus haciendo uso de la arquitectura Maestro/Esclavo. En la figura 2.18 se ilustra el protocolo Modbus con respecto al modelo de referencia OSI [24].



**Figura 2.18.** Protocolo Modbus relacionado con el Modelo OSI.

En la actualidad, el protocolo Modbus, es implementado sobre TCP/IP en redes Ethernet y transmisión serie asíncrona en interfaces físicas como RS-485 (Sección 2.4.2.2).

#### 2.4.3.1. Capa de enlace de datos del protocolo Modbus

El protocolo Modbus sobre una línea serie es un protocolo con arquitectura Maestro/Esclavo posicionado en la capa dos del modelo de referencia OSI (figura 2.18) [22, 24, 28].

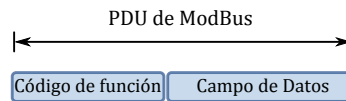
Los principios generales de la comunicación Modbus son: solo un nodo Maestro debe estar conectado al bus o red y un máximo de 247 nodos Esclavos, el nodo Maestro iniciará la comunicación, los nodos Esclavos no transmitirán datos sin haber recibido una petición del nodo Maestro, los nodos Esclavos no se comunicarán entre ellos y el nodo Maestro mantendrá comunicación con un solo nodo Esclavo a la vez.

El nodo Maestro puede emitir peticiones Modbus de dos modos:

- *Unicast*: Se usa cuando el Maestro se dirige a un nodo Esclavo en específico. El nodo Esclavo, después de recibir e interpretar la petición, retornará un mensaje o respuesta al nodo Maestro.
- *Broadcast*: En este modo el nodo Maestro manda una petición a todos los nodos Esclavo, en este modo no hay una respuesta retornada por los nodos Esclavo.

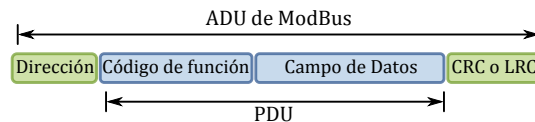
El espacio para direccionamiento del protocolo Modbus comprende 256 diferentes direcciones, las cuales están distribuidas de la siguiente manera: la dirección 0 está destinada para peticiones broadcast, por lo que todos los nodos Esclavo deben reconocer esta dirección; el nodo Maestro no tiene una dirección en particular; los nodos Esclavos tienen una dirección única en un bus serie, las direcciones de un nodo Esclavo pueden ser desde 1 a 247; las direcciones desde la 248 a la 255 están reservadas.

El protocolo Modbus de aplicación define una Unidad de Datos de Protocolo (PDU, *Protocol Data Unit*) [23], que se muestra en la figura 2.19, la cual es independiente de las capas inferiores de comunicación (modelo de referencia OSI).



**Figura 2.19.** Unidad de datos de protocolo de MODBUS.

La implementación del protocolo Modbus sobre una línea serie requiere campos adicionales en la PDU (figura 2.20) formando así una Unidad de Datos de Aplicación (ADU, *Application Data Unit*) [23].



**Figura 2.20.** Unidad de datos de aplicación de MODBUS.

Los campos de la ADU son:

- *Dirección*: En el cual el nodo Maestro indica a que nodo Esclavo se dirige y el nodo Esclavo coloca su propia dirección para indicar de qué nodo Esclavo proviene la respuesta.
- *Código de función*: Para indicar al nodo Esclavo que acción debe realizar.
- *Datos*: En donde el nodo Maestro establece los parámetros de la acción y el nodo Esclavo coloca la información o respuesta adecuada a la acción solicitada.
- *Verificación de error*: Es el resultado de un cálculo de verificación de redundancia realizado sobre el contenido del mensaje o ADU (todos los campos anteriores), para este cálculo se usan dos tipos de métodos según sea el modo de transmisión: CRC (*Cyclical Redundancy Check*) y LRC (*Longitudinal Redundancy Check*).

En Modbus la transmisión de ADUs se realizan de dos maneras:

- *RTU*: En donde la codificación es binaria, es decir, cada byte (8-bit) de información es enviado como dos caracteres de 4 bits hexadecimales. Cada carácter consta de un bit de inicio, 8 bits de codificación de datos, un bit de paridad (opcional) y uno o dos bits de paro, dando un total de 10 a 12 bits por carácter. Para la comprobación de errores en las tramas que usan RTU se emplea una secuencia de código CRC.
- *ASCII*: En el cual la codificación es hexadecimal, es decir, cada byte (8-bit) de información es enviado como dos caracteres ASCII. Por ejemplo, el byte de 0x5B es codificado como caracteres: 0x35 y 0x42 en código ASCII, en donde 0x35 = "5" y 0x42 = "B". Cada carácter consta de un bit de inicio, 7 bits de codificación de datos, un bit de paridad y uno o dos bits de paro, dando un total de 9 a 11 bits por carácter. Para la comprobación de errores en las ADUs se emplea una secuencia de código LRC.

#### 2.4.3.2. Descripción de los códigos de función del protocolo Modbus

Existen tres categorías de códigos de función Modbus: códigos públicos, códigos definidos por el usuario y los códigos reservados [22, 23]. En la tabla 2.4 se muestran los códigos de función públicos que se emplean con mayor frecuencia en equipos y dispositivos conectados en una red serie.

**Tabla 2.4.** Códigos de función MODBUS implementados en redes serie.

Códigos de Función (hex)	Descripción
0x01	Lectura de Salidas Digitales o Relevadores.
0x02	Lectura de Entradas Digitales.
0x04	Lectura de Registros de Entrada.
0x05	Escritura de una Salida Digital o Relevador Individual.
0x06	Escritura de un Registro Individual.

##### 2.4.3.2.1. 0x01 Lectura de Salidas Digitales o Relevadores

Este código de función es usado para leer los estados de 1 a 2000 salidas digitales continuas en un dispositivo remoto [22, 23].

La PDU de petición (tabla 2.5) del nodo Maestro especifica los siguientes campos: la dirección de la salida digital de inicio, es decir, la dirección de la primera salida digital especificada; y, la cantidad de salidas digitales. En la PDU las direcciones de las salidas digitales empiezan desde la dirección cero.

**Tabla 2.5.** PDU de Petición de la función 0x01.

Código de Función	Dirección de Inicio	Cantidad de Salidas
1 Byte	2 Bytes	2 Bytes
<b>0x01</b>	0x0000 hasta 0xFFFF	1 hasta 2000(0x7D0)

En la PDU de respuesta (tabla 2.6) del nodo Esclavo, cada estado de una salida digital es representado con un bit en el campo de datos. El estado lógico “1” indica que la salida digital está habilitada y el estado lógico “0” indica que está deshabilitada. El bit menos significativo indica el estado correspondiente a la primera salida digital y así sucesivamente hasta el bit más significativo. El campo de cantidad de bytes especifica la cantidad de bytes en el campo de datos.

**Tabla 2.6.** PDU de Respuesta de la función 0x01.

Código de Función	Cantidad de bytes	Array de Estado de Salidas
1 Byte	1 Bytes	n*Bytes
<b>0x01</b>	N bytes	n=N ó n=N+1

#### 2.4.3.2.2. 0x02 Lectura de Entradas Digitales

Este código de función es usado para leer los estados de 1 a 2000 entradas digitales continuas en un dispositivo remoto [22, 23].

La PDU de petición (tabla 2.7) del nodo Maestro especifica los siguientes campos: la dirección de la entrada digital de inicio, es decir, la dirección de la primera entrada digital especificada; y la cantidad de entradas digitales. En la PDU las direcciones de las entradas digitales empiezan desde la dirección cero.

**Tabla 2.7.** PDU de Petición de la función 0x02.

Código de Función	Dirección de Inicio	Cantidad de Entradas
1 Byte	2 Bytes	2 Bytes
<b>0x02</b>	0x0000 hasta 0xFFFF	1 hasta 2000(0x7D0)

En la PDU de respuesta (tabla 2.8) del nodo Esclavo, cada estado de una entrada digital es representado con un bit en el campo de datos. El estado lógico “1” indica que la entrada digital está habilitada y el estado lógico “0” indica que está deshabilitada. El bit menos significativo indica el estado correspondiente a la primera entrada digital y así sucesivamente hasta el bit más significativo. El campo de cantidad de bytes especifica la cantidad de bytes en el campo de datos.

**Tabla 2.8.** PDU de Respuesta de la función 0x02.

Código de Función	Cantidad de bytes	Array de Estado de Entradas
1 Byte	1 Bytes	n Bytes
<b>0x02</b>	N bytes	n=N ó n=N+1

#### 2.4.3.2.3. 0x04 Lectura de Registros de Entrada

Este código de función es usado para leer los estados de 1 a 125 registros de entrada continuos en un dispositivo remoto [22, 23].

La PDU de petición (tabla 2.9) del nodo Maestro especifica los siguientes campos: la dirección del registro de entrada de inicio, es decir, la dirección del primer registro de entrada especificado; y, la cantidad de registros de entradas. En la PDU las direcciones de los registros de entrada empiezan desde la dirección cero.

**Tabla 2.9.** PDU de Petición de la función 0x04.

Código de Función	Dirección de Inicio	Cantidad de Registros de Entrada
1 Byte	2 Bytes	2 Bytes
<b>0x04</b>	0x0000 hasta 0xFFFF	0x0001 hasta 0x007D

En la PDU de respuesta (tabla 2.10) del nodo Esclavo, los datos de cada registro de entrada constan de 2 bytes y su contenido binario está alineado a la derecha. Como cada registro de entrada es de 16 bits separados en 2 bytes, por cada registro, el primer byte contiene los bits de la parte alta y el segundo contiene los bits de la parte baja. El campo de cantidad de bytes especifica la cantidad de bytes en el campo de datos.

**Tabla 2.10.** PDU de Respuesta de la función 0x04.

Código de Función	Cantidad de bytes	Array de Estado de Entradas
1 Byte	1 Bytes	Nx2 Bytes
<b>0x04</b>	2xN	

#### 2.4.3.2.4. 0x05 Escritura de una Salida Digital o Relevador Individual

Este código de función es usado para escribir o establecer el estado de una salida digital individual en un dispositivo remoto [22, 23].

En la PDU de petición (tabla 2.11) del nodo Maestro se especifica la dirección de la salida digital individual y el estado (habilitado o deshabilitado) se especifica a través de un valor constante en el campo de datos de la PDU, el valor hexadecimal “FF 00” especifica que la salida digital sea habilitada y el valor hexadecimal “00 00” especifica que la salida digital sea deshabilitada.

**Tabla 2.11.** PDU de la función 0x05.

Código de Función	Dirección de la Salida	Valor de Salida
1 Byte	2 Bytes	2 Bytes
<b>0x05</b>	0x0000 hasta 0xFFFF	0x0000 ó 0xFF00

La PDU de respuesta (tabla 2.11) del nodo Esclavo es un eco de la PDU de petición, la cual usualmente es retornada después de haber establecido el estado de la salida individual.

#### 2.4.3.2.5. 0x06 Escritura de un Registro Individual

Este código de función es usado para escribir o establecer un registro individual en un dispositivo remoto [22, 23].

En la PDU de petición (tabla 2.12) del nodo Maestro, se especifica la dirección del registro individual a ser escrito. En la PDU las direcciones de los registros mantenidos empiezan desde la dirección cero.

**Tabla 2.12.** PDU de la función 0x06.

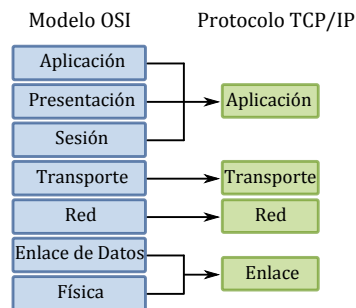
Código de Función	Dirección de la Salida	Valor de Salida
1 Byte	2 Bytes	2 Bytes
<b>0x06</b>	0x0000 hasta 0xFFFF	0x0000 hasta 0xFFFF

La PDU de respuesta (tabla 2.12), del nodo Esclavo es un eco de la PDU de petición, la cual usualmente es retornada después de haber escrito o establecido el registro individual.

#### 2.4.4. Suite de protocolos TCP/IP

El modelo de la Internet utiliza la suite de protocolos TCP/IP. IP es un protocolo de capa de red que proporciona mecanismos de interconexión entre redes de área local y TCP es un protocolo de capa de transporte que proporciona mecanismos de control de flujo y errores entre los extremos de la comunicación [5, 46, 48].

El esquema de capas TCP/IP organiza los grupos funcionales de protocolos en 4 capas: enlace, red, transporte y aplicación. El modelo de referencia OSI puede ser relacionado con la suite de protocolos TCP/IP (Figura 2.21).



**Figura 2.21.** Modelo OSI relacionado con el modelo TCP/IP.

##### 2.4.4.1. Capa de enlace

La capa de enlace o capa de interfaz de red es responsable de gestionar el medio físico, tal como LAN (Ethernet) o WAN. Todos los equipos conectados a Internet implementan este nivel [8, 16, 46].

Algunos ejemplos relevantes de protocolos en esta capa son los siguientes: ARP (*Address Resolution Protocol*), HDLC (*High-level Data Link Control*) y PPP (*Point-to-Point Protocol*).



#### 2.4.4.2. Capa de internet

La capa de Internet es responsable de la comunicación máquina a máquina, en donde una máquina puede ser un anfitrión<sup>5</sup> (*host*) o una puerta de enlace (*gateway*) [8, 16, 46].

El paquete (de información) recibido de la capa de transporte, para su transmisión es encapsulado en un datagrama de Protocolo de Internet. La capa de Internet determina si enviar el datagrama directamente o a través de una puerta de enlace, posteriormente envía el datagrama a la capa de enlace para la transmisión.

La capa de Internet también se encarga de los datagramas de entrada. Se verifica la validez de cada datagrama y se determina si el datagrama debería ser procesado de manera local o ser enviado.

Algunos protocolos de la capa de Internet son: IP e ICMP (*Internet Control Message Protocol*).

#### 2.4.4.3. Capa de transporte

Esta capa tiene la responsabilidad de proveer la comunicación entre las aplicaciones de *software* del *host* fuente y el *host* destino y regular el flujo de información entre ellos [8, 16, 46].

La capa de transporte también provee mecanismos confiables de transporte a la capa de aplicación, asegurándose que los datos son pasados a la capa de aplicación en la secuencia correcta, sin duplicidad y sin errores.

Algunos protocolos de la capa de transporte son: TCP y UDP (*User Datagram Protocol*).

#### 2.4.4.4. Capa de aplicación

En este nivel los usuarios solicitan procesos de aplicación que acceden a la red. La aplicación interactúa con los protocolos de la capa de transporte para enviar o recibir los datos emitidos por los procesos [8, 16, 46].

Algunos de los protocolos comunes de la capa de aplicación son: clientes y servidores de WWW, TELNET, FTP y SMTP.

## 2.5. Herramientas de Desarrollo de Aplicaciones Web y de Escritorio

Los usuarios finales, de un sistema de monitoreo y control, requieren de GUIs para interactuar con los dispositivos de una planta industrial. Generalmente, dichas interfaces gráficas consisten en aplicaciones Web y de escritorio.

Las aplicaciones Web son aplicaciones distribuidas que emplean la arquitectura Cliente/Servidor. Dicha arquitectura se caracteriza por tener los siguientes elementos [5]:

- Cliente: Realiza peticiones de servicio y normalmente es el que inician la comunicación.
- Servidores: Estos proveen servicios y esperan a recibir peticiones. Una vez que han recibido una petición, la resuelven y retornan una respuesta.

---

<sup>5</sup>Computadora.

Las aplicaciones Web se basan en el hecho de contar con toda la capacidad de procesamiento en un servidor Web que se accede desde un navegador. Las aplicaciones Web se desarrollan empleando varios lenguajes de programación, entre ellos destacan: PHP, Ruby, Python, JavaScript, SQL y HTML.

Actualmente para las aplicaciones Web existen herramientas que facilitan su desarrollo y mantenimiento. Para PHP predominan los *frameworks* con arquitectura Modelo Vista Controlador (MVC, *Model View Controller*), éstos facilitan la creación de aplicaciones Web gracias a que dividen el desarrollo en tareas sencillas e intuitivas para el desarrollador [43, 45]. Entre los frameworks más destacados se encuentran: Yii, Symfony, CodeIgniter, CakePHP y Zend.

Las interfaces de comunicación entre dispositivos (RTU) y computadoras de un sistema SCADA, usualmente requieren de aplicaciones de escritorio, las cuales se desarrollan empleando lenguajes de programación como: Java, C/C++ y Visual Basic.

En un sistema SCADA las aplicaciones Web y de escritorio requieren de bases de datos para almacenar información útil para el usuario, por lo que se hace uso de bases de datos relacionales, los DBMS más usados hoy en día son: MySQL, PostgreSQL, Oracle, Microsoft SQL Server, entre otros.

### 2.5.1. Java

Java es un lenguaje de programación de propósito general, concurrente, basado en clases, orientado a objetos e independiente de la plataforma (Windows o Unix). Java fue desarrollado por Sun Microsystems en la década de los 90s, gran parte de su sintaxis está basada en los lenguajes de programación C y C++, aunque su modelo de objetos es más simple y elimina las herramientas de bajo nivel [10].

El concepto orientado a objetos se refiere a un método de programación y al diseño del lenguaje. Es diseñar el *software* de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Por lo tanto, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el comportamiento (código) y el estado (datos).

### 2.5.2. PHP

PHP es un lenguaje interpretado por parte del servidor, diseñado originalmente para la creación de páginas Web dinámicas, el cual se caracteriza por su potencia, versatilidad, robustez y modularidad [43, 45].

Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor Web a través de un intérprete antes de transferir al cliente un resultado en forma de código HTML puro.

PHP es un lenguaje multiplataforma (Windows y Unix). En la actualidad PHP permite realizar una gran variedad de tareas en el entorno Web, por ejemplo: sistemas de correo electrónico vía Web, funciones de gestión y administración de bases de datos, funciones de gestión de directorios y

ficheros, funciones de tratamiento de imágenes y librerías de funciones gráficas y funciones para la generación de documentos en formato pdf.

### 2.5.3. MySQL

MySQL es un DBMS rápido, robusto y fácil de usar. Permite la administración de datos en un entorno de red, especialmente en las arquitecturas Cliente/Servidor. Cuenta con muchas herramientas y es compatible con varios lenguajes de programación (por ejemplo PHP y Java), también es compatible con el servidor de páginas Web Apache [34].

Las principales características de MySQL son las siguientes:

- Está escrito en C/C++ y probado en numerosos compiladores.
- Funciona en sistemas operativos Windows y Unix.
- Dispone de un driver ODBC para Windows, aportando compatibilidad con la mayoría de lenguajes de programación de este sistema operativo.
- Se puede comunicar haciendo uso del lenguaje SQL, lo cual permite que sea compatible con otros sistemas de administración de bases de datos.

### 2.5.4. Framework Yii

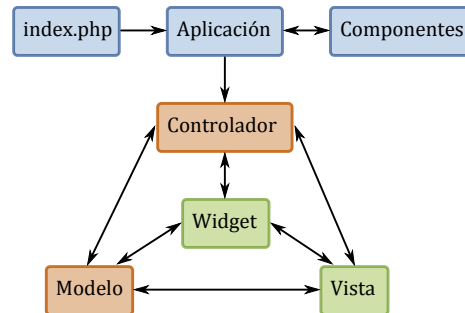
Yii es un *framework* de aplicaciones Web escrito en PHP5, de alto rendimiento y basado en componentes. El nombre Yii es un acrónimo de *Yes, it is*. Yii abarca la filosofía de convención sobre configuración, es decir, que tiene parámetros por defecto para casi todos los aspectos usados en la configuración de una aplicación. Gracias a esto, se requiere escribir menos código y toma menos tiempo el desarrollo de una aplicación. Sin embargo, permite personalizar todos los parámetros por defecto para un uso en particular [43, 45].

Yii puede ser usado en aplicaciones a cualquier escala, lo que significa que se puede desarrollar una aplicación sencilla (relativamente) como un sitio Web informativo, como una aplicación de uso empresarial o industrial. Yii fomenta la reusabilidad de código en la mayor medida posible en la programación Web, lo que acelera significativamente el proceso de desarrollo.

El *framework* Yii está diseñado para ayudar a desarrollar con la filosofía DRY (*Don't Repeat Yourself*), el cual es un concepto clave en el desarrollo de aplicaciones ágiles. Todas las aplicaciones Yii son construidas usando la arquitectura MVC, por ello, este patrón de desarrollo se fomenta con la asignación de directorios (Rutas) específicos en donde se almacena cada fragmento de código MVC. Esto minimiza la duplicación de código, promueve la reusabilidad y facilita el mantenimiento. En la figura 2.22 se ilustra el diagrama de la estructura estática de una aplicación Yii.

Los componentes de la arquitectura MVC se describen a continuación:

- *Modelo*: En una arquitectura MVC, el Modelo es responsable de la información y debe encapsular las reglas de intercambio (*Business Rules*) que aplican a dicha información. En Yii se hace uso del patrón de diseño *Active Record* (AR) para el acceso de base de datos abstracto en forma de objetos. Es decir, se obtienen las relaciones (Tablas) de la base de datos en forma de objetos PHP.



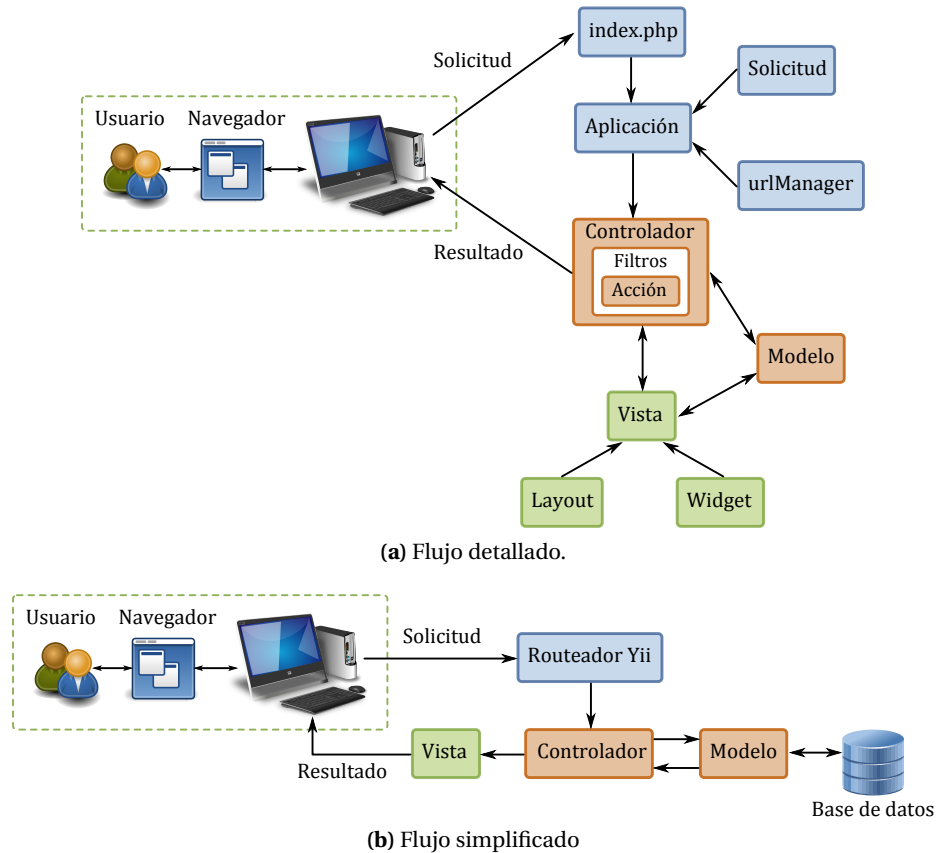
**Figura 2.22.** Arquitectura MVC del framework Yii.

- **Vista:** La Vista es responsable de presentar la interfaz de usuario, la cual frecuentemente está basada en la información contenida en los modelos. En la Vista se puede hacer uso de *Widgets*, que son pequeños programas o aplicaciones, usualmente presentados en archivos o ficheros pequeños que son ejecutados por un motor de widgets o *Widget Engine*. Actualmente existen muchos widgets para el framework Yii y son utilizados para acelerar el desarrollo.
- **Controlador:** El Controlador es el actor y director principal de una solicitud a una ruta (de la aplicación). Es responsable de tomar los datos de entrada del usuario (que realizó la solicitud), interactuar con el Modelo e indicar a la Vista que se actualice y despliegue apropiadamente la información solicitada.

En la figura 2.23a y 2.23b se ilustra, de manera detallada y simplificada respectivamente, el flujo de trabajo en una aplicación Yii cuando se procesa una solicitud realizada por el usuario final [43, 45].

Los diagramas de flujo de trabajo de la figura 2.23 se describen paso a paso de la siguiente manera:

- El usuario realiza una solicitud a través de una URL y el servidor Web la procesa ejecutando el *script* (archivo de código) principal *index.php*. El cual crea una instancia de la Aplicación y la ejecuta.
- La aplicación obtiene información detallada de la solicitud del usuario.
- La aplicación determina el Controlador y Acción a ejecutar con la ayuda del componente de la aplicación llamado *urlManager* y crea una instancia de dicho controlador para procesar la solicitud del usuario.
- Al realizar una Acción principalmente se tienen dos casos: que el usuario desee guardar o modificar información de un Modelo o simplemente pedir información del mismo.
- La Acción despliega una Vista con la información del Modelo. Generalmente las Vistas están asociadas a las Acciones de un Controlador.
- La Vista puede ejecutar widgets y el resultado de la presentación es empotrada (Integrada) en el diseño principal (*Layout*).
- Por último, la Acción completa la presentación de la Vista desplegando el resultado al usuario.



**Figura 2.23.** Flujo de trabajo en una aplicación Yii.

### 2.5.5. JavaScript

JavaScript es un lenguaje de programación para el desarrollo de aplicaciones con arquitectura Cliente/Servidor a través de Internet. Una aplicación en JavaScript tiene la particularidad de que está embebido en el mismo documento HTML, el cual presenta dicha aplicación al usuario [20].

Una aplicación en JavaScript, es un sistema interactivo que tiene la capacidad de detectar eventos o acciones realizadas por los usuarios. JavaScript fue diseñado para ser un lenguaje de elaboración de *scripts* que pudieran incrustarse en archivos HTML. No es compilado, sino que, en vez de ello, es interpretado por el navegador como código fuente.

En la actualidad existen muchas herramientas (*frameworks*) que aceleran el desarrollo de aplicaciones usando JavaScript. Los *frameworks* de JavaScript más populares son: JQuery, Ext JS, Prototype, entre otros.

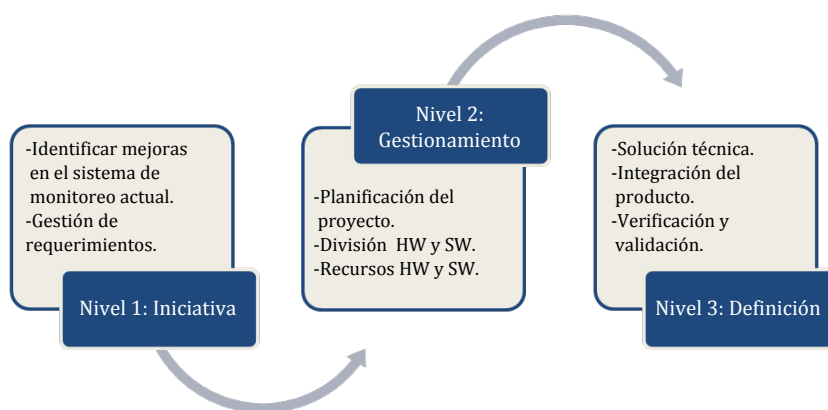


# 3

## Desarrollo del Sistema

En este capítulo se presenta el desarrollo del prototipo de sistema de monitoreo para la PTAR de la UTM utilizando una metodología de desarrollo para mejoramiento de procesos de producción. Dicha metodología es conocida como Modelo de Integración de Capacidad y Madurez (CMMI, *Capability Maturity Model Integration*) [27].

CMMI consiste de 5 niveles de madurez y cada nivel consta de un número de áreas de procesos. Sin embargo, debido a que en los diferentes niveles del modelo CMMI existen áreas de procesos enfocadas a organizaciones o empresas ya establecidas, la aplicación de CMMI en el desarrollo de este sistema se limitó a aspectos de ingeniería, por lo que solo se implementaron los niveles 1, 2 y 3. Estos niveles se detallan en la figura 3.1.



**Figura 3.1.** Niveles de la metodología CMMI.

### 3.1. Iniciativa

#### 3.1.1. Estado actual de la PTAR

En la PTAR de la UTM no se cuenta con un sistema automatizado para el monitoreo de pH y OD, por lo que este proceso se realiza manualmente.

El monitoreo de pH se realiza mediante los siguientes pasos:

1. Se toma una muestra de 50ml de agua residual.
2. Con un potenciómetro se toma la lectura del pH.
3. Se registra el valor de la medición en una bitácora.
4. Se tabulan los resultados.

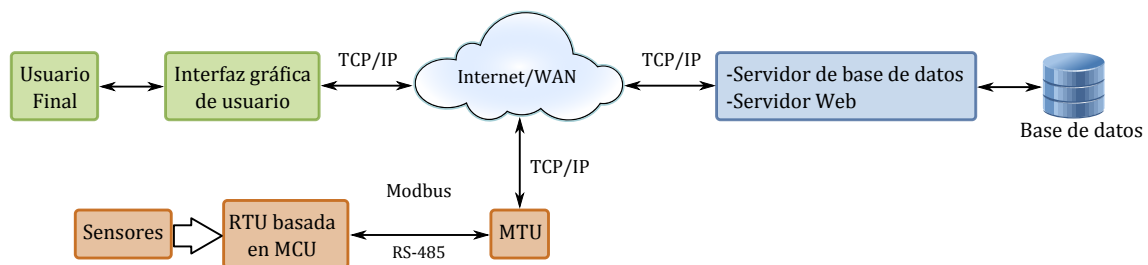
El monitoreo de OD se realiza de la siguiente manera:

1. Se calibra previamente el electrodo de OD, el cual se utiliza para medir este parámetro.
2. Se sumerge el electrodo en el bioreactor y se toman las lecturas.
3. Se registra el valor de la medición en una bitácora.
4. Se tabulan los resultados.

Por lo anterior y considerando que no se lleva a cabo el monitoreo de temperatura, se propone realizar la medición de pH, OD y temperatura mediante un sistema automatizado.

#### 3.1.2. Propuesta

El sistema que se propone desarrollar tiene como objetivo implementar un sistema SCADA para monitorear remotamente y de manera automatizada el estado de pH, OD y temperatura de la PTAR de la UTM. Dicho sistema consta de sensores de propósito industrial (Sección 2.3); una RTU; la MTU; un sistema de comunicaciones que consta de una interfaz serie RS-485; servidores de procesamiento distribuido; y, acceso Web con una interfaz de usuario. En la figura 3.2 se muestra un diagrama general del sistema propuesto.



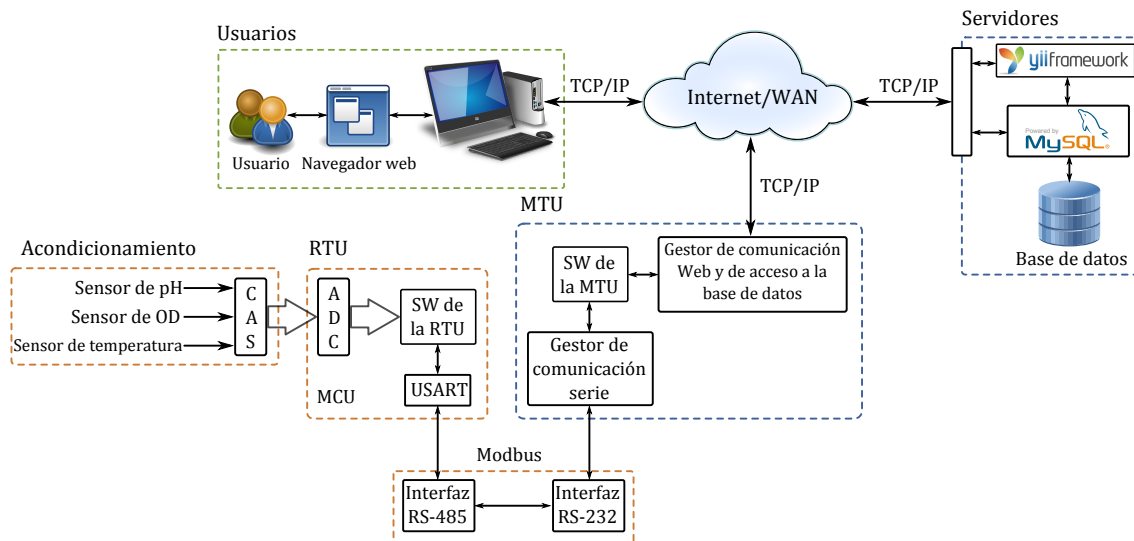
**Figura 3.2.** Diagrama general del sistema SCADA.



## 3.2. Gestionamiento

### 3.2.1. Requerimientos funcionales de la propuesta

En la figura 3.3 se muestra de manera detallada el sistema SCADA propuesto. Este sistema realizará el monitoreo remoto de pH, OD y temperatura de la PTAR. Las mediciones proporcionadas por cada sensor pasarán por una etapa de acondicionamiento de señal mediante un sistema electrónico (CAS, *Circuito de Acondicionamiento de Señal*). El MCU ATmega8 desempeñará el papel de RTU, implementará las funciones del protocolo Modbus, procesará solicitudes de la MTU y realizará la adquisición de datos de los sensores a través de un módulo ADC (integrado en el MCU). El sistema de comunicaciones implementará una interfaz serie RS-232/RS-485 a través de un transceptor (*Transceiver*), que servirá para la comunicación entre la RTU y la MTU. La MTU (Computadora) implementará las funciones del protocolo Modbus para solicitar información de la RTU, almacenará registros de mediciones en una base de datos y recibirá órdenes que emite el usuario a través de la interfaz Web. Los servidores proporcionan el gestor de bases de datos, la base de datos y la aplicación Web (GUI). La aplicación Web incluye, la gestión de usuarios (Administrador), visualización de tendencias y de entradas y salidas digitales.



**Figura 3.3.** Diagrama completo del sistema SCADA propuesto.

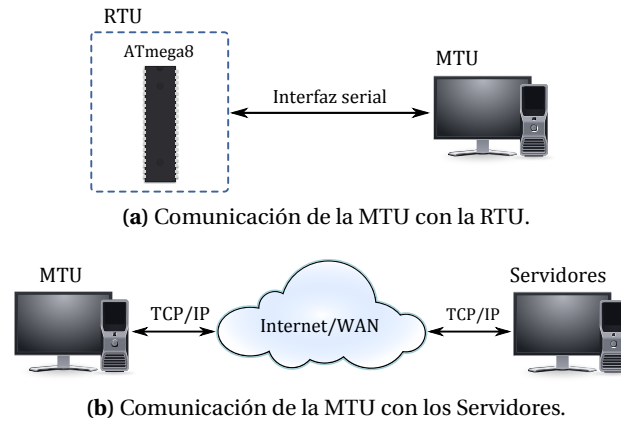
#### 3.2.1.1. Clasificación y análisis de requerimientos

Los requerimientos funcionales de la propuesta, para un mejor análisis, son clasificados de la siguiente manera:

- Acondicionamiento de señal:** La señal que proviene de los sensores requiere de una etapa de acondicionamiento, por ello se hace uso de un sistema de acondicionamiento de señal o CAS. El CAS emplea un sistema electrónico, el cual tiene como entrada la señal de un sensor

y proporciona en la salida un nivel de señal o voltaje adecuado para el funcionamiento de la etapa posterior de adquisición de datos.

- *Especificación del CAS de cada sensor*: Para los sensores de pH y temperatura, el CAS para cada uno de ellos se ha dividido en tres etapas: Aislamiento, Filtrado y Amplificación. El CAS para el sensor de OD se ha dividido en las siguientes etapas: Conversión de corriente a voltaje, Amplificación y Filtrado.
- *Software de la RTU*: Para implementar la RTU del protocolo Modbus en el MCU ATmega8, se requiere de un *software*. Dicho *software* se construye en lenguaje C, en el que se codifican los parámetros de configuración de periféricos de entrada/salida, los formatos de las ADU de solicitudes y respuestas y el procesamiento de solicitudes de la MTU.
- *Funciones de la RTU*: Las funciones primordiales del MCU (RTU) son obtener mediciones de los sensores, proporcionar el voltaje de alimentación para el sensor de OD, recibir y procesar las ADUs de solicitud que emite la MTU y construir y enviar las ADU de respuesta (con la información solicitada) a la MTU.
- *Comunicación MTU-RTU*: La comunicación entre la RTU y la MTU se realizará a través de una interfaz serie RS-485 como medio físico. La MTU (Computadora) cuenta con un puerto serie, cuyo funcionamiento se basa en el estándar RS-232, por ello, se requiere un transceptor entre las interfaces RS-485 y RS-232.
- *Software de la MTU*: El protocolo Modbus de la MTU se implementará desarrollando un *software* en lenguaje Java, en el cual se codifican los formatos de las ADU de solicitudes y respuestas Modbus. Este *software* contará con un controlador para la gestionar la comunicación serie con la RTU (Figura 3.49a), un gestor de monitoreo discreto de la RTU, un módulo que comunica la MTU con el servidor de bases de datos y un módulo de comunicación con la aplicación Web (Figura 3.49b).
- *Servidor Web y base de datos*: La aplicación Web se desarrollará haciendo uso del *framework* *Yii* (arquitectura MVC), dicha aplicación restringirá el acceso sólo al Administrador del sistema, permitirá (al Administrador) visualizar el estado de pH, OD y temperatura a través de gráficas de tendencias y de manera tabular, visualizar el estado de las alarmas a través de indicadores y cambiar el estado de las salidas digitales en la RTU.
- *Herramientas de diseño y desarrollo*: En el diseño *hardware* se emplearán herramientas de simulación y diseño de circuitos electrónicos. Para el desarrollo del *software* del MCU, se empleará un entorno de desarrollo exclusivo del MCU, el cual utiliza el lenguaje de programación C. En el desarrollo del *software* de la MTU y de la aplicación Web, se emplearán entornos de desarrollo para lenguajes de programación orientado a objetos y de alto nivel.
- *Entradas y salidas digitales*: El MCU se configuró para incluir dos entradas y dos salidas digitales, aunque su implementación con dispositivos externos no fue desarrollada, se permite conectar entradas y salidas digitales externas que estén debidamente aisladas y que operen con niveles de voltaje TTL.



**Figura 3.4.** Comunicación de la MTU con la RTU y con el servidor de la aplicación Web y la base de datos.

### 3.2.2. División *hardware* y *software*

Las fases de diseño e implementación de *hardware* y *software* del sistema SCADA se realizaron utilizando la metodología de desarrollo para Sistemas Empotrados (*Embedded Systems*) [25, 47]. En base a los requerimientos del sistema, la división de tareas *hardware* y *software* se listan en la tabla 3.1.

**Tabla 3.1.** Tareas *hardware* y *software*.

<i>Hardware</i>	<i>Software</i>
Acondicionamiento de los sensores	Diseño <i>software</i> de la RTU
Entradas y salidas de la RTU	Diseño <i>software</i> de la MTU
Interfaz RS-485	Diseño <i>software</i> de la aplicación Web
Transceptor RS-232/RS-485	

#### 3.2.2.1. Selección de hardware

Los elementos *hardware* seleccionados para satisfacer los requerimientos funcionales de la propuesta del sistema SCADA son:

- **CA3140:** Es un *op-amp* (*Amplificador Operacional*) con tecnología MOSFET que cuenta con una alta impedancia de entrada  $Z_{in} = 1.5T\Omega$ , opera con corrientes de entrada muy bajas  $I_{min} = 10pA$  a un voltaje de alimentación  $V_s = \pm 15V$ . Su campo de aplicación abarca las áreas de instrumentación, filtros activos, generadores de funciones, instrumentos portátiles, etc [15].

El *op-amp* CA3140 se utiliza para implementar todas las etapas del CAS (aislamiento, conversión corriente-voltaje, filtrado y amplificación) de cada sensor.

- **TL081:** Es un *op-amp* de bajo costo, de alta velocidad, con entrada JFET y con un voltaje offset de entrada internamente reducido. Este dispositivo requiere una baja corriente

de alimentación, sin embargo, ofrece un amplio ancho de banda de ganancia y rápida velocidad de respuesta [40].

EL circuito integrado (CI) TL081 puede ser usado en aplicaciones como integradores de alta velocidad, convertidores D-A (Digital Análogo), circuitos de muestreo y retención y otras.

- *Capacitor de poliéster*: Sus principales aplicaciones son en sistemas de audio, equipos de instrumentación y filtros. Por lo tanto, estos capacitores se utilizan en el diseño de la etapa de filtrado del CAS de cada sensor.
- *ATmega8*: Es un MCU de la marca ATMEL que cuenta con una arquitectura AVR RISC, memoria flash para código de 8kB, frecuencia máxima de operación de 16MHz, seis canales ADC, un dispositivo de comunicación serie USART (*Universal Synchronous and Asynchronous serie Receiver and Transmitter*), etc [3].

El ATmega8 se utiliza para implementar la RTU, lo cual se logra codificando en la memoria de código el protocolo Modbus y el *software* de control, los canales ADC se utilizan para obtener las mediciones de los sensores, se asigna un puerto de salida de 8 bits que indica al DAC el nivel de voltaje para alimentar el sensor de OD y el dispositivo USART se utiliza para la comunicación serie con la MTU.

- *MAX489*: Es un CI de la marca MAXIM que tiene la función de transeptor para comunicaciones RS-485, el cual permite una transmisión de datos libre de errores a una velocidad de 250kbps y se puede configurar para operar en modo *full-duplex* y *half-duplex* [19].

El CI MAX489 se utiliza para implementar la interfaz RS-485, lo cual se logra al conectarse con el dispositivo USART del ATmega8.

- *MAX232*: Es un CI de la marca MAXIM que convierte las señales de un puerto serie RS-232 a señales compatibles con niveles TTL [39].

El CI MAX232 se utiliza para la comunicación serie entre la MTU y la RTU, debido a que el puerto serie de la MTU (computadora) basa su funcionamiento de acuerdo al estándar RS-232.

- *DAC0808*: Es un DAC de 8 bits con un tiempo de establecimiento a escala completa de 150ms, cuenta con un registro de entrada de 8 bits que indica el nivel de voltaje a generar y opera con fuentes de alimentación de  $\pm 5V$  [38].

El CI DAC0808 se utiliza para generar el voltaje de alimentación del sensor de OD, por lo que el registro de entrada se conecta al registro de proporciona el ATmega8.

En la tabla 3.2 se muestra la relación entre las tareas HW y los elementos HW seleccionados.

**Tabla 3.2.** Relación entre tareas y componentes hardware.

Tareas hardware	CA3140	Capacitor	ATmega8	MAX232	MAX489	DAC0808
Acondicionamiento	X	X				
Entradas/salidas de la RTU			X			
Interfaz RS-232/RS-485			X	X	X	
Alimentación del sensor de OD			X			X

### 3.2.2.2. Selección de herramientas *software* de diseño y desarrollo

Las herramientas de diseño de *hardware* y *software* seleccionados para el desarrollo de la propuesta se describen a continuación:

- *AVR Studio*: Herramienta *software* de la compañía ATMEL versión 4.16 proporcionada de forma gratuita. Es un entorno de desarrollo de aplicaciones para MCUs del mismo fabricante, el cual permite escribir, compilar y depurar códigos en lenguaje C/C++ y lenguaje ensamblador.
- *Proteus Design Suite*: Herramienta *software* en la versión 7.6, la cual es desarrollada y comercializada por *Labcenter Electronics Ltd.* Permite el diseño de diagramas esquemáticos, simulación de circuitos mediante un sistema basado en SPICE e incluye una suite de herramientas de diseño de tarjetas de circuito impreso (PCB, *Printed Circuit Board*).
- *Cadence OrCAD*: Es una herramienta *software* comercial de *Cadence Design Systems, Inc.* en su versión 10.3 permite diseñar y editar diagramas esquemáticos para el desarrollo de PCB o simulación analógica utilizando modelado SPICE.
- *Docklight*: Es un producto *software* de *Flachmann & Heggelbacher*, en la versión 1.9.21, ayuda a realizar pruebas de conexión y comunicación en interfaces serie.
- *Netbeans*: Herramienta de *software* libre (*Open Source*) desarrollada y mantenida por la comunidad *Netbeans (NetBeans Community)*. Se utilizaron las versiones 7.0.1 y 7.3.1, esta herramienta es un entorno de desarrollo integrado (IDE, *Integrated Development Environment*) que permite el desarrollo de aplicaciones escritas en lenguaje *Java, JavaScript, PHP* y otros. También provee un módulo para el diseño de GUIs en Java.
- *Eclipse*: Herramienta de *software* libre desarrollada y soportada por la comunidad *Eclipse*, en la que personas y organizaciones colaboran para desarrollar y mejorar dicha herramienta. Se utilizó la versión 3.7, el cual es un IDE para el desarrollo de aplicaciones escritas en lenguaje C/C++, Python, Java, JavaScript, PHP y otros. Eclipse proporciona herramientas para el desarrollo de GUIs.
- *Enterprise Architect*: Es un producto propiedad de *SPARX SYSTEMS*. En la versión 9 de evaluación, es un conjunto de herramientas para modelado y diseño de *software*, por lo que sirve de asistente en la gestión de proyectos *software*, proporcionando múltiples herramientas de modelado y herramientas de creación de diagramas que facilitan el desarrollo.
- *XAMPP*: Las herramientas necesarias para el desarrollo de la aplicación Web, están contenidas en la suite XAMPP en la versión 1.8.1. Esta suite contiene las siguientes herramientas: PHP 5.4.7 (requerido para desarrollar la aplicación Yii), gestor de base de datos MySQL 5.5.27, servidor Apache 2.4.3 y otros.
- *Framework Yii*: Para desarrollar la aplicación Web se utilizó el *framework Yii* versión 1.1.13. Yii, por ser un *framework* de *software* libre, está en constante evolución, por lo que actualmente hay nuevas versiones que incluyen algunas mejoras.

### 3.3. Definición

#### 3.3.1. Diseño hardware

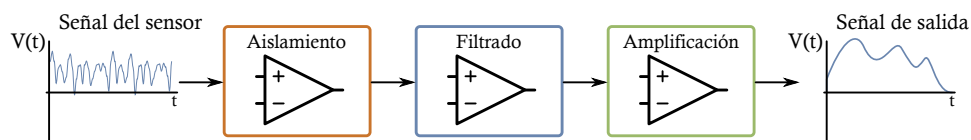
Como se describe en la tabla 3.1, el desarrollo del sistema se divide en cuatro etapas *hardware*. Estas etapas se describen a continuación:

- *Etapa de acondicionamiento de los sensores*: En esta etapa se detalla el diseño específico de acondicionamiento para cada sensor en particular.
- *Entradas y salidas de la RTU*: Esta etapa describe la configuración de entradas y salidas del MCU ATmega8 y la distribución de terminales.
- *Interfaz RS-485*: Describe la configuración de la interfaz serie RS-485.
- *Transceptor RS-232/RS-485*: Especifica el diseño de la comunicación entre las interfaces RS-485 y RS-232. La interfaz RS-485 es usada por la RTU y la interfaz RS-232 es usada por la MTU.

##### 3.3.1.1. Diseño de la etapa de acondicionamiento de los sensores

###### 3.3.1.1.1. CAS del sensor de pH

El diseño del CAS para el sensor de pH consiste de tres etapas, como se ilustra en la figura 3.5, las cuales son: aislamiento, filtrado y amplificación. No obstante, como se describe en la subsección 2.3.1, se indica que el voltaje de la señal de salida del sensor de pH ( $V_{pH}$ ) oscila entre  $-414.12\text{mV}$  y  $+414.12\text{mV}$ , por lo cual, se requiere de una etapa previa en la que se agrega un voltaje de desplazamiento (*offset*), el cual deberá ser mayor a  $414\text{mV}$ . Con ello se garantiza una señal de salida con voltaje positivo, que es aceptable para la entrada del ADC, ya que el ADC solo acepta niveles de voltaje positivos.



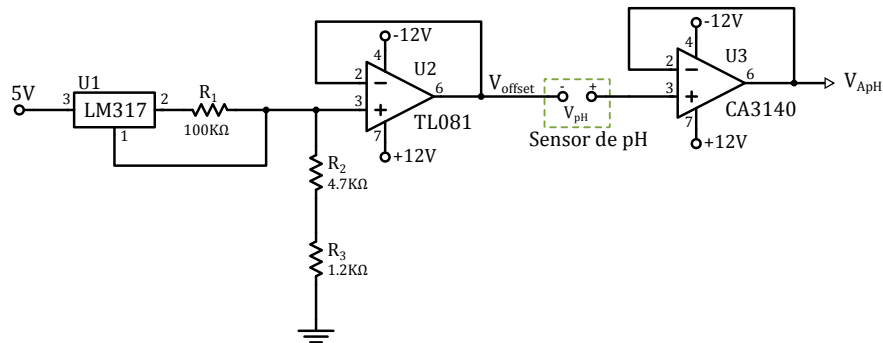
**Figura 3.5.** Etapas del CAS para el sensor de pH y de temperatura.

Para la etapa que genera el voltaje *offset* ( $V_{offset}$ ) se emplea el regulador de voltaje LM317 con la configuración de limitador de corriente y el op-amp TL081 como seguidor de voltaje (Figura 3.6a) [2, 9, 14].

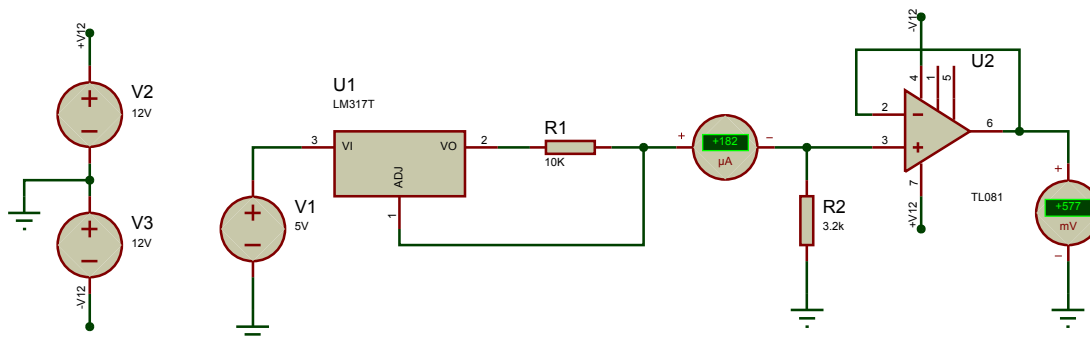
La idea de construir una fuente de voltaje fija para generar  $V_{offset}$  empleando el CI LM317, surgió de [31], en donde se muestra que es posible obtener una fuente de corriente constante a partir de dicho dispositivo y al variar la resistencia de carga se ajusta el voltaje de salida. En base a los cálculos propuestos en [31] se obtuvo la siguiente configuración (Figura 3.6b):  $R_1 = 10\text{k}\Omega$  y  $R_2 = 3.2\text{k}\Omega$  para obtener  $V_{offset} = 400\text{mV}$ . Sin embargo, en la simulación que se muestra en la figura 3.6b, se obtuvo  $V_{offset} = 577\text{mV}$ .

No obstante, debido a que las especificaciones del LM317 varían con cada fabricante, este diseño no funcionó de la misma manera en la práctica. Por lo que en la figura 3.6a se muestra la configuración funcional del LM317 en donde se obtiene  $V_{offset} = 505.6\text{mV}$  como se detalla en la sección 4.1.2. Con lo que se garantiza un desplazamiento de la señal  $V_{pH}$  para obtener niveles de voltaje positivo.

La etapa de aislamiento es requerida debido a la alta impedancia de salida del sensor de pH. En dicha etapa se utiliza el CI CA3140 configurado como seguidor de voltaje (figura 3.6a) con lo que se obtiene  $V_{ApH} = V_{pH} + V_{offset}$ .



(a) Adición de  $V_{offset}$  y aislamiento del sensor de pH.



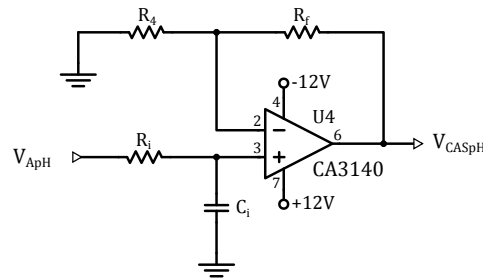
(b) Simulación de la etapa de adición de  $V_{offset}$ .

**Figura 3.6.** Configuración y simulación de la etapa de adición de  $V_{offset}$  y etapa de aislamiento para el CAS de pH.

En la etapa de filtrado se emplea la configuración de un filtro activo pasa-bajas, como el de la figura 3.7 [2, 9, 14], en donde el filtrado se realiza utilizando un circuito  $RC$  y el CI CA3140 como amplificador de ganancia variable, con ello la amplificación se incluye en el diseño de la misma etapa de filtrado<sup>1</sup>.

La frecuencia de corte  $f_c$  del filtro *pasa-bajas* se calcula con la ecuación 3.1 y la ganancia de amplificación  $G_a$  se calcula con la ecuación 3.2.

<sup>1</sup>La configuración completa del CAS del sensor de pH se muestra en el apéndice A.



**Figura 3.7.** Configuración del filtro *pasa-bajas* con ganancia ajustable.

$$\omega_c = \frac{1}{R_i C_i} = 2\pi f_c \quad (3.1)$$

$$G_a = \left(1 + \frac{R_f}{R_4}\right) \quad (3.2)$$

en estas ecuaciones se tiene que:

$\omega_c$  = Frecuencia de corte en [rad/s].

$f_c$  = Frecuencia de corte en [Hz].

$R_i$  = Resistencia del circuito RC.

$C_i$  = Capacitor del circuito RC.

$G_a$  = Ganancia del amplificador.

$R_f$  = Resistencia de retroalimentación del amplificador.

$R_4$  = Resistencia de entrada del amplificador.

Debido a que la señal del sensor de pH es propensa a contaminarse con ruido, se establece una frecuencia de corte baja  $f_c = 0.159\text{Hz}$ . Para una mejor compatibilidad con el ADC del ATmega8, se eligió una ganancia de amplificación  $G_a = 2$ . Los parámetros para el cálculo del filtro y el amplificador se muestran en la tabla 3.3.

**Tabla 3.3.** Parámetros usados en el CAS del sensor de pH.

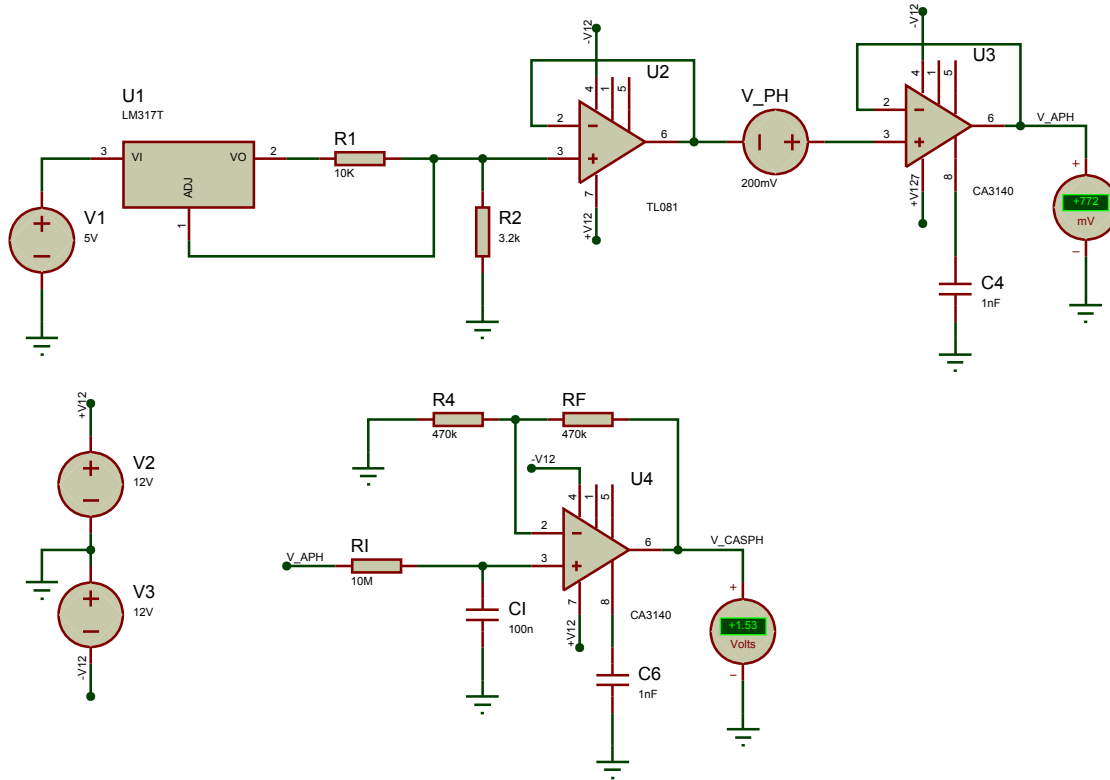
Parámetro	Valor	Unidad
$R_i$	10	M $\Omega$
$C_i$	100	nF
$f_c$	0.159	Hz
$R_f$	470	k $\Omega$
$R_1$	470	k $\Omega$
$G_a$	2	-

Por todo lo anterior, el modelo matemático del dispositivo CAS para el sensor de pH se describe en la siguiente ecuación:

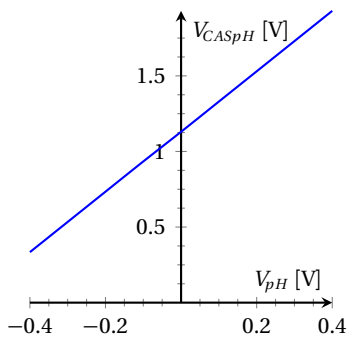
$$V_{CASpH} = 2(V_{pH} + V_{offset}) \quad (3.3)$$



En la figura 3.8 se muestra la simulación completa del CAS del sensor de pH. En la simulación se sustituye la entrada del sensor de pH por una fuente de voltaje ( $V_{pH}$ ). Se considera que el dominio (entrada del CAS) está en el rango de voltaje de  $-414.12\text{mV}$  a  $+414.12\text{mV}$ , con ello, en la figura 3.9 y en la tabla 3.4 se muestran los resultados que arroja la simulación de manera gráfica y tabular respectivamente.



**Figura 3.8.** Simulación Completa del CAS para el sensor de pH.



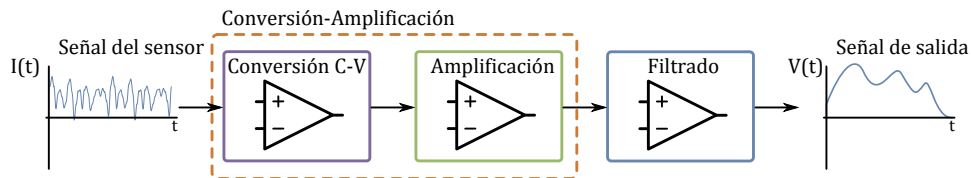
**Figura 3.9.** Gráfica de entradas y salidas de la simulación del CAS para el sensor de pH.

**Tabla 3.4.** Tabla de entradas y salidas de la simulación del CAS para el sensor de pH.

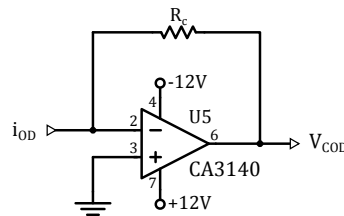
$V_{pH}$	$V_{CASpH}$
400mV	1.93V
300mV	1.73V
200mV	1.53V
100mV	1.33V
0V	1.13V
-100mV	934mV
-200mV	734mV
-300mV	534mV
-400mV	334mV

### 3.3.1.1.2. CAS del sensor de OD

El diseño del CAS para el sensor de OD se divide en dos etapas principales (figura 3.10): conversión-amplificación y filtrado. La etapa de conversión-amplificación se divide en dos subetapas, en donde se involucra el diseño de una etapa de conversión de corriente a voltaje y una etapa de amplificación.

**Figura 3.10.** Etapas del CAS para el sensor de OD.

Debido a que el sensor de OD, como se describió en la subsección 2.3.2, proporciona una corriente como señal de salida y el ADC del ATmega8 solo permite voltajes como señales de entrada, se requiere el diseño de un convertidor de corriente a voltaje, como el que se muestra en la figura 3.11 [2, 14], para acondicionar la señal.

**Figura 3.11.** Configuración de la etapa de conversión-amplificación .

La etapa de amplificación para el CAS del sensor de OD, se incluye en este mismo diseño, ya que el convertidor de la figura 3.11 se configura para que tenga una ganancia  $G_c = R_c$ . Este convertidor de corriente a voltaje se diseña utilizando la ecuación 3.4. Debido a que la corriente de la señal de salida del sensor de OD oscila entre 50nA y 110nA, se establece una ganancia de

conversión  $G_c = 1 * 10^6$ , con ello se obtiene un voltaje de conversión  $V_{COD}$  que oscila entre 50mV y 110mV. Estos niveles de voltaje permitirán al ADC operar de manera satisfactoria en la toma de mediciones.

$$V_{COD} = -R_c i_{OD} \quad (3.4)$$

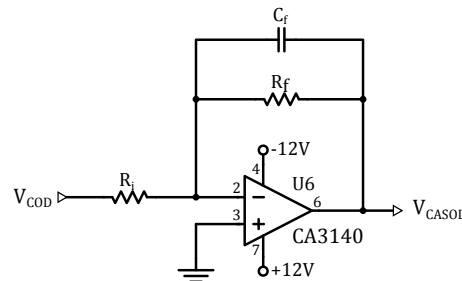
donde se tiene que:

$V_{COD}$  = Voltaje de salida del convertidor.

$i_{OD}$  = Corriente del sensor de OD.

$R_c$  = Resistencia de ganancia de conversión.

Para la etapa de filtrado, se emplea la configuración de un filtro activo pasa-bajas inversor [2, 9, 14]. Dicho filtro es de primer orden con ganancia unitaria como el que se muestra en la figura 3.12.



**Figura 3.12.** Configuración del filtro *pasa-bajas* inversor de ganancia unitaria.

La frecuencia de corte  $f_c$  de dicho filtro se calcula con la ecuación 3.5.

$$\omega_c = \frac{1}{R_f C_f} = 2\pi f_c \quad (3.5)$$

donde:

$\omega_c$  = Frecuencia de corte en [rad/s].

$f_c$  = Frecuencia de corte en [Hz].

$R_f$  = Resistencia de retroalimentación del filtro.

$C_f$  = Capacitor de retroalimentación del filtro.

$R_i$  = Resistencia de entrada al filtro, para la ganancia unitaria se hace  $R_i = R_f$ .

La frecuencia de corte de la etapa de filtrado elegida es  $f_c = 0.159\text{Hz}$ . En la tabla 3.5 se muestran los parámetros para el cálculo del convertidor de corriente a voltaje y del filtro pasa-bajas y en el apéndice A se muestra la configuración completa del CAS del sensor de OD.

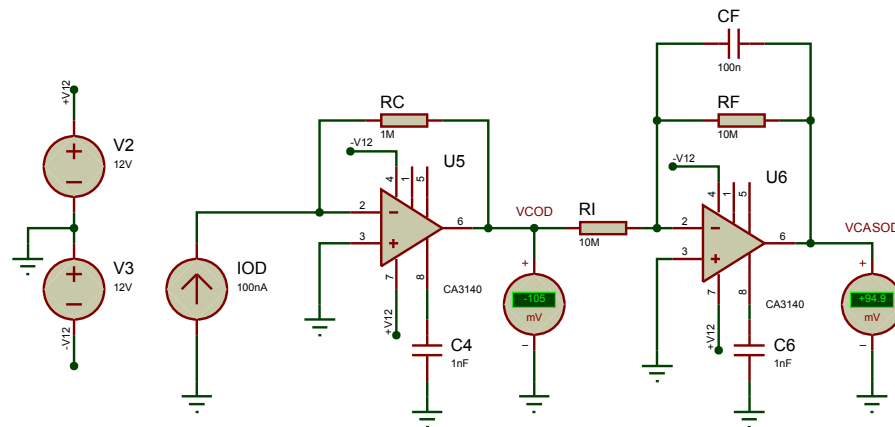
**Tabla 3.5.** Parámetros usados en el CAS del sensor de OD.

Parámetro	Valor	Unidad
$G_c$	$1 * 10^6$	-
$R_c$	1	$M\Omega$
$f_c$	0.159	Hz
$R_f$	10	$M\Omega$
$C_f$	100	nF
$R_i$	10	$M\Omega$

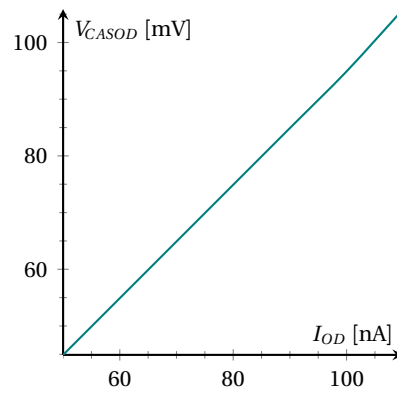
Basándose en el diseño anterior, el modelo matemático del dispositivo CAS para el sensor de OD se describe en la siguiente ecuación:

$$V_{CASOD} = (1 * 10^6)(i_{OD}) \quad (3.6)$$

En la figura 3.13 se ilustra el diagrama esquemático empleado en la simulación del CAS para el sensor de OD. En esta simulación se utiliza como entrada una fuente de corriente. Se considera que el dominio del CAS del sensor de OD es una corriente  $I_{OD}$  que oscila entre 50nA y 110nA, por lo que en la tabla 3.6 y en la figura 3.14 se muestran los resultados obtenidos de la simulación de manera tabular y gráfica respectivamente.

**Figura 3.13.** Simulación del CAS para el sensor de OD.**Tabla 3.6.** Tabla de entradas y salidas de la simulación del CAS para el sensor de OD.

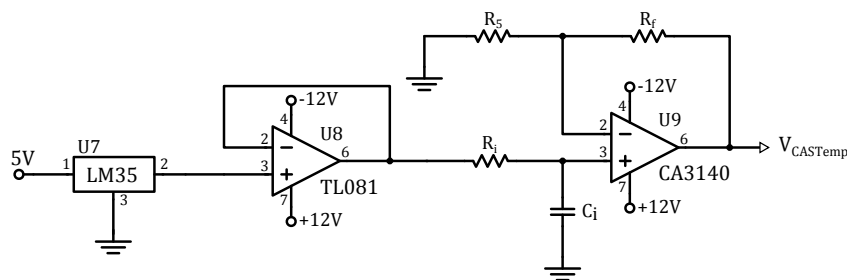
$I_{OD}$	$V_{CASOD}$
50nA	44.9mV
60nA	54.9mV
70nA	64.9mV
80nA	74.9mV
90nA	84.9mV
100nA	94.9mV
110nA	105.9mV



**Figura 3.14.** Gráfica de entradas y salidas de la simulación del CAS para el sensor de OD.

### 3.3.1.1.3. CAS del sensor de temperatura

El diseño del CAS del sensor de temperatura se dividió en las etapas de aislamiento, filtrado y amplificación (Figura 3.5).



**Figura 3.15.** Configuración del CAS para el sensor de temperatura.

La etapa de aislamiento emplea la configuración de un seguidor de voltaje (figura 3.15), en donde se obtiene un voltaje de salida  $V_{temp}$  que es proporcional a la temperatura ambiente con una sensibilidad de  $10\text{mV}/^{\circ}\text{C}$ .

Para la etapa de filtrado y amplificación se emplea la misma configuración usada para el sensor de pH (figura 3.15), la frecuencia de corte del filtro seleccionada es  $f_c = 1.59\text{Hz}$  y para una mejor resolución del módulo ADC del ATmega8 se eligió una ganancia de amplificación  $G_a = 4$ . En la tabla 3.7 se muestran los parámetros utilizados para el cálculo del filtro y el amplificador.

Del diseño anterior, el modelo matemático del dispositivo CAS para el sensor de temperatura se describe en la siguiente ecuación:

$$V_{CASTemp} = 4 * V_{temp} \quad (3.7)$$

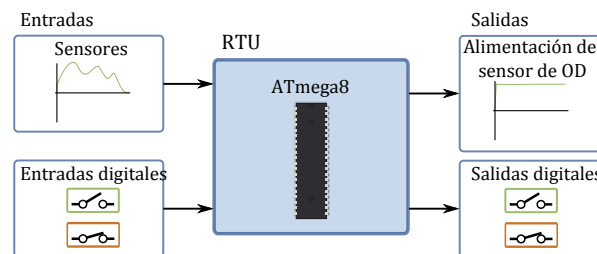


**Tabla 3.8.** Tabla de entradas y salidas de la simulación del CAS para el sensor de temperatura.

$V_{Temp}$	$V_{CASTemp}$
272mV	1.10V
282mV	1.14V
292mV	1.18V
302mV	1.23V
312mV	1.27V
322mV	1.31V

### 3.3.1.2. Entradas y salidas de la RTU

Para el diseño hardware de la RTU se definen las entradas y salidas del MCU ATmega8 (figura 3.18) [3]. Como entradas se tienen a las señales de los sensores y entradas digitales, y las salidas son la alimentación del sensor de OD y las salidas digitales.

**Figura 3.18.** Configuración de la RTU.

El voltaje de alimentación del sensor de OD se genera empleando el DAC0808, en donde el nivel de voltaje de salida se establece a través de un código binario contenido en un registro de entrada de 8 bits, el cual es generado por el ATmega8.

Las terminales del ATmega8 se distribuyen en cuatro áreas (figura 3.19): las entradas analógicas (señales de los sensores) que hacen uso de los canales ADC0-ADC2, la entrada y salida de datos serie a través de las terminales RXD y TXD respectivamente, las entradas y salidas digitales utilizan los puertos PD2-PD3 y PD4-PD5 respectivamente, y el registro de salida que indica el nivel de voltaje de alimentación para el sensor de OD utiliza el puerto B (PB0-PB7).

La configuración empleada para el DAC0808 se muestra en la figura 3.20, la cual proporciona un voltaje de salida  $V_{POD}$  con una resolución de 4.18mV/bit [38].

### 3.3.1.3. Interfaz RS-485

Para la comunicación de la RTU con la MTU se emplea una interfaz RS-485 como medio físico. Dicha interfaz se implementa utilizando el dispositivo USART que incluye el MCU, el cual se conecta al dispositivo MAX489 para implementar una línea de transmisión balanceada.

La configuración empleada para el diseño de la interfaz RS-485 se muestra en la figura 3.21 [19], en donde se utilizan los puertos RXD/TXD del ATmega8 para la recepción/transmisión de datos

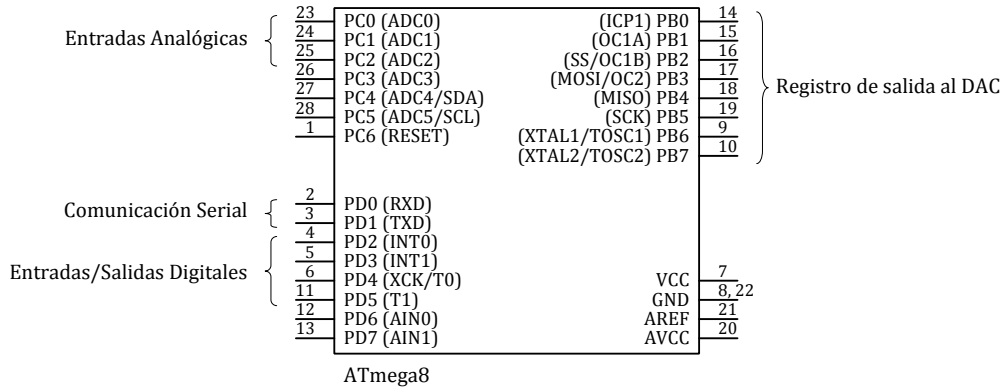


Figura 3.19. Distribución de terminales del ATmega8.

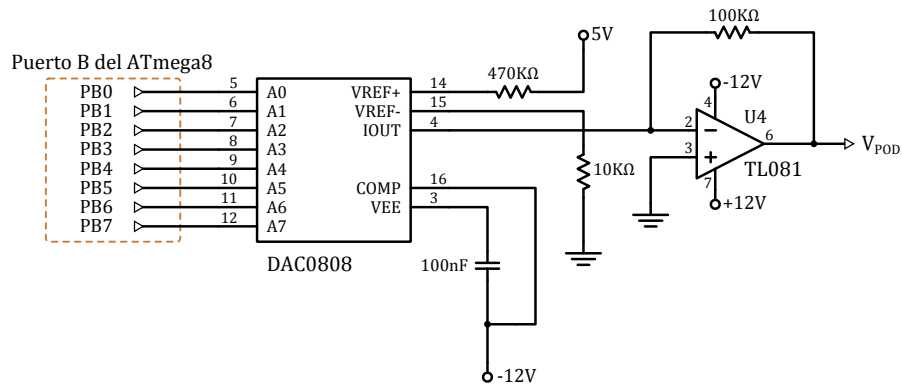


Figura 3.20. Configuración del DAC0808.

series mediante el dispositivo USART y el dispositivo MAX489 se configura para la operación en modo *full-duplex*.

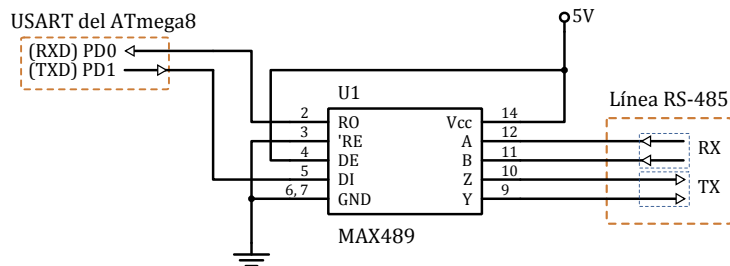
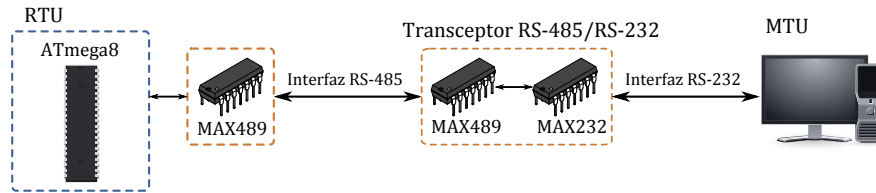


Figura 3.21. Configuración del dispositivo MAX489.



### 3.3.1.4. Transceptor RS-232/RS-485

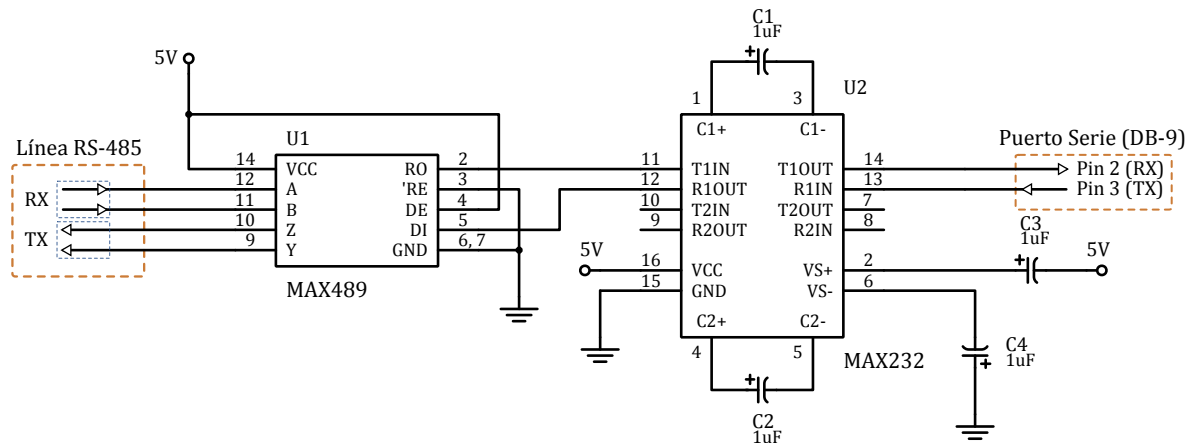
La MTU (computadora) cuenta con un puerto serie cuyo funcionamiento está basado en el estándar RS-232, debido a esto, para comunicar la MTU con la RTU es necesario implementar un transceptor RS-232/RS-485 (figura 3.22).



**Figura 3.22.** Comunicación serie de la RTU con la MTU.

El transceptor RS-232/RS-485 se divide en dos etapas, la primera etapa consiste en obtener los datos serie de la interfaz RS-485 en niveles TTL, lo cual se realiza empleando el dispositivo MAX489; y, la segunda etapa convierte los niveles TTL a voltajes del estándar RS-232, lo cual se realiza utilizando el dispositivo MAX232.

El dispositivo MAX489 se configura para operar en modo *full-duplex* y la configuración empleada para el dispositivo MAX232 utiliza capacitores externos para su funcionamiento [19, 39]. Dicha configuración se ilustra en la figura 3.23.



**Figura 3.23.** Transceptor RS-232/RS-485.

### 3.3.2. Diseño *software*

El desarrollo del sistema se divide en tres etapas *Software* (Tabla 3.1, ). El diseño de estas etapas se realizó empleando el Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*). Las etapas *software* se describen a continuación:

- *Software de la RTU*: En esta etapa se detalla el diseño *software* del MCU ATmega8. Con este *software*, el MCU implementará una RTU empleando el protocolo Modbus. El desarrollo se realizará en lenguaje C.
- *Software de la MTU*: Esta etapa describe a detalle el diseño del *software* de la MTU. La MTU de Modbus será implementada en una computadora. Este *software* se implementará en lenguaje Java.
- *Software de la aplicación Web*: Describe el diseño de la aplicación Web en base a los requerimientos establecidos. Se empleará el framework Yii y una base de datos MySQL.

#### 3.3.2.1. *software* de la RTU

##### 3.3.2.1.1. Requerimientos formales

En base a los requerimientos funcionales de la propuesta, se obtuvieron de manera concreta los requerimientos formales de las tareas que debe desempeñar el *software* de la RTU. Dichos requerimientos formales se describen en la tabla 3.9 y en la figura 3.24 se ilustra la relación entre los requerimientos mediante un diagrama de requerimientos.

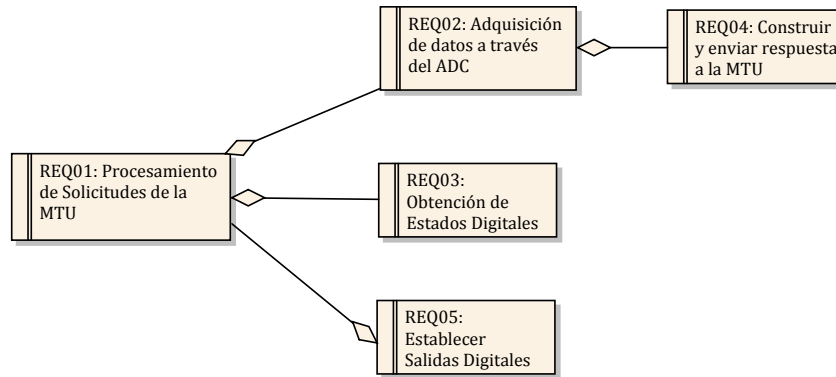
**Tabla 3.9.** Requerimientos formales del *software* de la RTU.

Código	Requerimiento	Descripción
REQ01	Procesamiento de Solicitudes de la MTU	Procesar las solicitudes o ADUs (Protocolo Modbus) que emite la MTU para monitorear el estado de la RTU.
REQ02	Adquisición de datos a través del ADC	Consiste en configurar el ADC para tomar la lectura de los sensores.
REQ03	Obtención de Estados Digitales	Para obtener los estados de las Entradas y Salidas Digitales.
REQ04	Construir y enviar respuesta a la MTU	Consiste en construir una ADU de respuesta a una solicitud emitida por la MTU.
REQ05	Establecer Salidas Digitales	Establecer las Salidas Digitales de acuerdo a los parámetros de las solicitudes de la MTU.

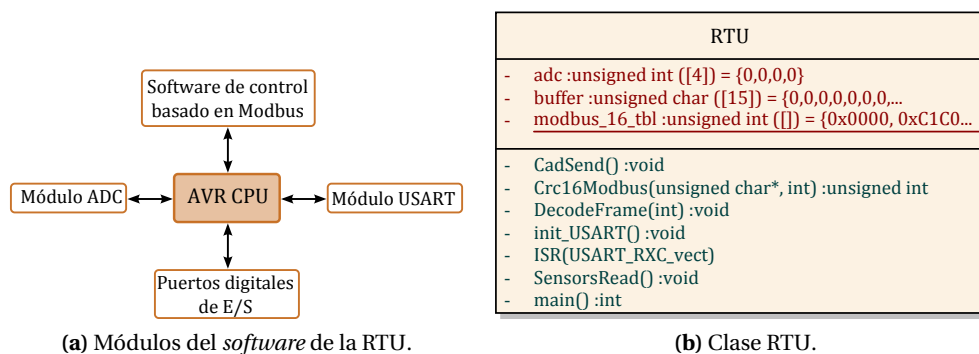
##### 3.3.2.1.2. Clases

El *software* de la RTU consta de cuatro módulos principales (Figura 3.25a): gestión del dispositivo ADC, puertos digitales de Entrada y Salida, gestión del dispositivo USART y el *software* principal de procesamiento de ADUs de solicitud emitidas por la MTU. La estructura *software* de la RTU consiste de una sola clase llamada RTU que se ilustra en la figura 3.25b.

La clase RTU consiste de siete operaciones o métodos que se describen en la tabla 3.10. Estos métodos contribuyen a que la RTU desempeñe de manera modular las actividades programadas.



**Figura 3.24.** Diagrama de requerimientos formales del *software* de la RTU.



**Figura 3.25.** Estructura *software* de la RTU.

### 3.3.2.1.3. Diagramas de actividades

En este apartado se detallan los flujos de trabajo de las actividades y operaciones que el *software* de la RTU debe realizar. Para la RTU se dividen sus actividades *software* en dos partes funcionales: inicialización de módulos y procesamiento de ADUs de solicitud.

La inicialización de módulos consiste en configurar los módulos principales de la RTU (MCU ATmega8) que se muestran en la figura 3.25a. Por esto, se configuran los puertos de entrada y salida<sup>2</sup> y registros internos<sup>3</sup> de configuración [3]. Estas tareas se ejecutan cada vez que se inicia el sistema o cuando ocurre un reset de la RTU.

El diagrama de actividades de la figura 3.26a describe la inicialización de módulos mediante cuatro acciones: inicializar puertos digitales de Entrada y Salida, inicializar el dispositivo USART y habilitar las interrupciones globales para eventos en la USART<sup>4</sup> [3].

<sup>2</sup>Cada módulo del MCU ATmega8 utiliza puertos de entrada y salida para interactuar con dispositivos externos.

<sup>3</sup>Los registros internos se utilizan para configurar el modo de operación de cada módulo: ADC, USART, etc.

<sup>4</sup>Las interrupciones de eventos en la USART permiten ejecutar un fragmento de código cuando un dato está disponible en el búfer de entrada.

Tabla 3.10. Operaciones de la clase RTU.

Código de Operación	Descripción
CadSend	Se encarga de enviar las ADU de respuesta (ya construidas) a la interfaz serie del MCU.
Crc16Modbus	Construye el código de verificación CRC, los parámetros de entrada son: la ADU a enviar (sin CRC) y el tamaño de la misma. Retorna el código CRC en 16 bits en un tipo de dato entero, pero se tomarán solo los valores en codificación Hexadecimal.
DecodeFrame	Decodifica las ADU de entrada y ejecuta las acciones de toma de mediciones, obtención de estados digitales y establecer salidas digitales.
initUSART	Establece la configuración inicial del dispositivo serie (USART) del MCU.
ISR(USART_RXC_vect)	Configura las operaciones que se realizarán al recibir información a través del puerto serie. La ejecución de estas operaciones dependen del disparador por eventos en la recepción de información en el dispositivo USART (Interrupciones).
SensorsRead	Toma muestras de los sensores a través del dispositivo ADC.

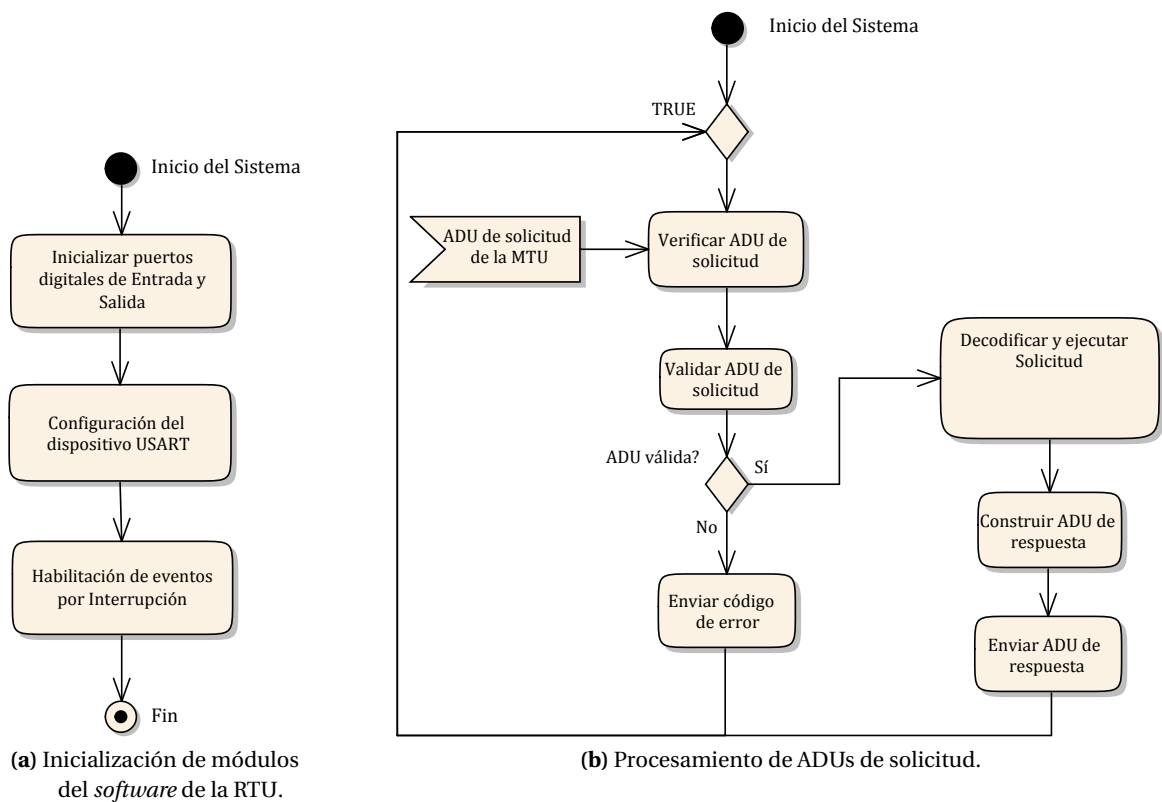


Figura 3.26. Diagramas de actividades del software de la RTU.

La configuración de los dispositivos del MCU ATmega8 se describe en la tabla 3.11 <sup>5</sup> [3].

<sup>5</sup>El Oscilador interno RC es una fuente de reloj que provee las siguientes frecuencias calibradas de operación: 1.0, 2.0, 4.0, y 8.0MHz. La configuración del dispositivo USART aplica en general para la comunicación serie, es decir, para la interfaz RS-232 y RS-485.

**Tabla 3.11.** Configuración de dispositivos del MCU ATmega8.

Dispositivo	Parámetro	Descripción
Oscilador interno RC	$f_o = 4\text{MHz}$	Se configuró para obtener una fuente de reloj de operación de $f_o = 4\text{MHz}$ . Por lo tanto, $f_o$ garantiza satisfacer la demanda en la velocidad de transferencia de la interfaz serie (dispositivo USART).
Dispositivo ADC	$f_{ADC} = f_o/32 = 125\text{kHz}$	La frecuencia de operación del ADC requiere una fuente de reloj de 50 – 200kHz.
Dispositivo USART	Velocidad de transferencia	38400bps
	Tamaño de datos	8 bits
	Paridad	Par
	Bits de paro	Uno
	Modo de Operación	Asíncrono
	Interrupciones	Por recepción de datos

En base a la distribución de terminales del MCU ATmega8 que se muestra en la figura 3.19, la configuración de terminales se describe en la tabla 3.12 [3].

**Tabla 3.12.** Configuración de terminales del MCU ATmega8.

Terminales	Configuración	Descripción
PC0-PC3	Entrada	Terminales usadas como canales de entrada del dispositivo ADC
PD2-PD3	Entrada	Entradas digitales
PD4-PD5	Salida	Salidas digitales
Puerto B (PB0-PB7)	Salida	Registro de salida para generar el voltaje de polarización del sensor de OD a través del dispositivo DAC.
PD0-PD1	Entrada/Salida	Terminales para recepción y transmisión de datos desde y hacia el dispositivo USART.

El procesamiento de ADUs de solicitud define las actividades y el orden en que se deben realizar para atender las solicitudes que emite la MTU. Dichas actividades son: verificar si se ha recibido una ADU a través del dispositivo USART, validar que sea una ADU consistente, decodificar la ADU<sup>6</sup> con el fin de realizar la tarea solicitada y enviar una respuesta, si se trata de una ADU inválida se enviará un código de error. Este flujo de trabajo se ilustra en el diagrama de actividades de la figura 3.26b.

Como se observa en la figura 3.26b, para comprobar que hay una ADU de entrada el *software* entra en un bucle *while*, se reciben datos a través del dispositivo USART<sup>7</sup>, mediante una bandera se indica cuando la ADU ha sido recibida en su totalidad y posteriormente se procesa la ADU. Se comprueba que la ADU sea válida, lo cual se realiza verificando el código CRC y en caso de un error en esta verificación se retorna un código de error. Después, se ejecuta la tarea solicitada<sup>8</sup>, se construye la ADU de respuesta y por último se envía la ADU de respuesta a la MTU a través del dispositivo USART.

<sup>6</sup>Solo si la ADU es válida.

<sup>7</sup>El dispositivo USART emite una interrupción cada vez que se recibe un *byte* de información.

<sup>8</sup>Leer entradas analógicas, obtener el estado de entradas y salidas digitales, establecer salidas digitales.

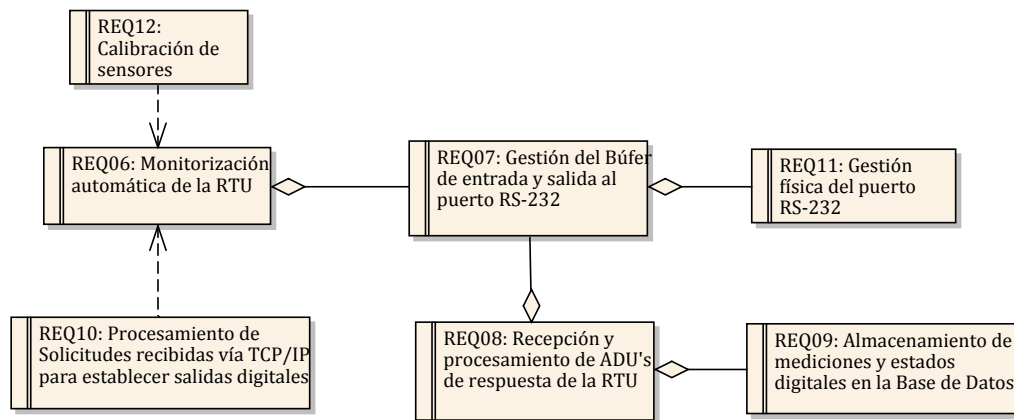
### 3.3.2.2. *software* de la MTU

#### 3.3.2.2.1. Requerimientos formales

Los requerimientos formales de las tareas *software* de la MTU (obtenidos a partir de los requerimientos funcionales de la propuesta) se describen en la tabla 3.13 y el diagrama de requerimientos de la figura 3.27 ilustra la relación entre dichos requerimientos.

**Tabla 3.13.** Requerimientos formales del *software* de la MTU

Código	Requerimiento	Descripción
REQ06	Monitorización automática de la RTU	Realizar las actividades de monitoreo discreto de la RTU a través de ADUs de solicitud, el monitoreo se debe realizar cada determinado tiempo establecido por el usuario
REQ07	Gestión del búfer de entrada y salida al puerto RS-232	Gestionar de manera eficiente el búfer de entrada y salida del puerto RS-232
REQ08	Recepción y procesamiento de ADUs de respuesta de la RTU	Procesar las ADUs que la RTU envía como respuesta ante una solicitud de monitoreo discreto
REQ09	Almacenamiento de mediciones y estados digitales en la Base de Datos	Realizar los cálculos pertinentes en la medición de cada sensor <sup>9</sup> y almacenar los resultados en la base de datos
REQ10	Procesamiento de Solicitudes recibidas vía TCP/IP para establecer salidas digitales	Procesar las solicitudes enviadas a través de la aplicación Web para establecer el estado de las salidas digitales, estas solicitudes son enviadas por el usuario final o Administrador del sistema
REQ11	Gestión física del puerto RS-232	Utilizar una librería para gestionar el uso de la interfaz serie RS-232 en un nivel físico, es decir, que permita configurar la interfaz serie con las especificaciones
REQ12	Calibración de sensores	Calibrar los sensores de pH y OD mediante la toma de mediciones a muestras fijas.

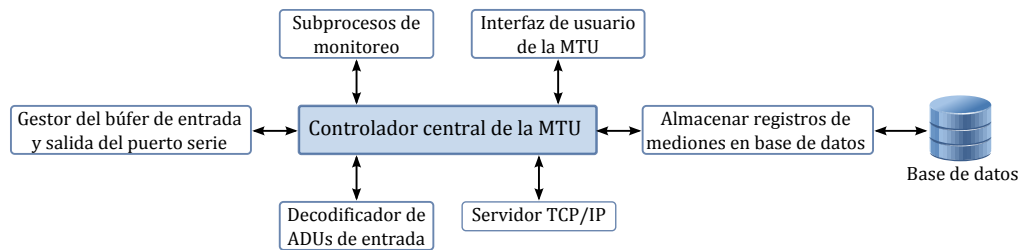


**Figura 3.27.** Diagrama de requerimientos formales del *software* de la MTU.

### 3.3.2.2. Clases

Para gestionar correctamente las entradas y salidas de la interfaz serie (Puerto RS-232), el *software* de la MTU se basa en el *Modelo del Gestor*<sup>10</sup>, el cual es un modelo de control centralizado para sistemas concurrentes [10]. Por ello, para un mejor desempeño, el *software* de la MTU se divide en múltiples tareas que se ejecutan de manera concurrente. Esto se logra mediante la programación de múltiples hilos o subprocesos de ejecución (*Multithreading*) [10].

Basándose en el *Modelo del Gestor* y los requerimientos funcionales, los subprocesos en que se divide el *software* de la MTU son: subprocesos de monitoreo discreto de la RTU, un gestor del búfer de entrada y salida del puerto serie (RS-232), procesador de ADUs de respuesta de la RTU, subproceso para almacenar registros de mediciones y estados digitales en la base de datos, un servidor TCP/IP para recibir solicitudes emitidas desde la aplicación Web y el manejador de eventos de la interfaz gráfica<sup>11</sup>. En la figura 3.28 se ilustra en un diagrama el Modelo del Gestor para el *software* de la MTU.



**Figura 3.28.** División de subprocesos del *software* de la MTU.

Los subprocesos de monitoreo de la RTU implementan las funciones del protocolo Modbus y generan las ADUs de cada función. Por esto, se crea un subproceso para cada función usada en el sistema propuesto. Los subprocesos emiten solicitudes o ADUS de monitoreo cada cierto tiempo determinado (Tiempo de muestreo<sup>12</sup>), por lo que, el resto del tiempo el subproceso pasa a un estado temporalmente fuera de ejecución (*Sleep*).

Los subprocesos requieren acceso al búfer de salida del puerto serie de manera esporádica<sup>13</sup> para enviar las ADUs de solicitud. Por todo lo anterior, se presenta el problema de programación conocido como la relación *productor-consumidor*, en donde el productor genera datos y los almacena en un objeto compartido, y los consumidores leen los datos de dicho objeto.

El objeto compartido se denomina *Búfer Sincronizado*, el cual restringe el acceso a la interfaz serie de salida (búfer de salida) a solo un subproceso de monitoreo (Productor<sup>14</sup>) a la vez y se implementa una cola de espera para que los subprocesos puedan acceder a él de manera

<sup>10</sup>En programación, el Modelo del Gestor consiste en un proceso central que crea, ejecuta y administra un grupo de subprocesos (Thread Pool).

<sup>11</sup>El manejador de eventos es un mecanismo que se genera al crear una interfaz gráfica en Java.

<sup>12</sup>El tiempo de muestreo lo establece el usuario o Administrador.

<sup>13</sup>No se puede determinar el orden en el que los subprocesos cambian su estado a *ejecución*.

<sup>14</sup>En este sistema, el productor es un sistema compuesto que consiste de los subprocesos de monitoreo y las respuestas la RTU.

ordenada. De manera similar, el búfer de entrada restringe al subproceso que decodifica ADUs de respuesta (Consumidor) para que pase a un estado *sleep* hasta que una respuesta este disponible en el búfer [10].

Las clases en que se divide el *software* de la MTU se describen en el apéndice C.1, en donde se muestran los diagramas de clase para ilustrar la relación entre ellas. Los diagramas de clase se dividen en cuatro secciones: monitoreo de entradas analógicas, monitoreo de entradas digitales, diagrama de monitoreo y control de salidas digitales y calibración de sensores.

Cabe mencionar que, para poder usar el puerto serie de la computadora, el *software* de la MTU emplea el *Driver* de la marca *Giovynet*<sup>15</sup>. Y para acceder a la base de datos desde el *software* de la MTU se usa el driver *JDBC* (*Java Database Connectivity*).

### 3.3.2.2.3. Casos de uso

Para satisfacer los requerimientos formales del *software* de la MTU, los casos de uso que se diseñaron de acuerdo a la interacción del Administrador con el *software* para realizar las siguientes actividades<sup>16</sup>: monitoreo principal y calibración de sensores.

El monitoreo principal consiste en iniciar y configurar el *software* de la MTU, el cual de manera automática crea y ejecuta los subprocesos de monitoreo y permite al Administrador configurar el tiempo de muestreo para cada subproceso de monitoreo. Los casos de uso para las actividades de monitoreo principal se detallan en la tabla 3.14 y en la figura 3.29 se ilustra la interacción con el Administrador.

**Tabla 3.14.** Casos de uso del monitoreo principal.

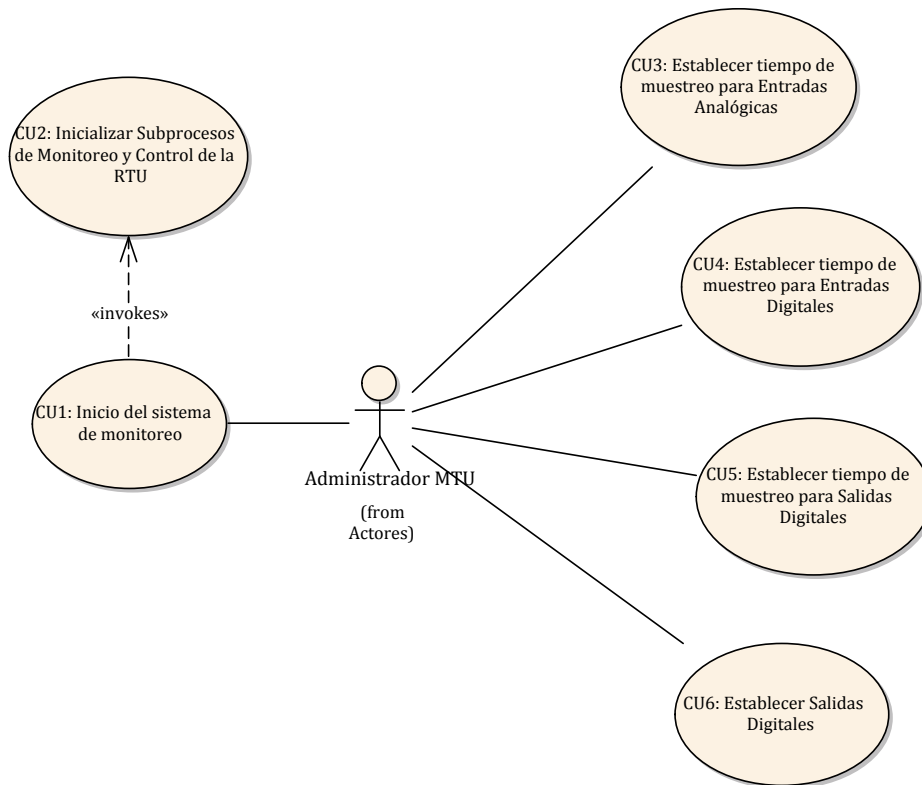
Código	Caso de uso	Descripción
CU1	Inicio del sistema de monitoreo	El Administrador inicia el <i>software</i> de la MTU.
CU2	Inicializar Subprocesos de Monitoreo y Control de la RTU	El proceso central del <i>software</i> de la MTU, en su papel de gestor, crea e inicia los subprocesos de monitoreo.
CU3	Establecer tiempo de muestreo para Entradas Analógicas	El <i>software</i> permite al Administrador cambiar el tiempo de muestreo del subproceso de monitoreo para entradas analógicas.
CU4	Establecer tiempo de muestreo para Entradas Digitales	El <i>software</i> permite al Administrador configurar el tiempo de muestreo para el subproceso de monitoreo para entradas digitales
CU5	Establecer tiempo de muestreo para Salidas Digitales	El Administrador puede cambiar el tiempo de muestreo del subproceso de monitoreo para salidas digitales.
CU6	Establecer Salidas Digitales	Permite al Administrador establecer el estado ( <i>ON-OFF</i> ) de las salidas digitales.

La calibración de sensores define las actividades que el usuario debe realizar con el fin de realizar satisfactoriamente la calibración de los sensores de pH y OD. Para el sensor de pH se requiere como parámetro el valor de la muestra de pH que se va a medir, los valores por defecto establecen

<sup>15</sup>El *Driver Giovynet* es un framework que permite crear aplicaciones Java y comunicar circuitos externos y la computadora. Se puede descargar la versión gratuita de prueba desde <http://www.giovynet.com>.

<sup>16</sup>Actividades del Administrador.





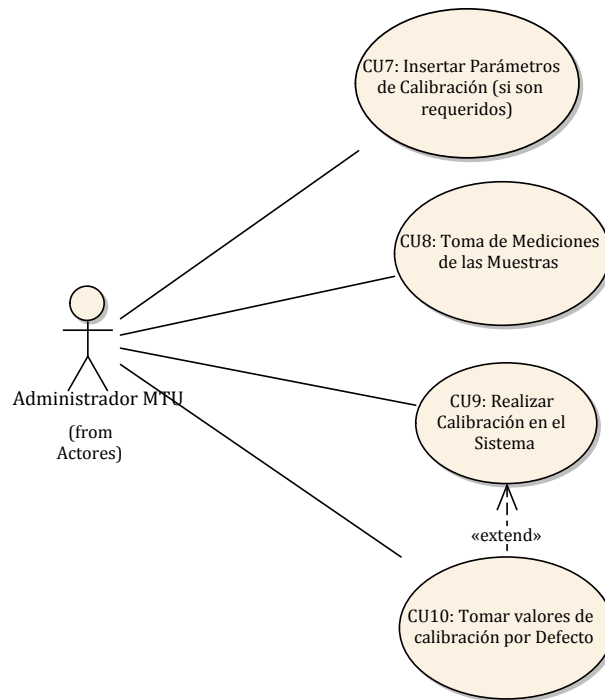
**Figura 3.29.** Casos de uso para monitoreo principal.

la calibración de fábrica (Considerando el comportamiento descrito en la sección 2.3.1) y el sensor de OD requiere la presión barométrica<sup>17</sup>. En la tabla 3.15 se describen los casos de uso y en la figura 3.30 se ilustra la interacción con el Administrador.

**Tabla 3.15.** Casos de uso de la calibración de sensores.

Código	Caso de uso	Descripción
CU7	Insertar parámetros de calibración (si son requeridos)	El usuario puede ingresar los parámetros necesarios para realizar la calibración de un sensor.
CU8	Toma de mediciones de las muestras	El <i>software</i> indicará al subproceso de monitoreo de entradas analógicas que la próxima medición se tomará para calibración.
CU9	Realizar calibración en el sistema	Se realizará la calibración de acuerdo a la medición que se halla tomado.
CU10	Tomar valores de calibración por defecto	El <i>software</i> tomará los valores por defecto o de fábrica de los sensores.

<sup>17</sup>La presión barométrica está en función de la altitud del lugar donde se tomará la medición.



**Figura 3.30.** Casos de uso para la calibración de sensores.

#### 3.3.2.2.4. Diagramas de actividades

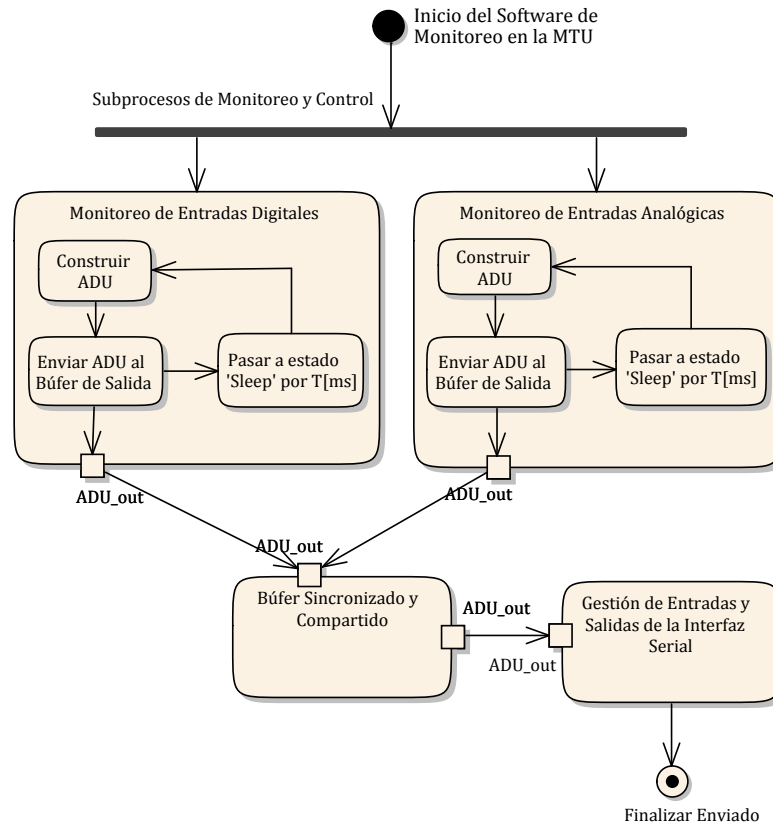
El *software* de la MTU debe realizar una serie de actividades para que los casos de uso sean llevados a cabo de manera satisfactoria. Estas actividades se han dividido en seis secciones: monitoreo automático, gestión de la interfaz serie, procesamiento de ADUs de respuesta (de la RTU), establecer tiempo de muestreo, calibración del sensor de pH y calibración del sensor de OD.

Las actividades de monitoreo automático consisten en que el proceso central crea y ejecuta los subprocesos de monitoreo. Los subprocesos de monitoreo generan la ADU correspondiente, intenta enviar dicha ADU al búfer de la interfaz serie, cuando la ADU ha sido enviada el subproceso pasa al estado *sleep* por el tiempo de muestreo establecido<sup>18</sup> y posteriormente, cuando pasa al estado de ejecución se vuelven a realizar las mismas actividades en un bucle *while*. En la figura 3.31 se ilustra el diagrama de actividades para los subprocesos de monitoreo de entradas analógicas y de entradas digitales<sup>19</sup>.

Para gestionar la interfaz serie, se requieren de dos búfers sincronizados y compartidos, uno de salida y uno de entrada. El búfer de salida acepta ADUs de solicitud de los subprocesos de monitoreo. El subproceso de gestión toma la ADU del búfer, la envía a la interfaz serie y espera un

<sup>18</sup>Los subprocesos de monitoreo tienen un tiempo preestablecido de muestreo, el cual puede ser modificado por el Administrador

<sup>19</sup>No se incluyeron los demás subprocesos, ya que el diagrama sería muy extenso.



**Figura 3.31.** Diagrama de actividades de monitoreo automático.

tiempo para que la RTU emita una respuesta. Esta ADU de respuesta es pasada al búfer de entrada y el subproceso de procesamiento toma dicha ADU para analizar y procesar la respuesta.

El procesamiento de ADUs de respuesta consiste en obtener del búfer de entrada la ADU de respuesta, verificar que la ADU sea válida, si es válida, se genera el registro de medición o estado digital y se almacena en la base de datos y por último se actualiza la información de las mediciones que han realizado en la interfaz gráfica. En caso que no sea una ADU válida se mostrará un mensaje de error en la interfaz gráfica<sup>20</sup>.

La calibración del sensor de pH se puede realizar de dos maneras: tomar valores por defecto (sección 2.3.1) y tomar mediciones de dos muestras de pH fijas. Para calibrar el sensor de OD se requiere la presión barométrica, se puede tomar el valor por defecto, que en este caso es la presión barométrica de la ciudad de Huajuapán de León, Oaxaca. Posteriormente se realiza la medición de una solución saturada de oxígeno y el *software* realizará la calibración correspondiente.

Para las secciones de actividades descritas anteriormente, los diagramas de actividades correspondientes se encuentran en el apéndice C.2.

<sup>20</sup>La interfaz gráfica cuenta con áreas de texto en donde se muestran los flujos de entrada y salida de la interfaz serie

### 3.3.2.3. *software* de la aplicación Web

#### 3.3.2.3.1. Requerimientos formales

Para la aplicación Web se dividen sus tareas *software* en tres áreas principales: gestión de sesión de usuario, presentación de mediciones y estados digitales y gestión de salidas digitales. Estas áreas de tareas *software* se describen en las tablas 3.16, 3.17 y 3.18 respectivamente.

**Tabla 3.16.** Requerimientos formales de la aplicación Web para sesión de usuario.

Código	Requerimiento	Descripción
REQ13	Gestión de Sesión de Usuarios	Ofrecer un entorno de seguridad mediante la gestión de los usuarios que pueden acceder a la aplicación Web
REQ14	Iniciar Sesión	Permitir al usuario iniciar sesión mediante un perfil de usuario
REQ15	Cerrar Sesión	El usuario podrá cerrar su sesión
REQ16	Validar Usuario	Validar los datos del usuario que solicita iniciar sesión

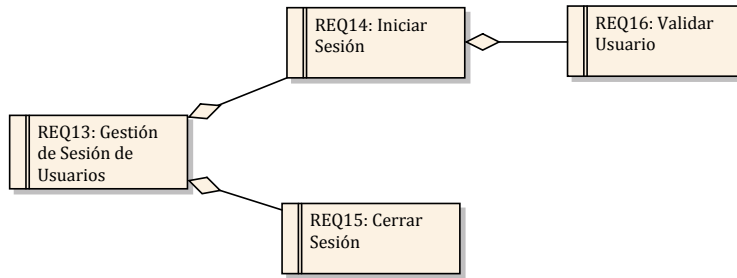
**Tabla 3.17.** Requerimientos formales de la aplicación Web para la presentación de mediciones y estados digitales.

Código	Requerimiento	Descripción
REQ17	Administración de vista principal	Presentar una vista principal en la que se da la bienvenida al sistema
REQ18	Visualizar mediciones de pH	Mostrar al usuario las mediciones de pH
REQ19	Visualizar Estados Digitales	Informar al usuario el estado de las entradas y salidas digitales
REQ20	Visualizar mediciones de Oxígeno Disuelto	Presentar al usuario las mediciones de OD
REQ21	Visualizar mediciones de Temperatura	Mostrar las mediciones de Temperatura

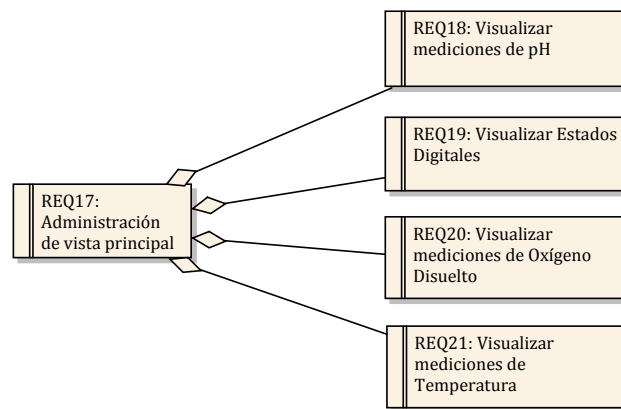
**Tabla 3.18.** Requerimientos formales de la aplicación Web para gestionar las salidas digitales.

Código	Requerimiento	Descripción
REQ22	Administración de Salidas Digitales	Se presenta el formulario para establecer las salidas digitales
REQ23	Enviar solicitud a la MTU	Enviar solicitud a la MTU para establecer las salidas digitales

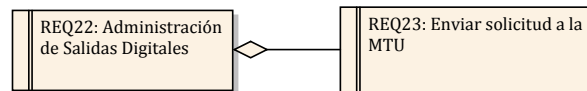
En las figuras 3.32, 3.33 y 3.34 se ilustran las áreas de requerimientos formales mediante diagramas de requerimientos.



**Figura 3.32.** Diagrama de requerimientos formales de la aplicación Web para sesión de usuario.



**Figura 3.33.** Diagrama de requerimientos formales de la aplicación Web para la presentación de mediciones y estados digitales.



**Figura 3.34.** Diagrama de requerimientos formales de la aplicación Web para gestionar las salidas digitales.

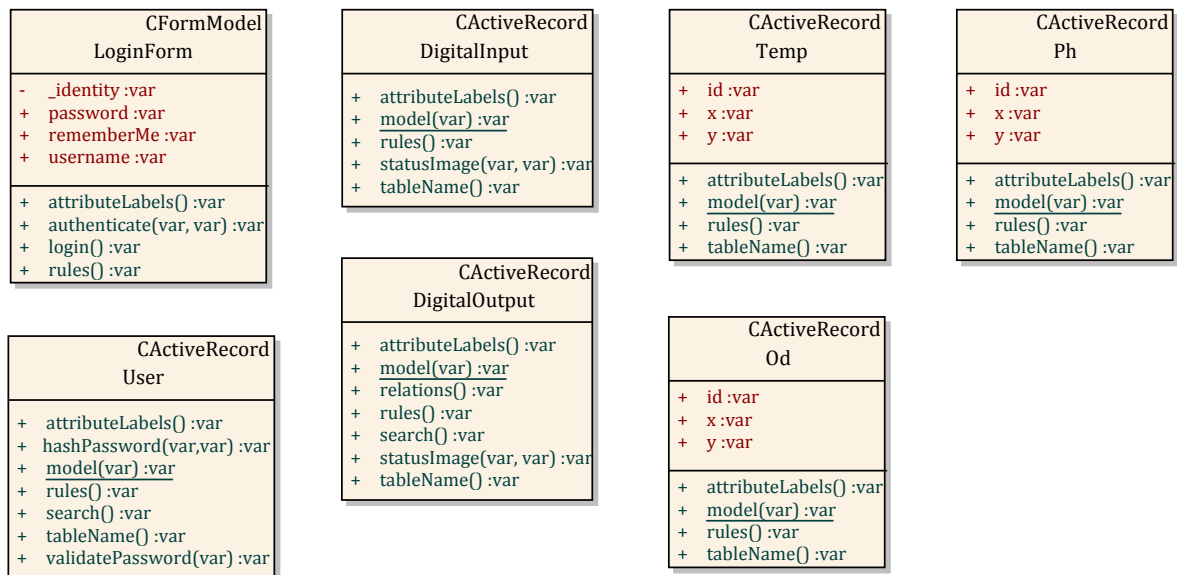
### 3.3.2.3.2. Clases

Siguiendo el paradigma MVC y la dinámica de desarrollo del framework *Yii*, las clases de la aplicación Web se dividen en dos partes principales: modelos y controladores<sup>21</sup>.

Basándose en los requerimientos formales, las clases modelo de la aplicación Web son las siguientes: User, LoginForm, DigitalInput, DigitalOutput, Temp, Od y Ph. Estas clases modelo se describen en la tabla 3.19 y se ilustran en la figura 3.35.

**Tabla 3.19.** Descripción de clases de modelos (MVC) de la aplicación Web.

Clase	Descripción
User	Clase modelo para abstraer usuarios registrados.
LoginForm	Sirve para generar de manera automática el formato del formulario para inicio de sesión.
DigitalInput	Modelo para gestionar las entradas digitales.
DigitalOutput	Se utiliza para gestionar las salidas digitales.
Temp	Esta clase se utiliza para abstraer las mediciones de temperaturas.
Od	Mediante esta clase las mediciones de OD son abstraídas de la base de datos.
Ph	Permite abstraer las mediciones de pH.



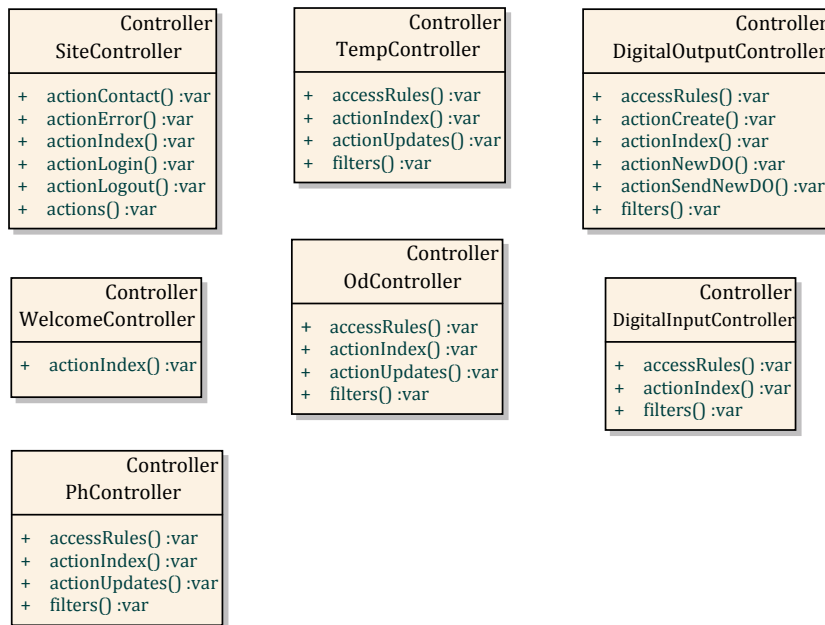
**Figura 3.35.** Clases de modelos (MVC) de la aplicación Web.

Las clases de los controladores de la aplicación Web son las siguientes: SiteController, WelcomeController, PhController, TempController, OdController, DigitalOutputController, DigitalInputController. Estas clases de controladores se describen en la tabla 3.20 y se ilustran en la figura 3.36.

<sup>21</sup>Las vistas están relacionadas con las acciones dentro de los controladores.

**Tabla 3.20.** Descripción de clases de controladores (MVC) de la aplicación Web.

Clase	Descripción
SiteController	Controlador básico que provee el framework Yii para asistir el desarrollo.
WelcomeController	Gestiona las rutas de bienvenida al sistema.
PhController	Provee las funcionalidades que permiten visualizar las mediciones de pH.
TempController	Permite visualizar las mediciones de temperatura.
OdController	Se utiliza para mostrar las mediciones de OD.
DigitalOutputController	Este controlador permite visualizar las salidas digitales y gestionar las salidas digitales.
DigitalInputController	Permite visualizar las entradas digitales.

**Figura 3.36.** Clases de controladores (MVC) de la aplicación Web.

### 3.3.2.3.3. Modelo entidad-relación

El modelo o diagrama entidad-relación (E-R, *Entity Relationship*) permite representar las entidades de las clases de modelo definidas en la sección anterior [7, 35]. En este modelo se muestran las propiedades de cada entidad y se usa básicamente en el diseño de la base de datos<sup>22</sup> MySQL. En la figura 3.37 se ilustra el modelo E-R para la aplicación Web.

### 3.3.2.3.4. Casos de uso

Basándose en los requerimientos formales, los casos de uso para la aplicación Web se dividen en dos áreas: sesión de usuarios y funcionalidades Web. Estas áreas de casos de uso se describen en las tablas 3.21 y 3.22 respectivamente

<sup>22</sup>Bajo este modelo, el *software* de la MTU almacena los registros de mediciones y estados digitales.

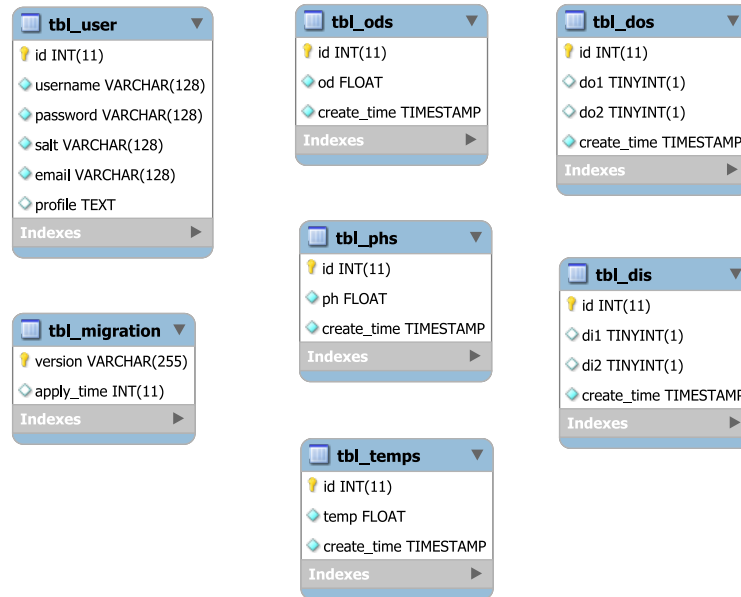


Figura 3.37. Modelo E-R de la aplicación Web.

Tabla 3.21. Casos de uso para sesión de usuarios.

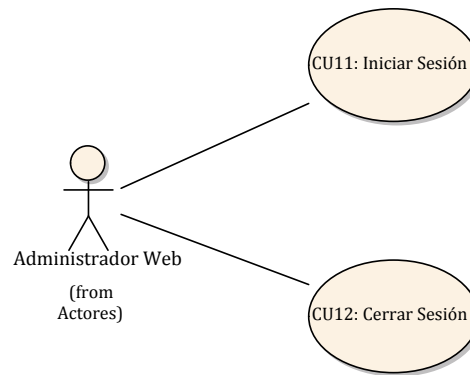
Código	Caso de uso	Descripción
CU11	Iniciar Sesión	El usuario deberá iniciar sesión en la aplicación Web para acceder a las funcionalidades que este ofrece.
CU12	Cerrar Sesión	De igual manera que el caso de uso anterior, el usuario debe cerrar sesión para salir de la aplicación.

Tabla 3.22. Casos de uso para funcionalidades Web.

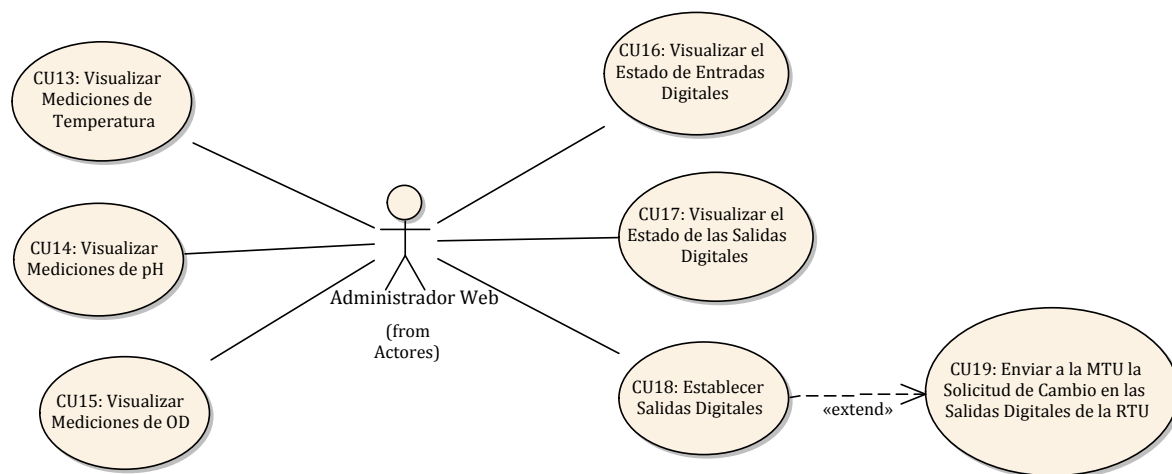
Código	Caso de uso	Descripción
CU13	Visualizar Mediciones de Temperatura	Mostrar los registros de mediciones de temperatura
CU14	Visualizar Mediciones de pH	Permite mostrar las mediciones de pH
CU15	Visualizar Mediciones de OD	El usuario podrá visualizar las mediciones de OD
CU16	Visualizar el Estado de Entradas Digitales	El usuario podrá observar el estado de las entradas digitales
CU17	Visualizar el Estado de las Salidas Digitales	Se mostrará el estado de las salidas digitales
CU18	Establecer Salidas Digitales	Se presentará el formulario para establecer el estado de las salidas digitales.
CU19	Enviar a la MTU la Solicitud de Cambio en las Salidas Digitales de la RTU	Se enviará una ADU de solicitud al <i>software</i> de la MTU para cambiar el estado de las salidas digitales en la RTU.

En las figuras 3.38 y 3.39 se ilustran los casos de uso para sesión de usuarios y funcionalidades Web respectivamente a través de diagramas de casos de uso.





**Figura 3.38.** Diagrama de casos de uso para sesión de usuarios.



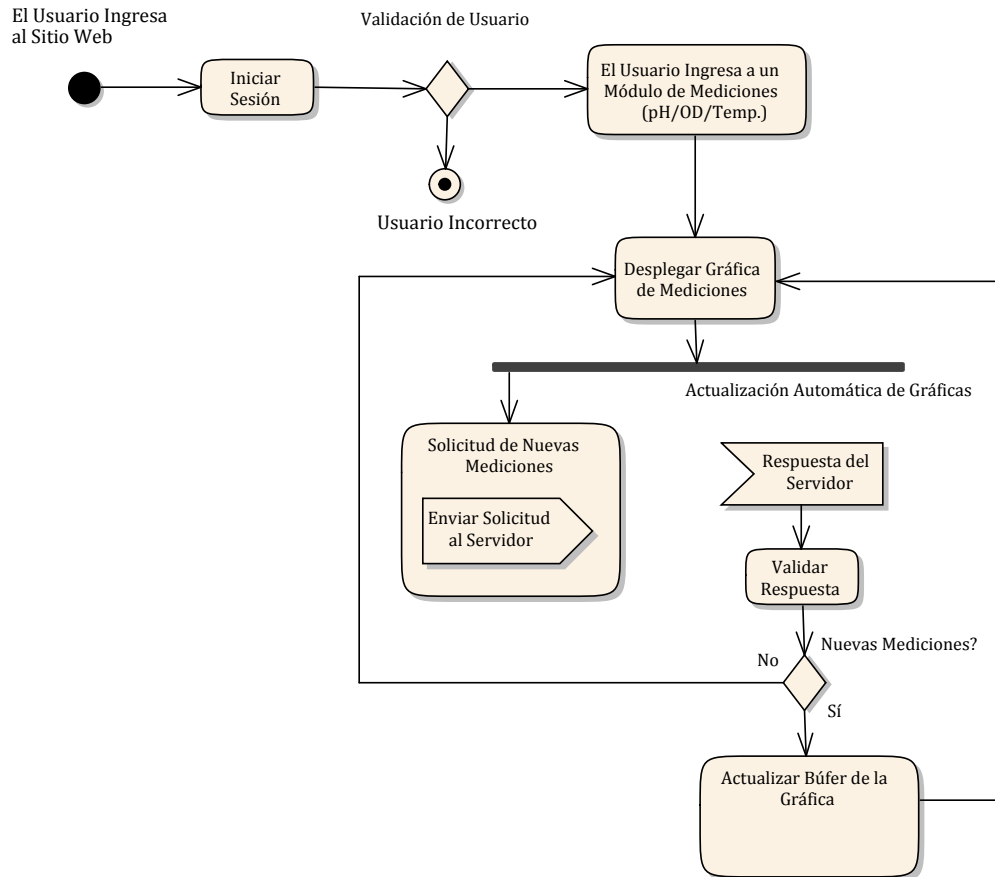
**Figura 3.39.** Diagrama de casos de uso para funcionalidades Web.

### 3.3.2.3.5. Diagramas de actividades

A partir de los casos de uso descritos anteriormente, las actividades que la aplicación Web debe realizar básicamente se dividen en dos grupos: presentar información dinámica de los registros de mediciones y estados digitales y gestionar las salidas digitales.

La presentación de las mediciones se efectúa empleando gráficas dinámicas y tablas. Los estados digitales se muestran de manera tabular. Las gráficas dinámicas se implementan usando las extensiones de Yii: *Highcharts*, *YiiBooster* y *JQuery (JavaScript)*. Se requiere el uso de solicitudes AJAX para actualizar las gráficas de manera transparente al usuario. Las actividades para ingresar a los módulos de visualización de entradas analógicas<sup>23</sup> y la actualización dinámica de gráficas se describen en el diagrama de actividades de la figura 3.40.

<sup>23</sup>Mediciones de pH, OD y temperatura.



**Figura 3.40.** Diagrama de actividades para la presentación de mediciones analógicas.

Las actividades para establecer las salidas digitales consisten básicamente en introducir los estados deseados en el formulario proporcionado por la aplicación Web. Este flujo de trabajo se ilustra en la figura 3.41.

### 3.3.3. Integración de elementos *hardware* y *software*

La integración de elementos *hardware* y *software* consiste en interconectar cada una de las etapas *hardware* y *software* que se diseñaron en las secciones 3.3.1 y 3.3.2. El objetivo principal de la integración es construir el sistema propuesto a partir de los elementos en que fue dividido dicho sistema. En la figura 3.42 se ilustra de manera simplificada las etapas que se van a integrar.

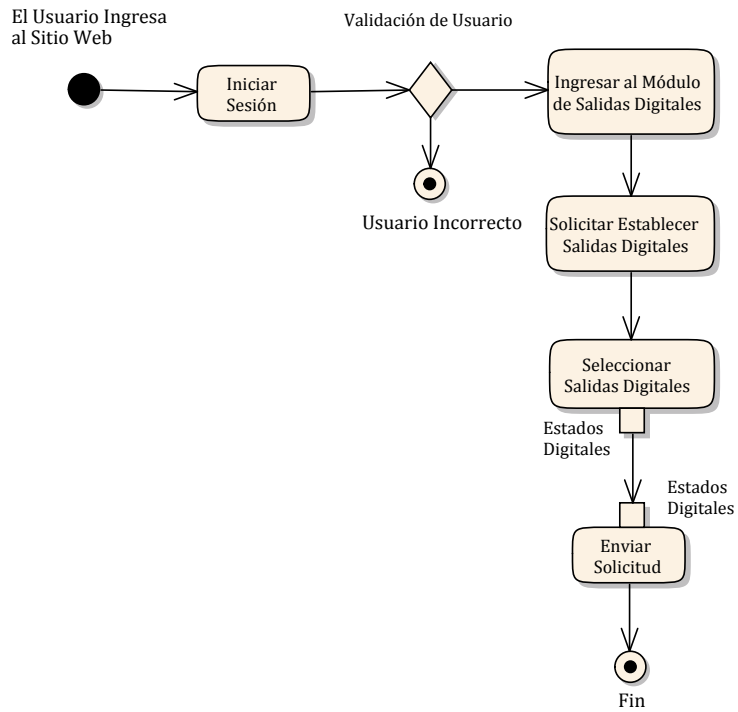


Figura 3.41. Diagrama de actividades para establecer salidas digitales.

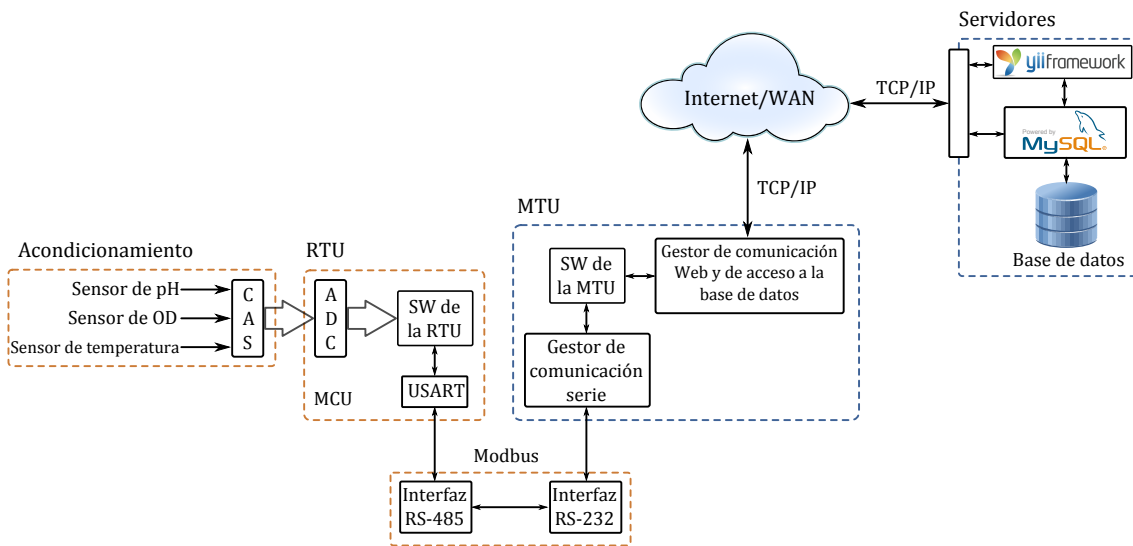


Figura 3.42. Integración del sistema SCADA propuesto.

### 3.3.3.1. Diseño del PCB de los módulos *hardware*

Para una mejor integración, los elementos hardware se organizaron en tres módulos principales. El primer módulo consiste del dispositivo CAS para el sensor de pH, este módulo toma la fuente de voltaje de alimentación ( $\pm 12V$ ) y mediante un regulador de voltaje de 5V (CI 7805) proporciona la alimentación para otros dispositivos que lo requieren.

El segundo módulo consta del dispositivo CAS para el sensor de OD y el CAS para el sensor de temperatura. Y, el tercer módulo consiste del MCU ATmega8 (como RTU), el dispositivo de comunicación serie RS-485 y el sistema generador de voltaje para polarizar el sensor de OD.

Los módulos se ordenan, de la manera en que se describe anteriormente, para lograr que los PCB sean aproximadamente del mismo tamaño y se pueda diseñar un gabinete en donde sean montados dichos módulos. El tamaño promedio de cada módulo es de 4.75cm y 10.75cm de ancho y largo respectivamente.

#### 3.3.3.1.1. Diseño del PCB para el CAS del sensor de pH

La fuente de alimentación principal para la RTU consiste en una fuente de voltaje dual de  $\pm 12V$  para alimentar correctamente los op-amps en los dispositivos CAS<sup>24</sup>. El dispositivo para el CAS del sensor de pH cuenta con un regulador de voltaje de 5V (CI 7805), el cual proporciona el voltaje fuente para otros elementos como: el dispositivo que genera el voltaje  $V_{offset}$  para el sensor de pH, para el sensor de temperatura y los dispositivos digitales como el MCU ATmega8 y el dispositivo MAX489 para comunicación. El diseño PCB para el CAS del sensor de pH se muestra en la figura 3.43.

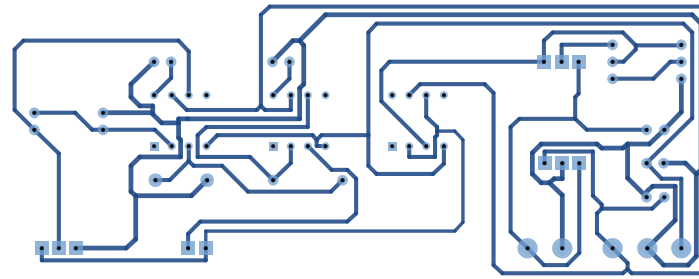
Como se observa en la figura 3.43, el módulo cuenta con un conector de entradas para el sensor de pH, en donde se conectan las terminales positiva y negativa, sin embargo, las salidas del sensor de pH están integradas en un conector tipo BNC hembra, por lo que se debe adaptar un conector BNC macho para poder obtener las salidas positiva y negativa.

Este módulo también cuenta con un conector de salidas, el cual proporciona las siguientes salidas: GND<sup>25</sup> o tierra, el voltaje  $V_{offset}$  y la salida del CAS de pH  $V_{CASpH}$ . Estas salidas se conectan al conector de entradas correspondiente en el módulo del MCU (Sección 3.3.3.1.3).

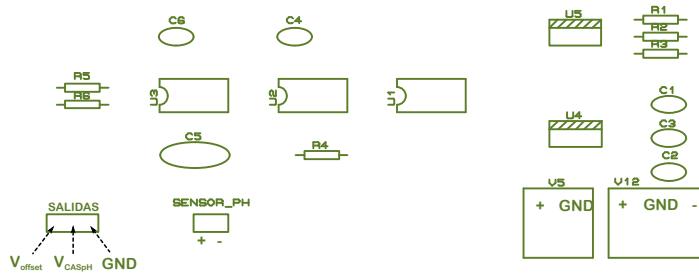
La lista de los materiales empleados en este diseño se describe en la tabla 3.23 y en la figura 3.44 se ilustra una vista en 3D del dispositivo.

<sup>24</sup>Las fuentes de alimentación principal son compartidas a través de los bloques de terminales V12 y V5 de cada módulo hardware.

<sup>25</sup>Es muy importante compartir la tierra en la salida de los dispositivos CAS, ya que esta práctica previene la generación de ruido en las señales de salida.



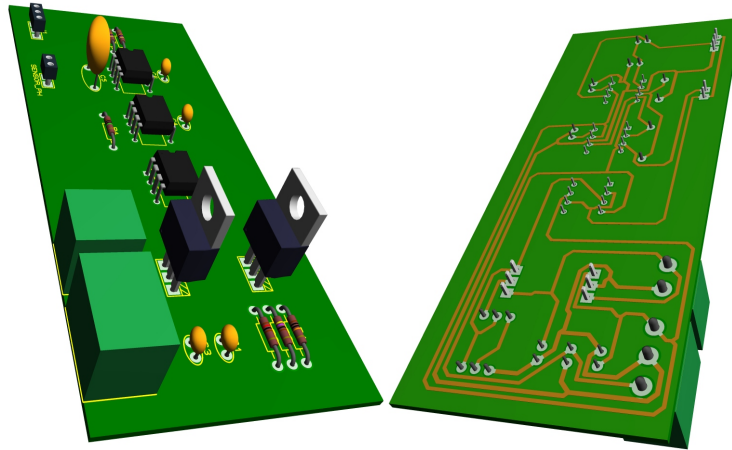
(a) Diseño de pistas PCB.



(b) Ordenamiento de CIs.

**Figura 3.43.** PCB del CAS para el sensor de pH, las imágenes no están a escala.**Tabla 3.23.** Descripción de materiales empleados en el diseño PCB del CAS para el sensor pH.

Cantidad	Referencia	Valor
<b>Resistores</b>		
1	R1	100k $\Omega$
1	R2	4.7k $\Omega$
1	R3	1.2k $\Omega$
1	R4	10M $\Omega$
2	R5,R6	470k $\Omega$
<b>Capacitores</b>		
5	C1-C4,C6	1nF
1	C5	100nF (Poliéster)
<b>Circuitos Integrados (CIs)</b>		
1	U1	TL081
2	U2,U3	CA3140
1	U4	7805
1	U5	LM317T
<b>De propósito general</b>		
1	SALIDAS	Conector macho de 3 terminales
1	SENSOR_PH	Conector macho de 2 terminales
1	V12	Bloque de 3 terminales
1	V5	Bloque de 2 terminales



**Figura 3.44.** Vista 3D del dispositivo CAS para el sensor de pH.

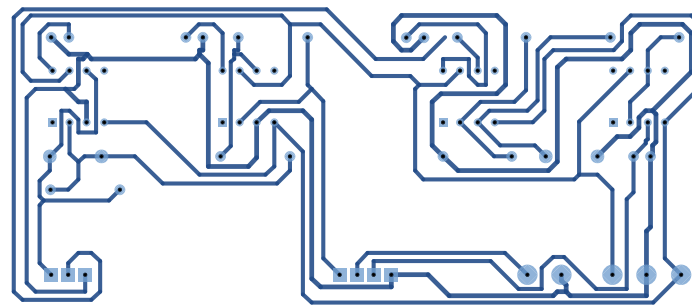
### 3.3.3.1.2. Diseño PCB para el CAS del sensor de OD y temperatura

Debido al tamaño de los dispositivos CAS del sensor de OD y temperatura, ambos dispositivos se incluyen en un mismo PCB, dichos dispositivos se integran como se muestra en la figura 3.45. Como se observa, el conector de entradas proporciona el voltaje de alimentación  $V_S$  para el sensor de temperatura, la entrada  $V_{in}$  es la salida de dicho sensor y la entrada  $I_{OD}$  es la corriente del sensor de OD. Las salidas proporcionan los voltajes de los dispositivos CAS:  $V_{CAS_{Temp}}$  y  $V_{CAS_{OD}}$ .

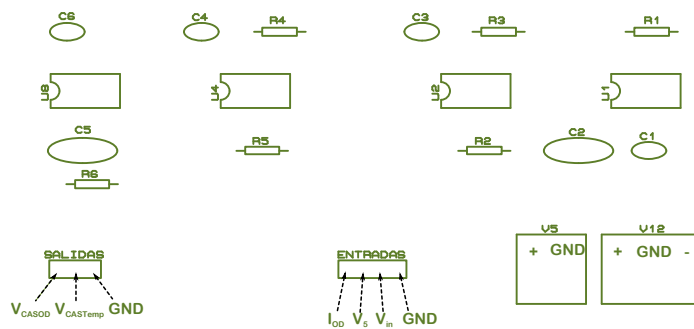
Los materiales empleados en este diseño se describen en la tabla 3.24 y en la figura 3.46 se ilustra una vista en 3D del dispositivo.

**Tabla 3.24.** Descripción de materiales del diseño PCB para el CAS del sensor de OD y temperatura.

Cantidad	Referencia	Valor
<b>Resistores</b>		
3	R1,R5,R6	10M $\Omega$
1	R2	15k $\Omega$
1	R3	47k $\Omega$
1	R4	1M $\Omega$
<b>Capacitores</b>		
1	C1	100nF
2	C2,C5	100nF (Poliéster)
3	C3,C4,C6	1nF
<b>Circuitos Integrados (CIs)</b>		
1	U1	TL081
3	U2,U4,U8	CA3140
<b>De propósito general</b>		
1	ENTRADAS	Conector macho de 4 terminales
1	SALIDAS	Conector macho de 3 terminales
1	V12	Bloque de 3 terminales
1	V5	Bloque de 2 terminales

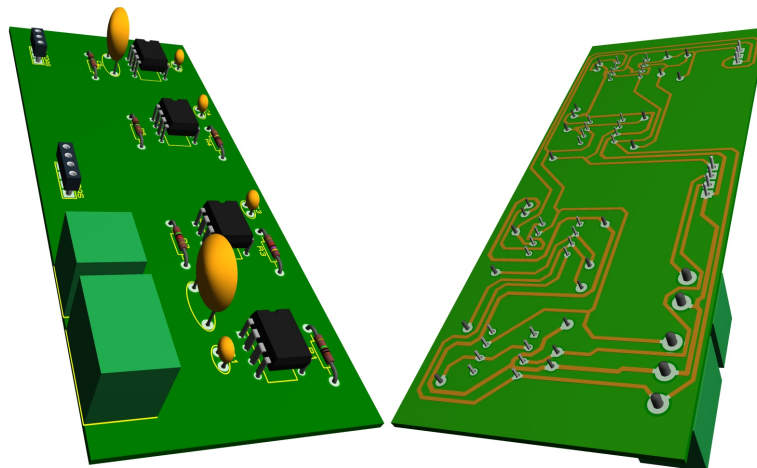


(a) Diseño de las pistas PCB.



(b) Distribución de CIs.

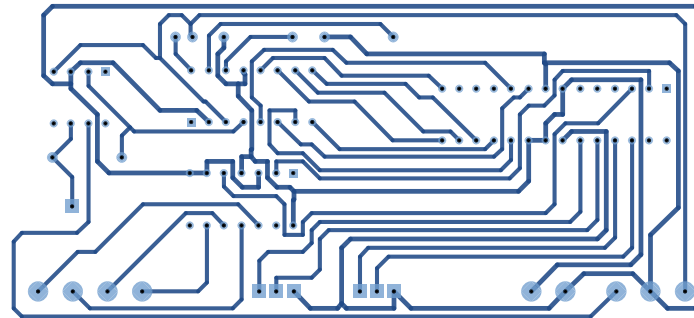
**Figura 3.45.** PCB del CAS para el sensor de OD y temperatura, las imágenes no están a escala.



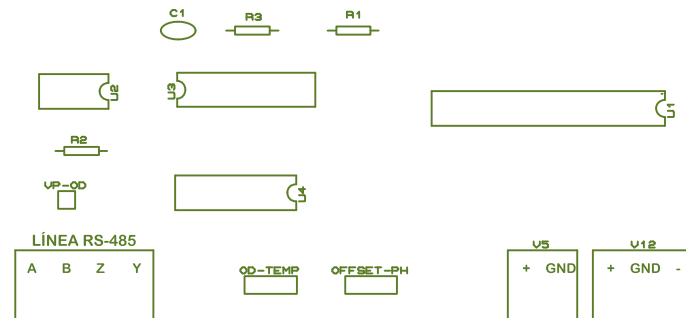
**Figura 3.46.** Vista 3D del dispositivo CAS para el sensor de OD y temperatura.

### 3.3.3.1.3. Diseño del PCB para el módulo del MCU

El PCB de este módulo consiste de tres partes principales: el MCU que desempeña el papel de RTU, el módulo para generar el voltaje de polarización del sensor de OD con el CI DAC0808 y el módulo de comunicación serie RS-485 empleando el CI MAX489. Este diseño PCB se muestra en la figura 3.47.



(a) Diseño de las pistas PCB.



(b) Distribución de CIs.

**Figura 3.47.** PCB para el módulo del MCU, las imágenes no están a escala.

Como se observa en la figura 3.47, el módulo del MCU cuenta con dos conectores de entrada, uno para las salidas del módulo del dispositivo CAS del sensor de pH y otro para el módulo del CAS del sensor de OD y temperatura<sup>26</sup>. Estas entradas están conectadas directamente a las terminales del ADC del MCU ATmega8<sup>27</sup> en el siguiente orden: la entrada  $V_{CASOD}$  se conecta a la terminal ADC0,  $V_{CASTemp}$  a la terminal ADC1,  $V_{offset}$  se conecta a la terminal ADC2 y  $V_{CASpH}$  se conecta a la terminal ADC3. Por último, la terminal  $V_{POD}$  proporciona el voltaje de polarización del sensor de OD.

La lista de materiales empleados en este diseño se describe en la tabla 3.25 y en la figura 3.46 se muestra una vista en 3D del módulo del MCU.

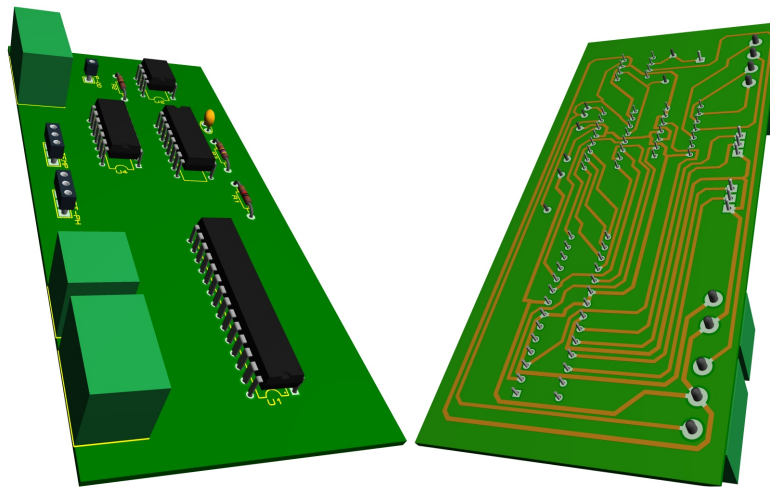
<sup>26</sup>Para evitar problemas de conexión, el orden de las terminales es el mismo en los conectores de salida de los módulos CAS y los conectores de entrada en el módulo del MCU.

<sup>27</sup>La distribución de terminales del MCU ATmega8 se describe en la sección 3.3.1.2.



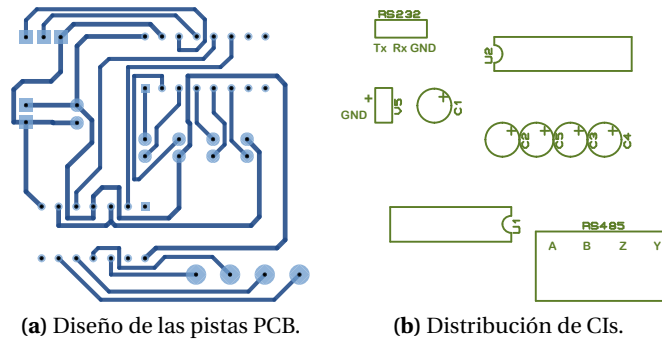
**Tabla 3.25.** Descripción de materiales empleados en el diseño PCB de la RTU.

Cantidad	Referencia	Valor
<i>Resistores</i>		
1	R1	10k $\Omega$
1	R2	100k $\Omega$
1	R3	470k $\Omega$
<i>Capacitores</i>		
1	C1	100nF
<i>Circuitos Integrados (CIs)</i>		
1	U1	ATMEGA8
1	U2	TL081
1	U3	DAC0808
1	U4	MAX489
<i>De propósito general</i>		
1	LÍNEA RS-485	Conector macho de 4 terminales
2	OD-TEMP, OFFSET-PH	Conector macho de 3 terminales
1	V12	Bloque de 3 terminales
1	V5	Bloque de 2 terminales
1	VP-OD	Conector macho de 1 terminal

**Figura 3.48.** Vista 3D del módulo del MCU.

### 3.3.3.1.4. Diseño del PCB para el transceptor RS-232/RS-485

El diseño del PCB para el transceptor RS-232/RS-485 se muestra en la figura 3.49. Este módulo cuenta con el dispositivo de comunicación con la interfaz RS-232<sup>28</sup> empleando el CI MAX232 y para la interfaz serie RS-485 se usa el CI MAX489.



**Figura 3.49.** PCB del transceptor RS-232/RS-485, las imágenes no están a escala.

Debido a que el transceptor RS-232/RS-485 estará a una distancia relativamente grande de la RTU, este módulo requiere una fuente de alimentación diferente a la fuente de alimentación de la RTU. Por ello, se incluye un conector (V5) para dicha fuente de alimentación. La conexión cableada RS-485 entre este módulo y el módulo del MCU requiere la conexión entre las terminales de los conectores RS-485 de cada módulo, la configuración de dicha conexión se detalla en los apartados 3.3.1.3 y 3.3.1.4.

Los materiales empleados para este diseño se describen en la tabla 3.26 y en la figura 3.50 se muestra una vista en 3D del transceptor RS-232/RS-485.

**Tabla 3.26.** Descripción de materiales empleados en el diseño PCB de la RTU.

Cantidad	Referencia	Valor
<i>Capacitores</i>		
5	C1-C5	1 $\mu$ F
<i>Circuitos Integrados (CIs)</i>		
1	U1	MAX489
1	U2	MAX232
<i>De propósito general</i>		
1	RS485	Bloque de 4 terminales
1	RS232	Conector macho de 3 terminales
1	V5	Conector macho de 2 terminales

<sup>28</sup>La interfaz RS-232 es la interfaz serie con la que cuenta la MTU.

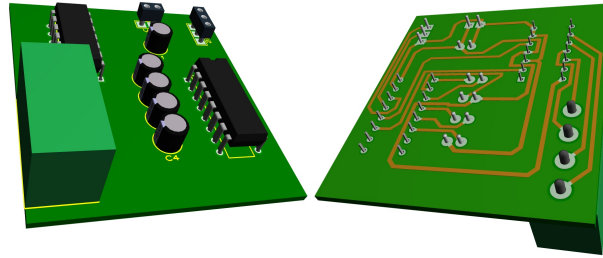


Figura 3.50. Vista 3D del Transceptor RS-232/RS-485.

### 3.3.3.2. Diseño de interfaces gráficas

En las figuras 3.51 se muestra la ventana principal de la GUI del *software* de la MTU, esta interfaz fue desarrollada empleando el IDE *NetBeans*. En la figura 3.52 se muestra la vista de la ruta principal de la aplicación Web, la cual solo muestra una vista de bienvenida a la aplicación Web. Para acceder a la aplicación Web se requiere el uso de navegadores Web como *Chrome*, *Mozilla*, *Internet Explorer*, *Opera*, entre otros.

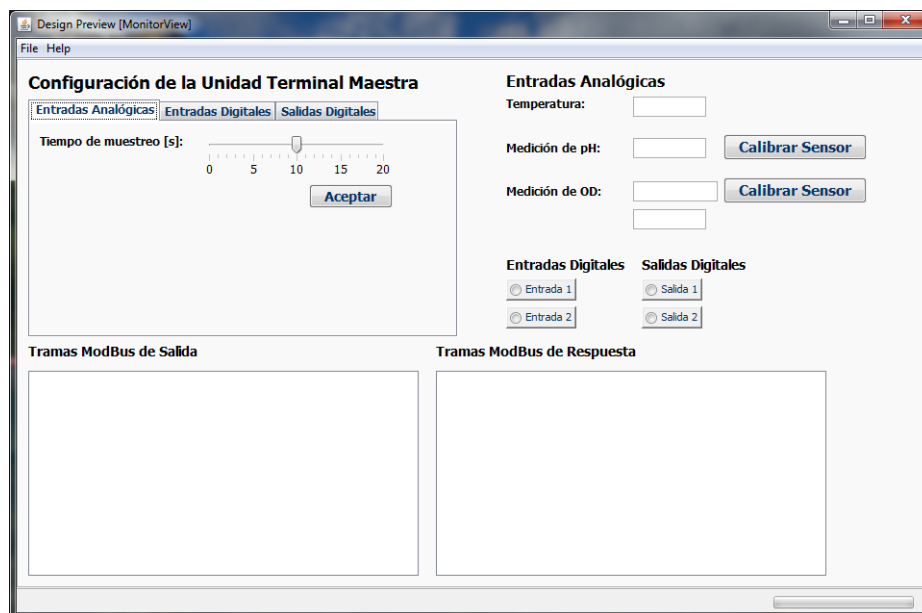


Figura 3.51. Ventana principal de la GUI del *software* de la MTU.

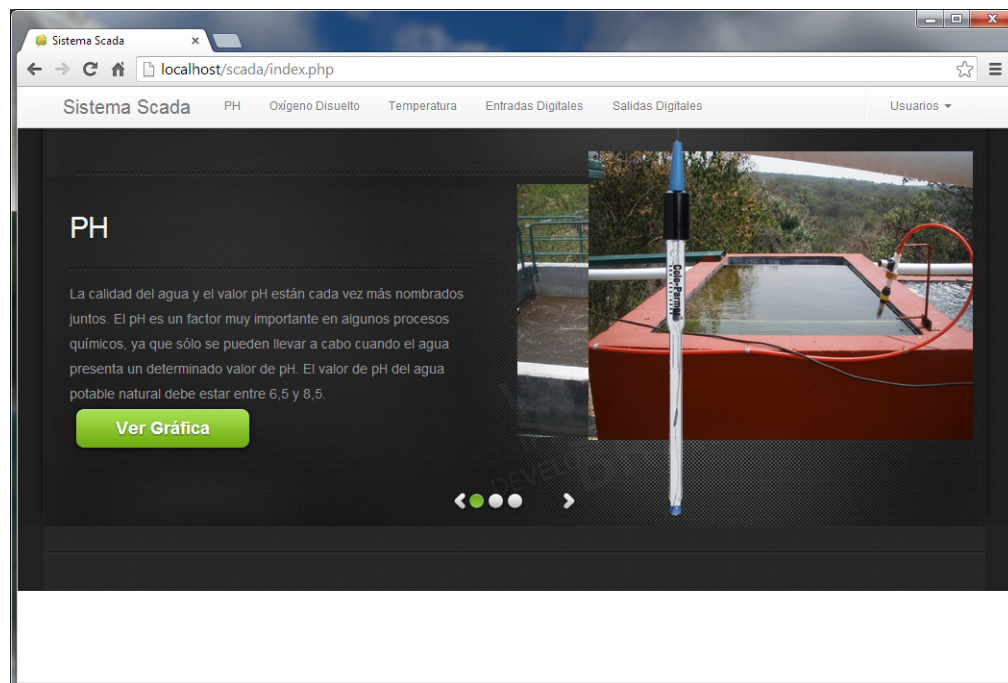


Figura 3.52. Vista principal de la aplicación Web.

# 4

## Verificación y Validación del Producto Final: Pruebas y Resultados

En este apartado se presenta la verificación o comprobación de la hipótesis inicialmente planteada y la validación de que el sistema cumple con los requerimientos funcionales de la propuesta.

La verificación consiste en realizar pruebas de funcionalidad e integridad que validan el cumplimiento de los requerimientos funcionales. Para abarcar todos los casos de prueba posibles, las pruebas se dividen en tres etapas: pruebas de caja blanca, pruebas de caja gris y pruebas de caja negra.

Las pruebas de caja blanca (*White Box Testing*) consisten en probar más allá de la interfaz de usuario, es decir, las pruebas se efectúan en los aspectos esenciales del sistema. El sistema es visualizado como una caja transparente en donde claramente se observan dichos aspectos [1, 18].

Generalmente, las pruebas de caja gris (*Gray Box Testing*) son orientadas a pruebas de funcionalidad o dominios de intercambio (*Business Domain*) [1, 18]. Básicamente, las pruebas de funcionalidad son efectuadas para verificar la interacciones entre sistemas [36].

La técnica de pruebas de caja negra (*Black Box Testing*) se emplea (de manera superficial) en pruebas de funcionalidad sin observar la estructura interna [1, 18]. Este tipo de pruebas están basados totalmente en los requerimientos y la especificación del sistema [36].

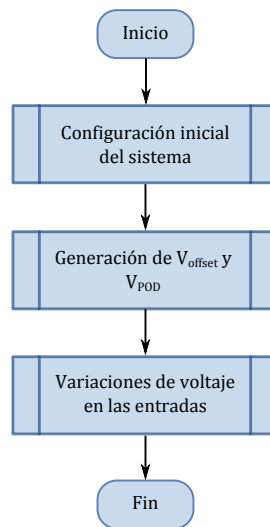
Dicho lo anterior, las pruebas de caja blanca se enfocan específicamente en el funcionamiento de los elementos o módulos *hardware*, mientras que las pruebas de caja gris toman un enfoque general, es decir, el funcionamiento del sistema es sometido a pruebas en su totalidad. Las pruebas de caja negra están orientadas para realizar pruebas de funcionamiento en algunos casos

de uso de la aplicación Web. El plan de pruebas destinado a las tres etapas descritas anteriormente se detallan en las siguientes secciones.

## 4.1. Pruebas de Caja Blanca

Las pruebas de caja blanca consisten en verificar que las conexiones entre módulos *hardware* sean realizadas satisfactoriamente, es decir, que no se presenten problemas de cortocircuito, que no existan problemas de compatibilidad de impedancias de entrada y salida<sup>1</sup> y que las salidas de los módulos *hardware* sean consistentes con el diseño (Apartado 3.3.1).

Las pruebas de caja blanca se llevan a cabo siguiendo el diagrama de actividades de la figura 4.1. Estas pruebas se dividen en tres secciones: configuración inicial del sistema, generación de  $V_{offset}$  y  $V_{POD}$  y por último variaciones de voltaje en las entradas.



**Figura 4.1.** Diagrama de actividades para las pruebas de caja blanca.

En general, los instrumentos de medición empleados para realizar las pruebas son los siguientes: multímetro marca *FLUKE* modelo 189 (Figura 4.2a) y multímetro marca *BK PRECISION* modelo 2831E (Figura 4.2b). La fuente de voltaje utilizada es de la marca *MATRIX* modelo *MPS-3005L-3* (Figura 4.5), la cual cuenta con dos salidas de voltaje variable y una fuente de voltaje con una salida fija de 5V.

### 4.1.1. Prueba de configuración inicial del sistema

La prueba de configuración inicial consiste en que los módulos *hardware* (Sección 3.3.3.1) sean conectados entre sí para formar la RTU. Para un mejor manejo de la RTU, ésta se monta en un gabinete construido a base de acrílico (Figura 4.3).

<sup>1</sup>Estos problemas se presentan frecuentemente en las entradas y salidas de los dispositivos op-amp y en las salidas de los sensores.



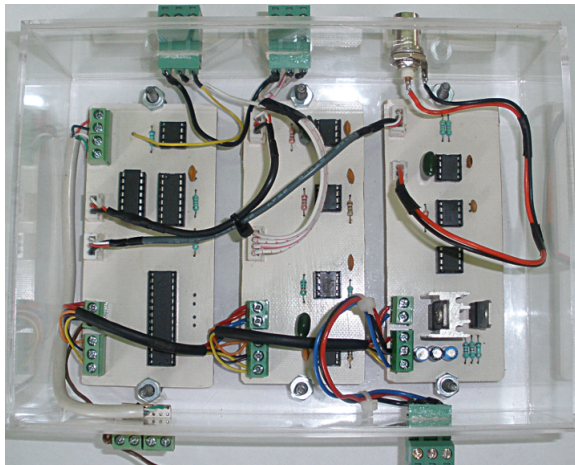
(a) Multímetro  
FLUKE 189.



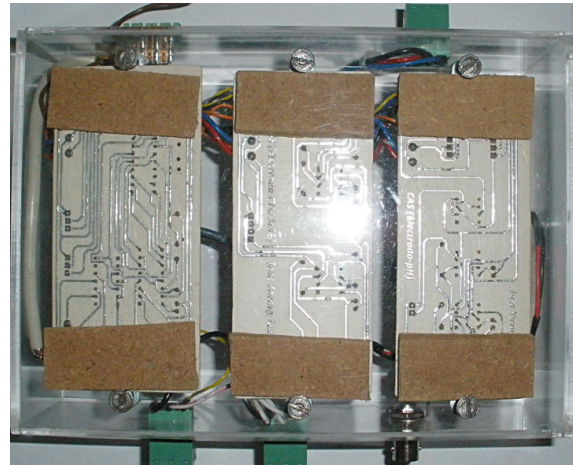
(b) Multímetro BK PRECISION 2831E.

**Figura 4.2.** Instrumentos de medición empleados en las pruebas.

El módulo del dispositivo CAS del sensor de pH y el módulo del CAS del sensor de OD y temperatura se conectan<sup>2</sup> al módulo del MCU para formar la RTU. Debido a que la fuente de alimentación se comparte entre los distintos módulos *hardware*, se configura una conexión cableada entre los conectores de alimentación de cada módulo. En la figura 4.3 se muestran las conexiones de módulos *hardware* y la instalación de dichos módulos en el gabinete. La configuración de las terminales de entrada y salida externas del gabinete de la RTU se detallan en la figura 4.4.



(a) Vista frontal de la RTU.

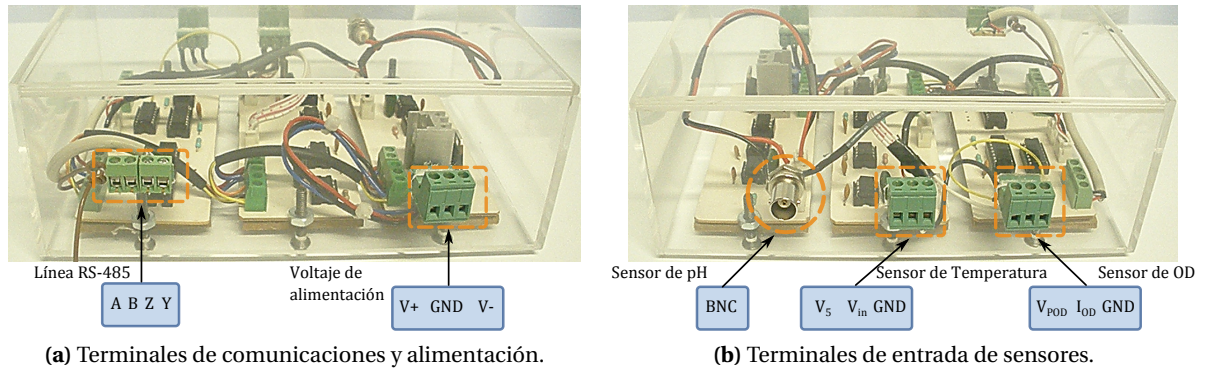


(b) Vista trasera de la RTU.

**Figura 4.3.** Presentación final de la RTU.

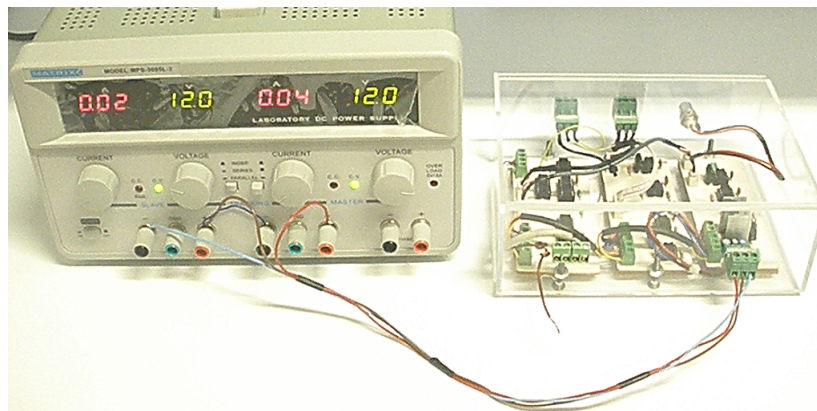
<sup>2</sup>La conexión entre los módulos *hardware* se realiza a través de los conectores de entrada y salida descritos en la sección 3.3.3.1.





**Figura 4.4.** Configuración de terminales en el gabinete de la RTU.

La fuente de voltaje MATRIX cuenta con botones de configuración para las salidas de voltaje, por lo que dichos botones se configuran para obtener una fuente de voltaje simétrica<sup>3</sup> y se establece un voltaje de salida de 12V para cada una de dichas salidas. En la figura 4.5 se muestra la RTU conectada a la fuente de voltaje o alimentación, y se valida que la conexión de módulos *hardware* de la RTU se ha realizado satisfactoriamente.



**Figura 4.5.** Prueba de conexiones de la RTU.

#### 4.1.2. Prueba de generación de $V_{offset}$ y $V_{POD}$

Esta prueba se realiza para verificar el funcionamiento de los dispositivos que generan el voltaje  $V_{offset}$  y  $V_{POD}$ . La medición del voltaje  $V_{offset}$  se realiza en la entrada negativa del sensor de OD, por ello, la medición se toma en la parte externa del conector BNC<sup>4</sup>.

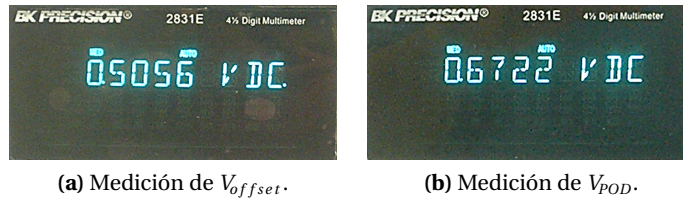
El valor medido de  $V_{offset}$  es de 505.6mV y para  $V_{POD}$  es de 672.2mV, con lo cual se valida que el funcionamiento de dichos dispositivos es correcto en base al diseño (Figura 4.6).  $V_{offset}$  garantiza

<sup>3</sup>Esta configuración es equivalente a conectar en serie ambas salidas de voltaje.

<sup>4</sup>En general, todas las mediciones se realizan haciendo referencia al plano de tierra.



que se obtengan niveles de voltaje positivos del sensor de pH y el voltaje  $V_{POD}$ , aunque presenta un error mínimo, garantiza un voltaje de polarización adecuado para el sensor de OD. Por lo tanto estos resultados son congruentes con el diseño de la sección 3.3.1.1.1.

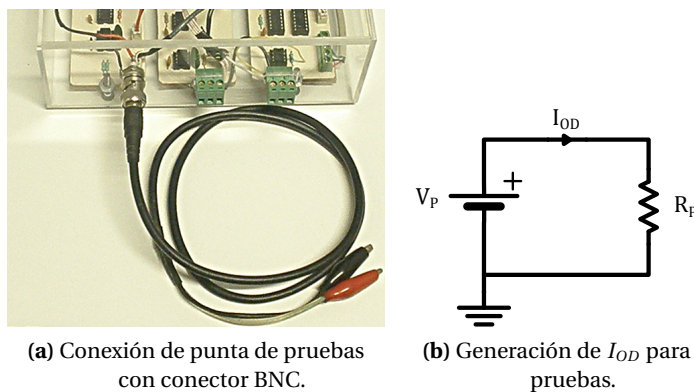


**Figura 4.6.** Mediciones de las salidas en los dispositivos para generar  $V_{offset}$  y  $V_{POD}$ .

#### 4.1.3. Prueba de variaciones de voltaje en las entradas

Esta prueba consiste en introducir voltajes o corriente (Según sea requerido) en las entradas de los dispositivos CAS para verificar su funcionamiento. Estas entradas están en el rango y dominio de las salidas de los sensores descritos en la sección 2.3.

El método empleado para simular la entrada del sensor de pH consiste en conectar una punta de prueba con conector macho tipo BNC e introducir niveles de voltaje en el rango definido para dicho sensor (Figura 4.7a). Para simular el sensor de OD se utiliza un arreglo simple que consta de una fuente de voltaje ( $V_p$ ) y una resistencia ( $R_p$ ) para generar una corriente ( $I_{OD}$ ) dentro del rango definido para dicho sensor (Figura 4.7b). Por su simplicidad, para el sensor de temperatura, solo se utiliza una fuente de voltaje variable.



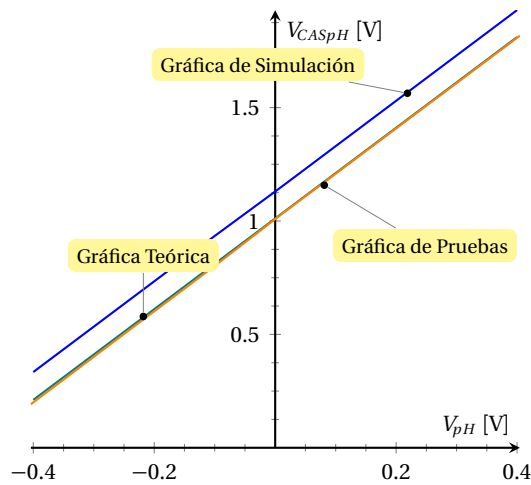
**Figura 4.7.** Configuración para pruebas de variaciones de voltaje.

En la tabla 4.1 se muestran las mediciones tomadas del dispositivo CAS del sensor de pH. Para esta prueba se hizo un barrido de voltaje en la entrada respetando el dominio definido en la sección 3.3.1.1.1.

Con la finalidad de hacer una comparación de resultados, en la tabla 4.1 se incluyen los resultados de la simulación y los valores teóricos, los cuales se calculan empleando la ecuación 3.3. Se realiza una comparación gráfica como se ilustra en la figura 4.8, en donde se observa que el dispositivo CAS para el sensor de pH presenta un mínimo error en comparación con la gráfica teórica, lo cual valida el funcionamiento de dicho dispositivo.

**Tabla 4.1.** Mediciones  $V_{CASpH}$  con variaciones en el voltaje de entrada del sensor de pH.

$V_{pH}$	$V_{CASpH}$	$V_{CASpH}$ Teórico	$V_{CASpH}$ Simulado
404.7mV	1.8172V	1.8112V	1.93V
307.8mV	1.6236V	1.6112V	1.73V
209.7mV	1.4277V	1.4112V	1.53V
100.7mV	1.2097V	1.2112V	1.33V
4mV	1.0165V	1.0112V	1.13V
-102.3mV	797.6mV	811.2mV	934mV
-209.4mV	583mV	611.2mV	734mV
-307mV	387mV	411.2mV	534mV
-404mV	192.1mV	211.2mV	334mV



**Figura 4.8.** Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de pH.

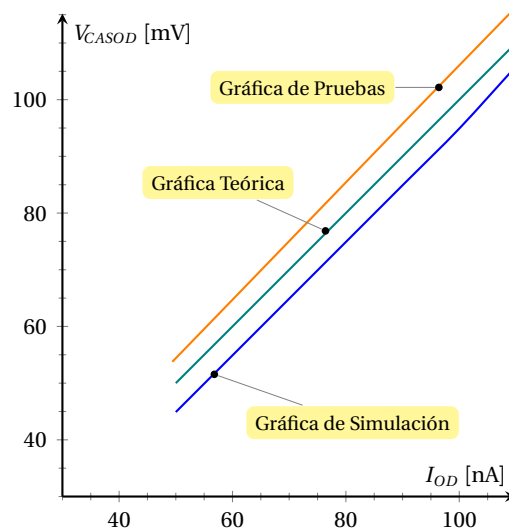
En la prueba del dispositivo CAS para el sensor de OD se toma una resistencia  $R_P = 10M\Omega$  y se hace un barrido en la fuente  $V_P$  de 300mV a 1.2mV, con lo cual se obtiene una corriente  $I_{OD}$  de 50nA a 110nA (Figura 4.7b), que es el dominio de salida del sensor de OD.

Las mediciones tomadas del CAS para el sensor de OD se muestran en la tabla 4.2, en donde se incluyen las mediciones de la simulación y los valores teóricos en base a la ecuación 3.6. En la

figura 4.9 se muestra de manera gráfica dichos resultados<sup>5</sup>, en donde las gráfica de las pruebas valida el funcionamiento del dispositivo CAS para el sensor de OD.

**Tabla 4.2.** Mediciones  $V_{CASOD}$  con variaciones en la corriente de entrada del sensor de OD.

$V_P$	$I_{OD}$	$V_{CASOD}$	$V_{CASOD}$ Teórico	$V_{CASOD}$ Simulado
502mV	49.385nA	53.76mV	49.385mV	44.9mV
600mV	59.026nA	63.7mV	59.026mV	54.9mV
713mV	70.142nA	75.2mV	70.142mV	64.9mV
812mV	79.881nA	85.29mV	79.881mV	74.9mV
910mV	89.522nA	95.26mV	89.522mV	84.9mV
1V	99.163nA	105.2mV	99.163mV	94.9mV
1.107V	108.903nA	115.2mV	108.903mV	105.9mV



**Figura 4.9.** Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de OD.

Para el dispositivo CAS del sensor de temperatura, en la tabla 4.3 se muestran las mediciones tomadas del CAS, de la simulación y los valores teóricos en base a la ecuación 3.7. Para esta prueba se consideró un dominio de entrada de 280mV a 320mV, que equivalentemente son valores de temperatura de 28 a 32°C.

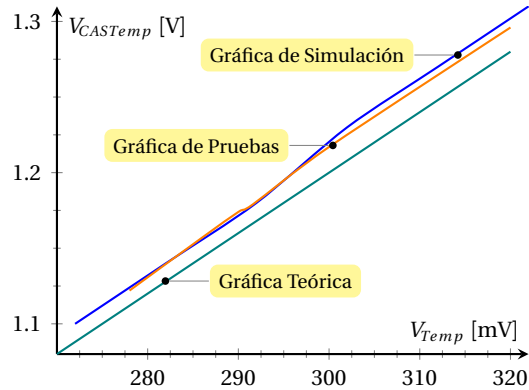
En la figura 4.10 se ilustra de manera gráfica los resultados obtenidos<sup>6</sup>, en donde el comportamiento de la gráfica de pruebas valida el funcionamiento del CAS para el sensor de temperatura.

<sup>5</sup>Tomando como referencia a los valores teóricos, existe un error en  $V_{CASOD}$ , lo cual se debe a que los elementos *hardware* presentan un margen de error según cada fabricante

<sup>6</sup>Se observa que en base a los valores teóricos el dispositivo CAS presenta un error, nuevamente debido a los márgenes de error en los elementos *hardware*

**Tabla 4.3.** Mediciones  $V_{CASTemp}$  con variaciones en el voltaje de entrada del sensor de temperatura.

$V_{Temp}$	$V_{CASTemp}$	$V_{CASTemp}$ Teórico	$V_{CASTemp}$ Simulado
278mV	1.122V	1.11V	1.10V
286mV	1.157V	1.14V	1.14V
290mV	1.174V	1.16V	1.18V
300mV	1.217V	1.2V	1.23V
320mV	1.296V	1.28V	1.31V

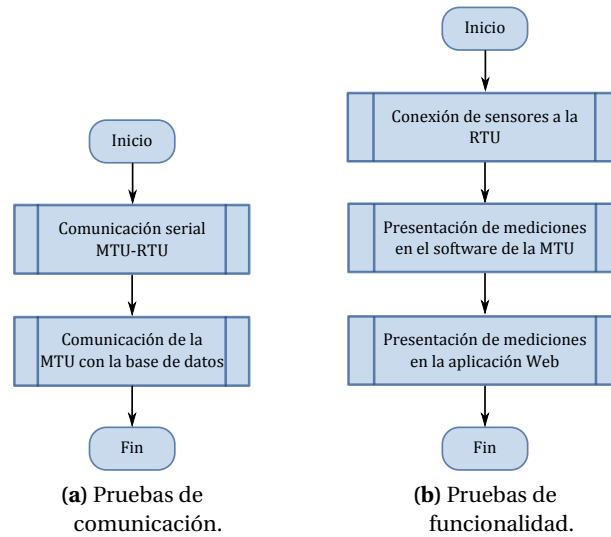
**Figura 4.10.** Gráfica de resultados arrojados en las pruebas, la simulación y de manera teórica para el CAS del sensor de temperatura.

## 4.2. Pruebas de Caja Gris

Las pruebas de caja gris se dividen en dos grupos con diferentes objetivos: realizar pruebas de comunicación entre subsistemas y pruebas de funcionalidad del sistema en general.

Las pruebas de comunicación consisten básicamente en verificar que la comunicación entre la RTU y la MTU se realice de manera satisfactoria, y que se lleve a cabo la comunicación entre el *software* de la MTU y la base de datos. Este grupo de pruebas se ilustra en el diagrama de flujo de la figura 4.11a.

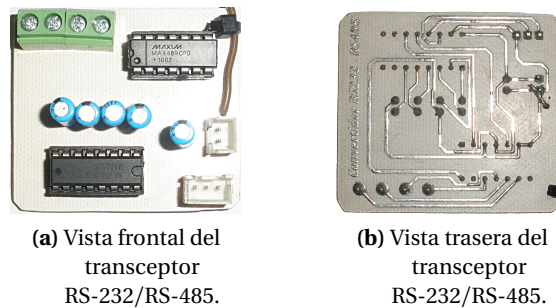
Para verificar que de manera general el sistema funciona adecuadamente, se realizan pruebas con los sensores reales, con ello, se verifica que el *software* de la MTU y la aplicación Web muestren las mediciones de manera correcta. En la figura 4.11b se ilustra el diagrama de flujo para este grupo de pruebas.



**Figura 4.11.** Diagramas de actividades para las pruebas de caja gris.

#### 4.2.1. Prueba de comunicación serie entre la MTU y la RTU

La prueba de comunicación serie entre la MTU y la RTU consiste en verificar que la comunicación entre la RTU y el *software* de la MTU se lleve a cabo bajo el estándar definido por el protocolo Modbus y la configuración de la interfaz serie definida en el apartado 3.3.2.1.3. Dicha comunicación serie se realiza a través del transceptor RS-232/RS-485, cuya presentación final se muestra en la figura 4.12.



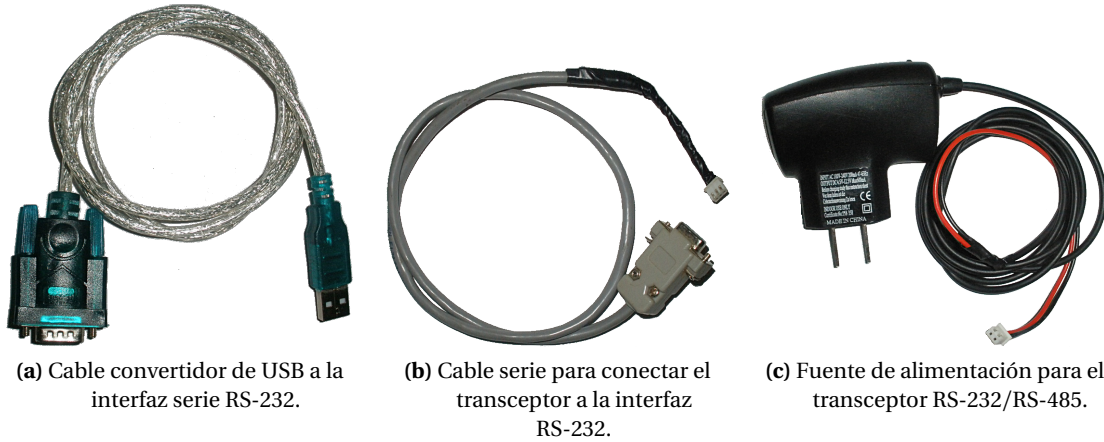
**Figura 4.12.** Presentación final del transceptor RS-232/RS-485.

Debido a que en esta prueba se utiliza una computadora portátil (*Notebook*), se requiere usar un dispositivo o cable convertidor de USB a la interfaz serie RS-232 (Figura 4.13a).

Para conectar el transceptor RS-232/RS-485 a la interfaz serie RS-232 se construye un cable serie

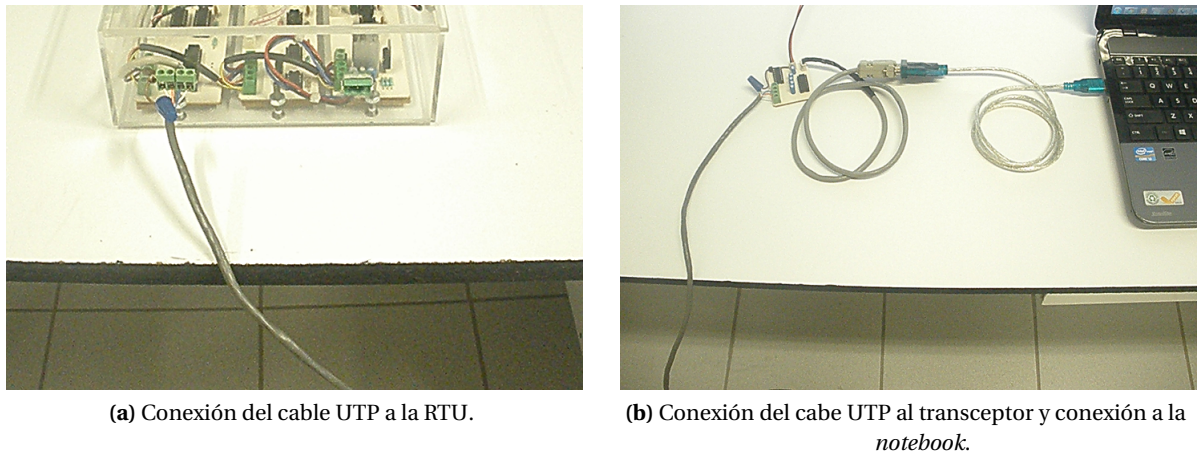
que consta de un conector *DB-9* hembra y un conector hembra de tres terminales<sup>7</sup> que se conecta a dicho transceptor (Figura 4.13b).

La fuente de alimentación utilizada para el transceptor RS-232/RS-485 es un convertidor de voltaje (o eliminador) de 4.5V, el cual se muestra en la figura 4.13c.



**Figura 4.13.** Elementos requeridos para la comunicación serie.

La configuración de conexiones para la comunicación entre la RTU y la MTU se muestra en la figura 4.14. Como se observa, la conexión entre la RTU y el transceptor RS-232/RS-485 se realiza a través de un cable de par trenzado (UTP, *Unshielded Twisted Pair*).

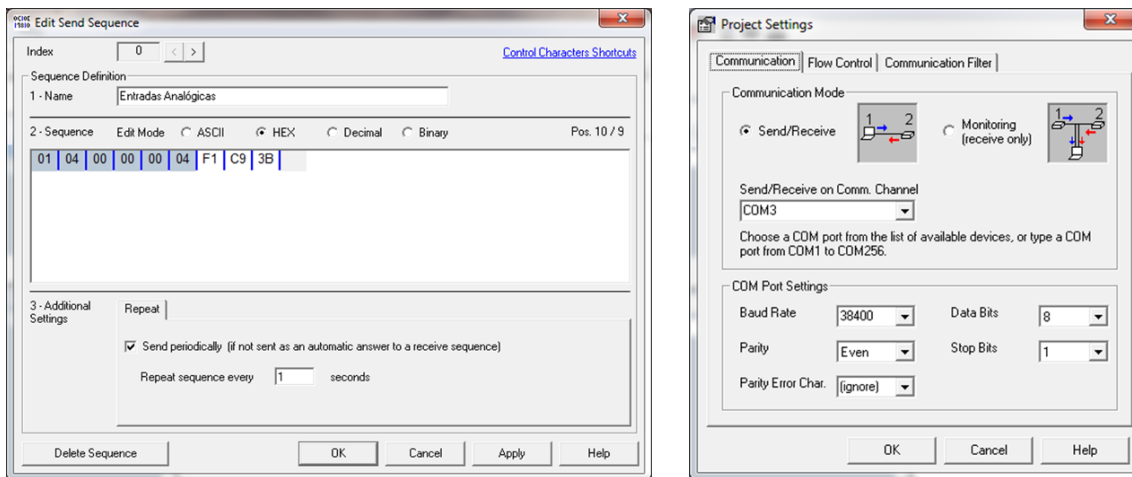


**Figura 4.14.** Configuración de conexiones para la comunicación entre la RTU y la MTU.

<sup>7</sup>La configuración de terminales para el cable serie se describe en la sección 3.3.3.1.4.

Para verificar que la comunicación se lleve a cabo satisfactoriamente, las pruebas de comunicación se realizan empleando dos cables UTP de diferentes longitudes: uno de aproximadamente 1m y otro de 3m, con esto se valida que la comunicación se mantiene conforme la distancia aumenta.

Se utiliza el *software Docklight*, con el que se envía a la RTU una ADU válida de manera constante simulando el monitoreo del *software* de la RTU (Figura 4.15a). *Docklight* se configura en base a los parámetros del diseño de la interfaz serie, según se muestra en la figura 4.15b.



(a) ADU de prueba para Docklight.

(b) Configuración del *software* Docklight.

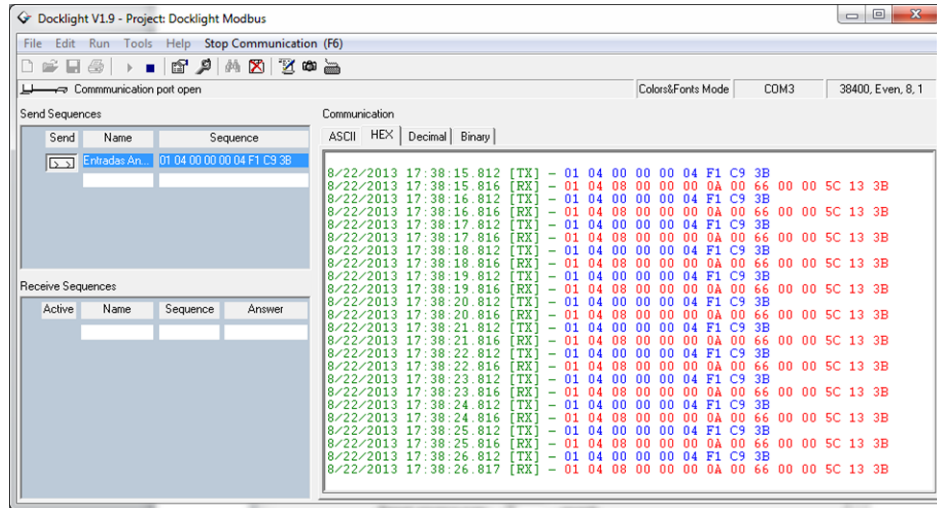
**Figura 4.15.** Configuración del *software* Docklight.

En la figura 4.16 se muestran los resultados de las pruebas de comunicación serie para los cables UTP de 1m y 3m respectivamente. En estas pruebas se consideraron tiempos de muestreo de 1s y 3s, sin embargo, solo se muestra la prueba de 1s debido a que es la prueba con mayor demanda en la comunicación serie. Con estos resultados se valida que la comunicación serie entre la RTU y la MTU se realiza satisfactoriamente.

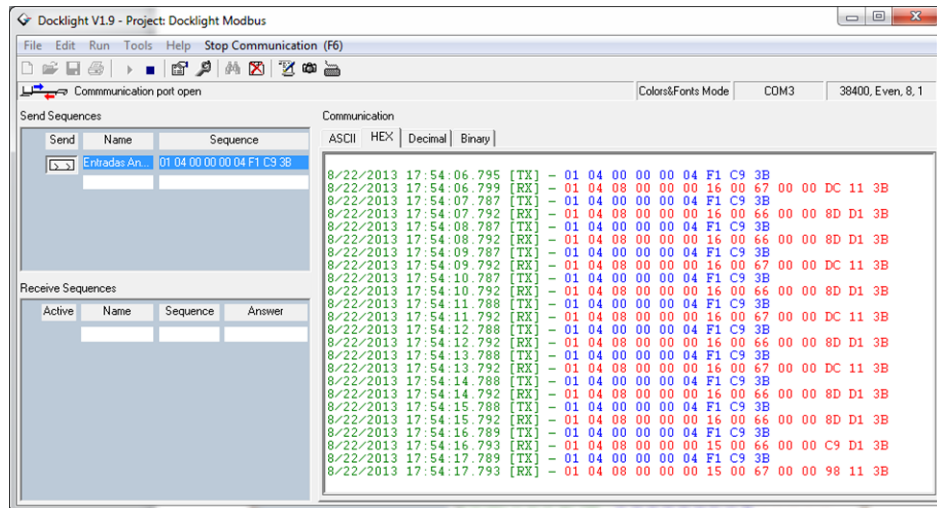
Como se observa en la parte de los registros de comunicación de *Docklight*, el texto en color verde muestra el registro de hora y fecha en que las ADUs son emitidas y recibidas, el texto en color azul muestra la ADU (en código hexadecimal) emitida y en color rojo se muestra la ADU que la RTU responde.

En la figura 4.17 se muestra el comportamiento del *software* de la MTU ante dos situaciones: cuando la RTU está desconectada de la interfaz serie, está apagada o se presentó un problema de comunicación y cuando la RTU está en funcionamiento y conectada adecuadamente a la interfaz serie.





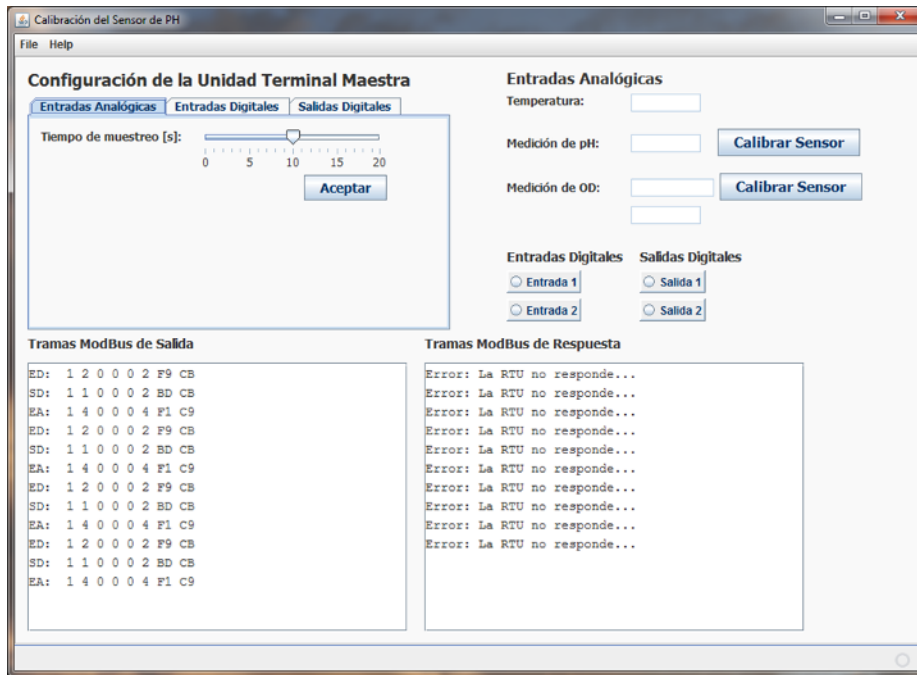
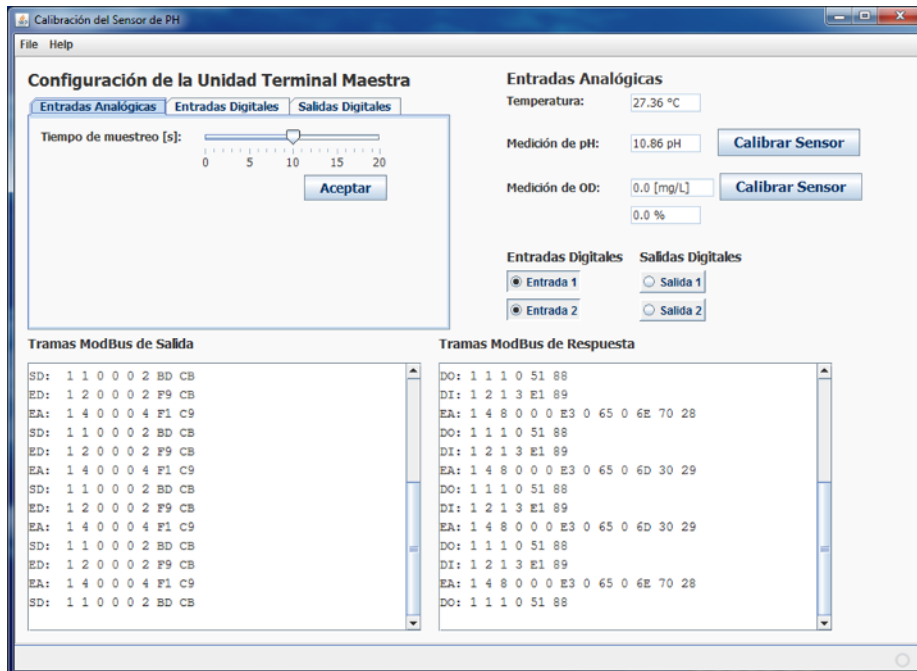
(a) Prueba de comunicación serie con el cable UTP de 1m de longitud.



(b) Prueba de comunicación serie con el cable UTP de 3m de longitud.

Figura 4.16. Resultados de las pruebas de comunicación serie con cables UTP de diferente longitud.



(a) Comportamiento del *software* de la MTU cuando no hay comunicación con la RTU.(b) Comportamiento del *software* de la MTU cuando hay comunicación con la RTU.Figura 4.17. Comportamiento del *software* ante fallos en la comunicación serie.

### 4.2.2. Prueba de comunicación entre la MTU y la base de datos

Esta prueba se realiza para verificar que el *software* de la MTU se comunique adecuadamente con la base de datos. En la figura 4.17b se muestra el comportamiento del *software* de la MTU cuando la conexión con la base de datos se realiza de manera satisfactoria. En cambio, la figura 4.18 muestra su comportamiento ante los siguientes casos: el servidor de base de datos está apagado y cuando se han presentado pérdidas de información en la comunicación.

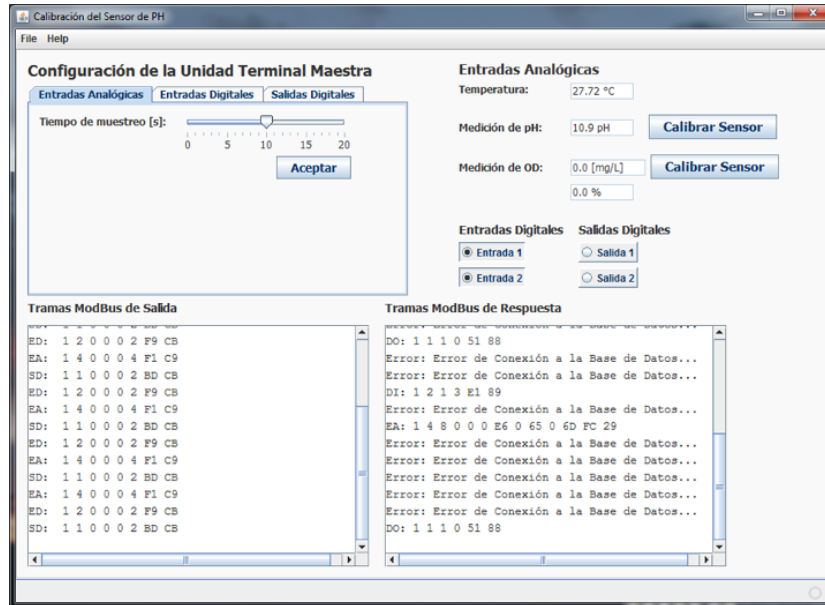


Figura 4.18. Comportamiento del *software* de la MTU ante fallos de comunicación con la base de datos.

### 4.2.3. Prueba de conexión de sensores a la RTU

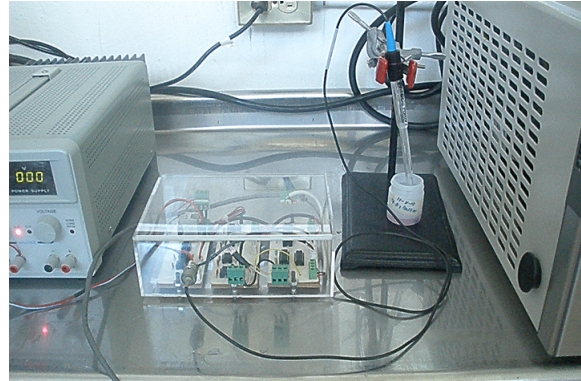
Esta prueba consiste en verificar que la conexión entre los dispositivos CAS y cada sensor (respectivamente) se realice sin problemas en cuanto a compatibilidad de impedancias y que la salida de los dispositivo CAS sea correcta. Por ello, se conectan los sensores a la RTU y se toman las mediciones en las salidas del dispositivo CAS de cada sensor.

Para probar el sensor de pH se utilizan tres muestras de laboratorio con los siguientes valores de pH fijo (Figura 4.19a): 4.01, 7.0 y 10.0. En la figura 4.19b se muestra la instalación del sensor de pH y su conexión con la RTU.

Los resultados al tomar las mediciones de cada muestra de laboratorio se describen en la tabla 4.4. En dicha tabla se incluye la medición de la salida correspondiente a  $V_{pH} + V_{offset}$  que sirve para calcular el valor de pH base a las especificaciones del sensor y el diseño *hardware*. El valor de pH calculado valida que la conexión del sensor de pH al dispositivo CAS se ha realizado satisfactoriamente.



(a) Muestras de laboratorio con valores de pH fijo.



(b) Instalación del sensor de pH y conexión con la RTU.

**Figura 4.19.** Conexión del sensor de pH a la RTU.**Tabla 4.4.** Resultados en la prueba de conexión del sensor de pH.

pH de la muestra	$V_{pH} + V_{offset}$	$V_{CASpH}$	pH Calculado
4.01	640.3mV	1.2785V	4.62
7.0	455.9mV	913.8mV	7.74
10.0	272.5mV	543.2mV	10.84

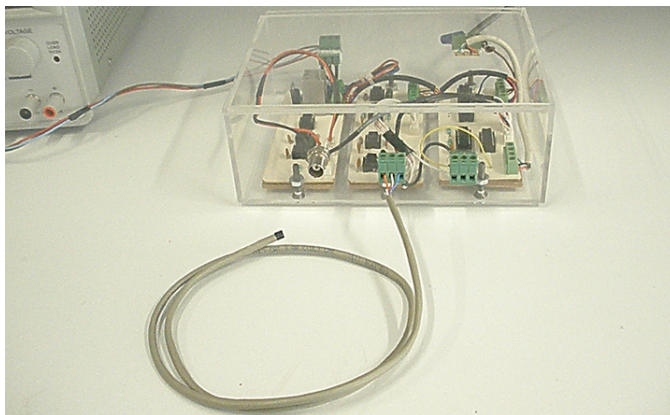
Cabe mencionar que el sensor de pH puede perder su precisión tras el uso constante, por ello, el valor de pH obtenido del cálculo difiere ligeramente al de la muestra, este error se corrige en el *software* de la MTU al realizar la calibración del sensor.

Para probar el sensor de temperatura se construyó una extensión, la cual consiste en el sensor de temperatura y un cable con tres salidas conectadas a cada terminal del sensor, de esta manera se facilita el manejo y la instalación de dicho sensor. En la figura 4.20 se muestra la configuración y la conexión del sensor de temperatura a la RTU.

Los resultados de la prueba de conexión del sensor de temperatura se muestran en la tabla 4.5, en donde se muestran las mediciones del sensor y la temperatura calculada en base a las especificaciones del sensor y el diseño *hardware*. Estas mediciones validan que el sensor de temperatura se ha conectado de manera satisfactoria a la RTU.

**Tabla 4.5.** Resultados en la prueba de conexión del sensor de temperatura.

Voltaje del sensor	$V_{CASTemp}$	Temperatura Calculada
275mV	1.114V	27.45°
278mV	1.122V	27.69°
283mV	1.140V	28.14°
290mV	1.173V	28.95°
300mV	1.211V	29.89°
310mV	1.250V	30.85°



**Figura 4.20.** Conexión del sensor de temperatura a la RTU.

Debido a que el sensor de OD (*Mettler-Toledo InPro 6800*) con el que cuenta el Instituto de Agroindustrias de la UTM no funciona adecuadamente, no fue posible llevar a cabo la prueba para este sensor. En cambio se verifica que, en base a las especificaciones de este sensor descritas en el apartado 2.3.2, el dispositivo CAS entrega las salidas acorde a dichas especificaciones y necesidades del diseño *hardware* y *software*. En efecto, las pruebas de la sección 4.1.3 validan el funcionamiento de dicho dispositivo CAS.

#### 4.2.4. Prueba del *software* de la MTU

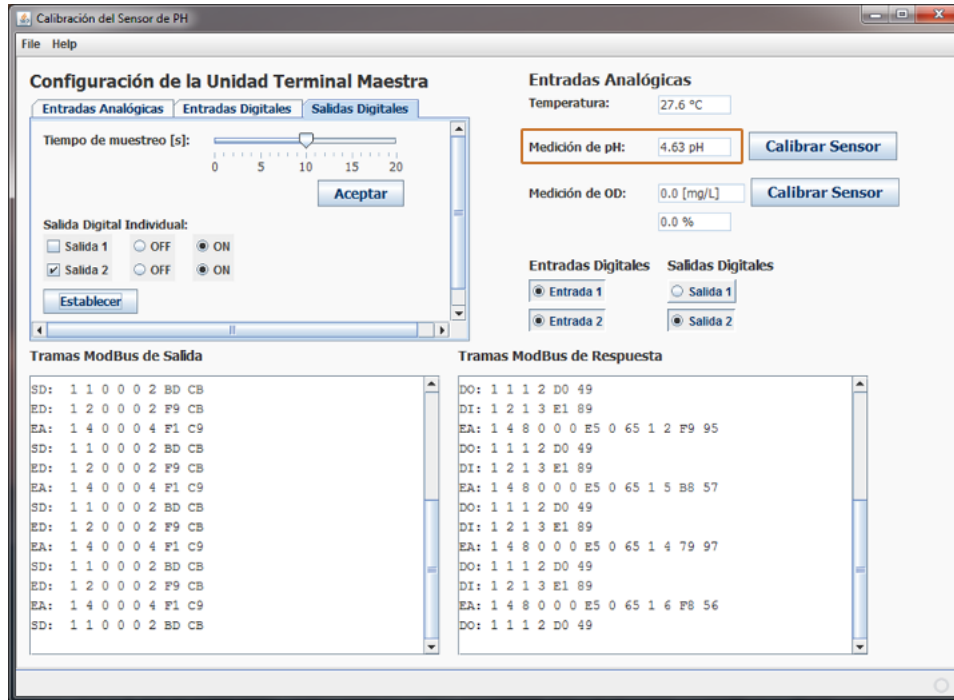
En esta prueba se verifica que el *software* de la MTU realiza los cálculos de las mediciones basándose en el diseño *hardware* y las especificaciones de los sensores. El *software* de la MTU presenta dichos cálculos en la interfaz de usuario y almacena un registro de medición en la base de datos.

En la figura 4.21 se muestra el estado del *software* de la MTU mientras se realizan las pruebas del sensor de pH<sup>8</sup> de la sección 4.2.3. Como se observa, las mediciones que este *software* presenta en la interfaz gráfica, son consistentes con los cálculos de la tabla 4.4, por lo que se genera el registro de medición que es almacenado en la base de datos. De esta manera se valida que el *software* de la MTU presenta las mediciones adecuadamente.

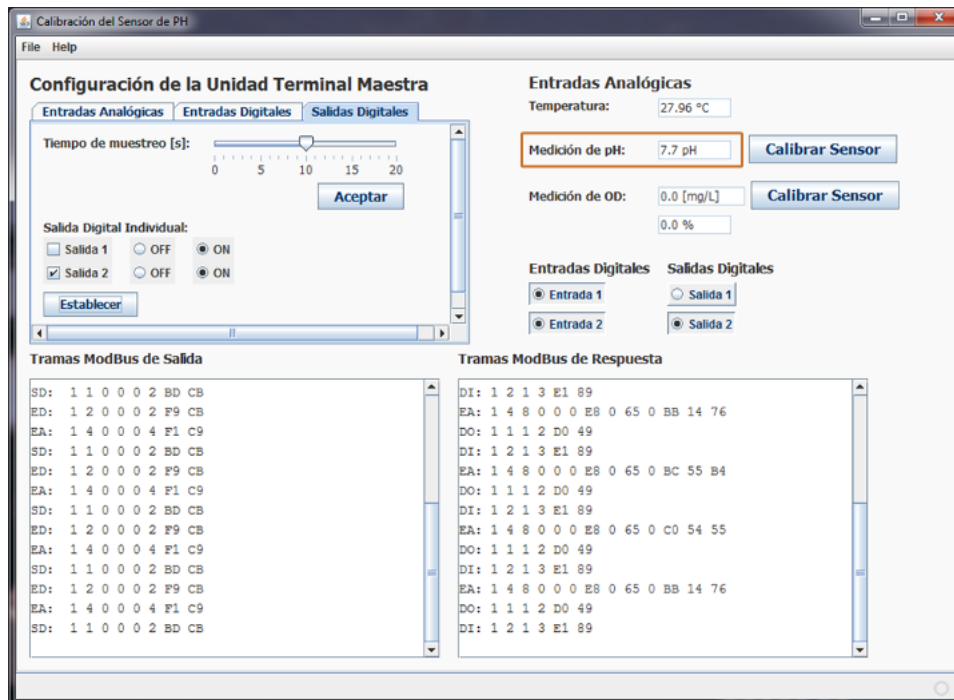
Como se mencionó anteriormente, es posible corregir el error de medición que se presenta debido al desgaste del sensor de pH. Ésto se logra haciendo una calibración, la cual permite ajustar la ecuación de dicho sensor y así obtener de manera correcta los valores de las muestras.

El proceso de calibración consiste en tomar dos mediciones de muestras con pH fijo, se toma una a la vez y el *software* de la MTU ajusta de manera automática la ecuación de pH para el sensor. En la figura 4.22 se muestra la ventana de calibración de pH y cuando se finaliza la calibración. Como se observa, en la calibración se pueden escoger de tres muestras con diferentes valores de pH, para esta prueba se tomaron las muestras con valor de pH de 4 y 7.

<sup>8</sup>Solo se presentan las mediciones de las muestras con valor de pH de 4.01 y 7.0.



(a) Mediciones para la muestra con valor de pH de 4.01.



(b) Mediciones para la muestra con valor de pH de 7.0.

Figura 4.21. Presentación de mediciones de pH en el *software* de la MTU.



**Figura 4.22.** Pruebas de calibración del sensor de pH.

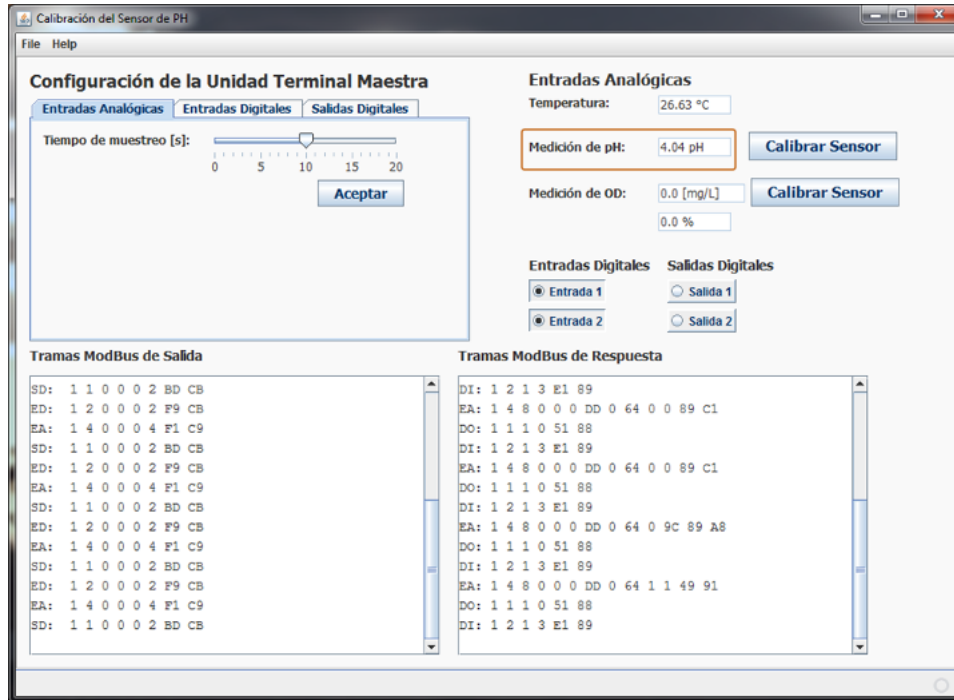
Cuando la calibración ha finalizado, se vuelve a hacer el proceso de medición de las muestras de pH fijo. El resultado se muestra en la figura 4.23, en donde se observa que el *software* de la MTU ajusta la gráfica de pH de acuerdo al funcionamiento actual del sensor, esto permite obtener mediciones mas precisas y corregir los errores de medición debido al desgaste del sensor por el uso constante. Con estos resultados se valida que el módulo de calibración para el sensor de pH realiza el ajuste de manera exitosa.

En la figura 4.24 se muestra como el *software* de la MTU presenta las mediciones de temperatura, dichas mediciones se registran y almacenan en la base de datos. Esta medición se toma mientras se realiza la prueba de la sección 4.2.3, por ello, estas mediciones de temperatura son consistentes con la tabla 4.3.

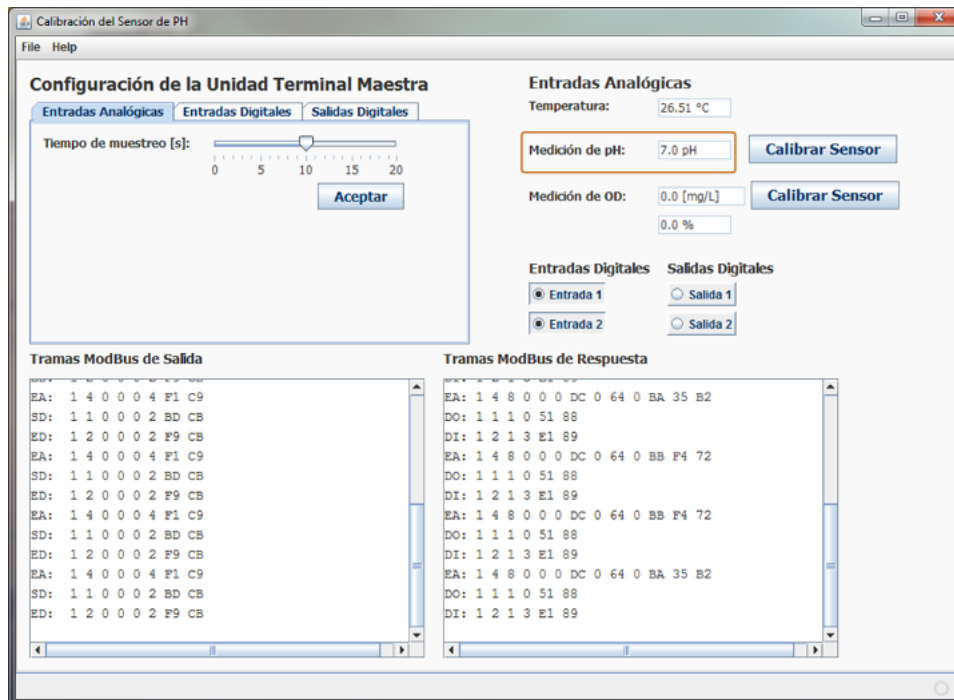
Para las mediciones de OD se considera la corriente de 110nA como la corriente máxima de saturación, es decir, indica el 100% de saturación de OD<sup>9</sup>. En base a las referencias [6, 42], se obtiene que para la ciudad de Huajuapán de León, Oaxaca, el índice máximo de saturación de oxígeno es de  $6.864 \frac{\text{mgO}_2}{\text{L}}$ . En la figura 4.25 se muestra que, al hacer el barrido de corriente como en la tabla 4.2 de la sección 4.1.3, el índice y el porcentaje de saturación de OD varían en proporción a la corriente de entrada al dispositivo CAS del sensor de OD.

<sup>9</sup>De acuerdo a las especificaciones del sensor de OD





(a) Mediciones para la muestra con valor de pH de 4.01.



(b) Mediciones para la muestra con valor de pH de 7.0.

**Figura 4.23.** Mediciones en el *software* de la MTU después de realizar la calibración del sensor de pH.

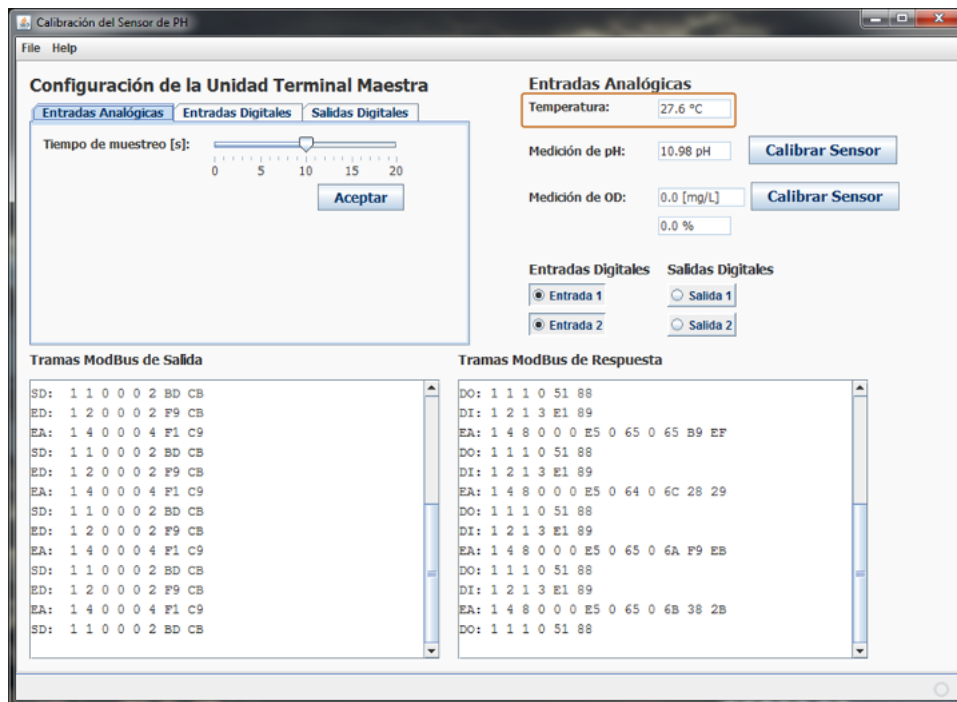
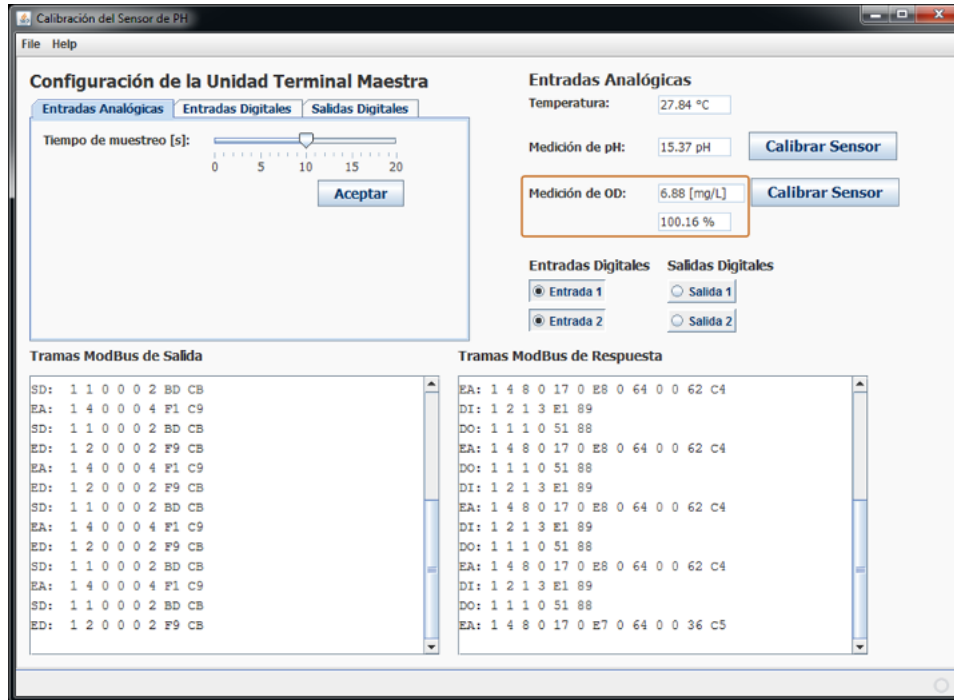
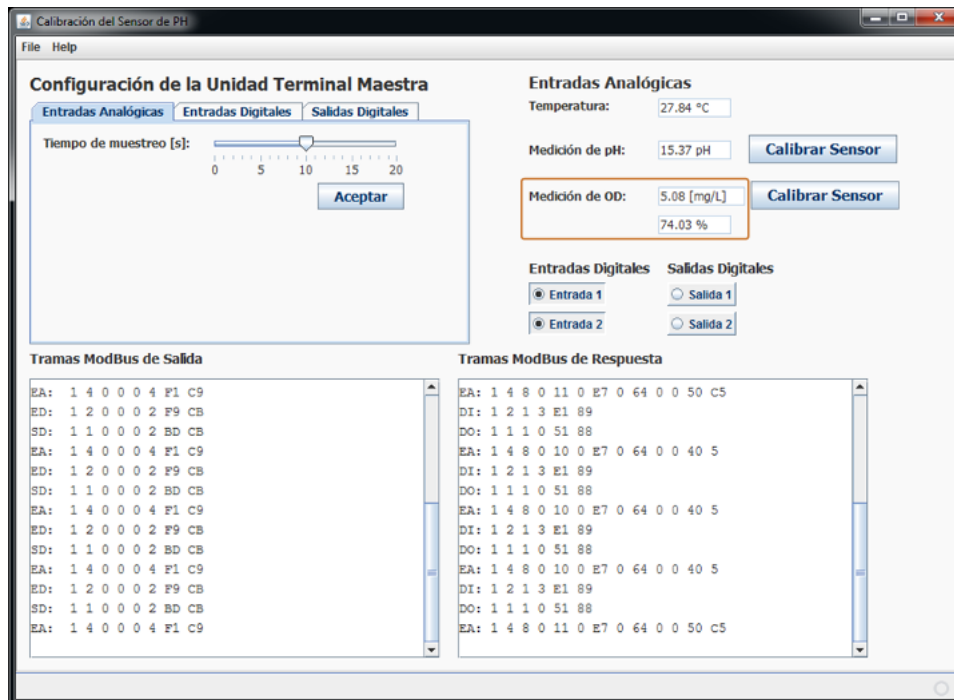


Figura 4.24. Presentación de mediciones de temperatura en el *software* de la MTU.





(a) Mediciones para un porcentaje de saturación de OD de 100% aproximadamente.



(b) Mediciones para un porcentaje de saturación de OD de 75% aproximadamente.

Figura 4.25. Presentación de mediciones de OD en el software de la MTU

Se considera que el sensor de OD puede cambiar su precisión conforme el uso, dicho deterioro se compensa en el *software* de la MTU a través de la calibración de dicho sensor. Esta calibración consiste en tomar un nuevo valor de corriente (del sensor de OD) como corriente máxima de saturación, no obstante, se requiere la presión barométrica para ajustar el índice máximo de saturación de OD conforme a la altitud del lugar donde se tomen las mediciones.

De acuerdo a la referencia [6], la presión barométrica varía en función de la altitud del lugar, por lo que la presión barométrica para la ciudad de Huajuapán de León, Oaxaca, es de 631.19mmHg<sup>10</sup>. Finalmente, el *software* de la MTU ajusta los cálculos de OD de acuerdo a dichos parámetros. En la figura 4.26 se muestra la ventana de calibración del sensor de OD.

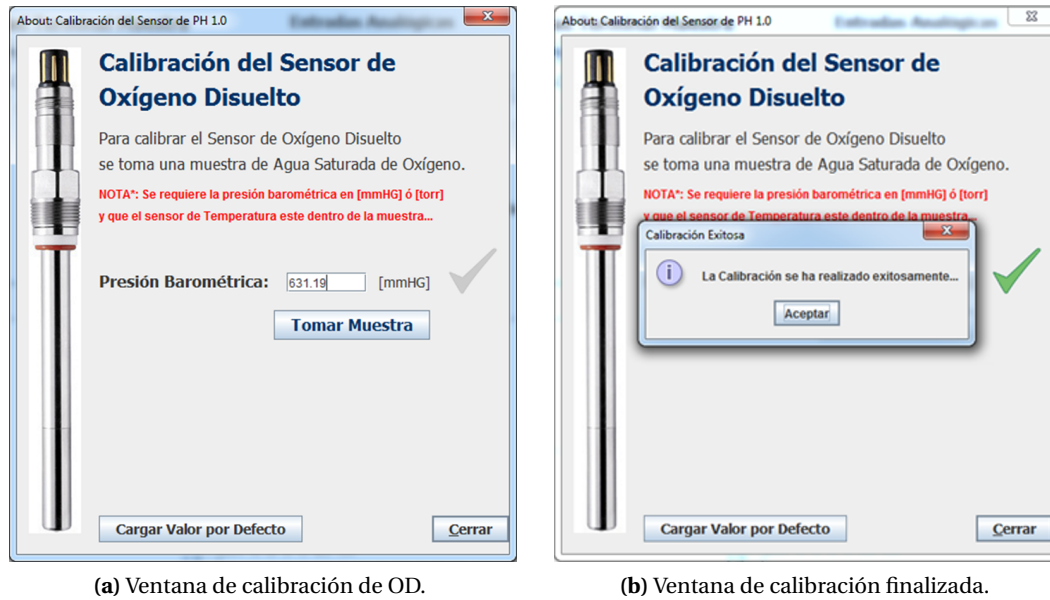


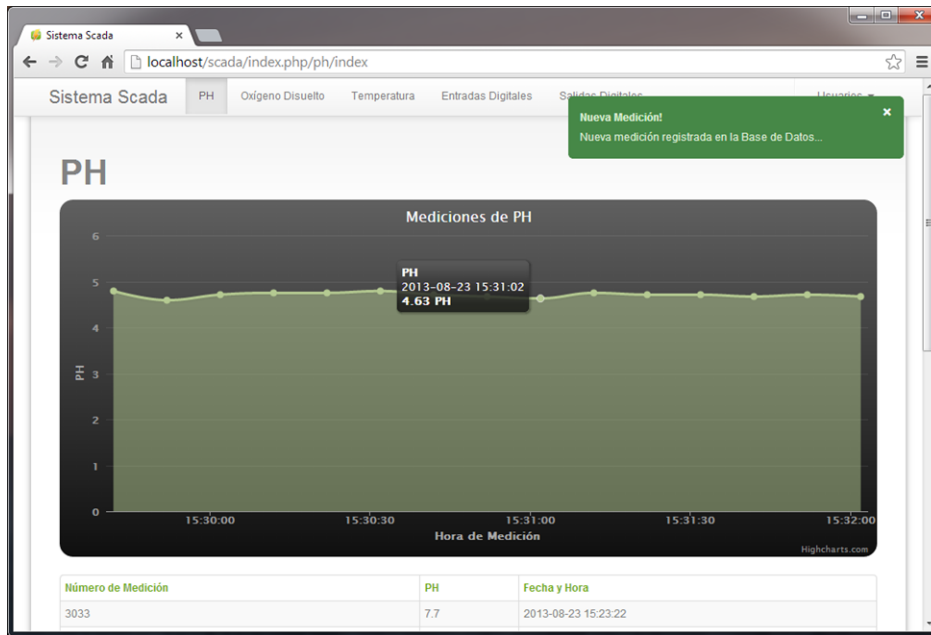
Figura 4.26. Pruebas de calibración del sensor de OD.

#### 4.2.5. Prueba de la aplicación Web

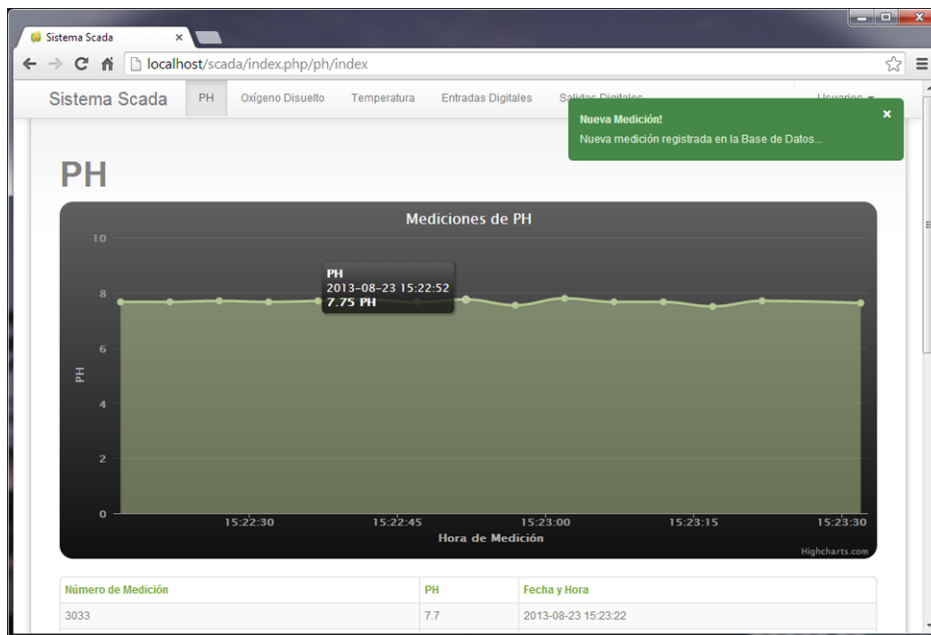
Esta prueba consiste en verificar que la aplicación Web presente los registros de mediciones almacenados en la base de datos. Por comodidad, la aplicación Web se divide en cinco módulos funcionales: mediciones de pH, mediciones de OD, mediciones de temperatura, entradas digitales y salidas digitales. Estas pruebas se enfocan en verificar que los registros de mediciones de los módulos de mediciones de pH, temperatura y OD se muestren de manera satisfactoria.

En la figura 4.27 se muestra la gráfica de las mediciones de pH, en la figura 4.28 se muestran las gráficas de temperatura y de OD y en la figura 4.29 se muestran los registros de medición en forma tabular. Estas capturas fueron tomadas mientras se realizaban las pruebas de la sección 4.2.3 (para pH y temperatura) y la sección 4.1.3 (para OD).

<sup>10</sup>Este valor de presión barométrica se tomará como valor por defecto si otro no es especificado y cuando se selecciona la opción de cargar valores por defecto. Las unidades mmHg significan milímetros de mercurio.



(a) Gráfica de mediciones para la muestra con un valor de pH de 4.

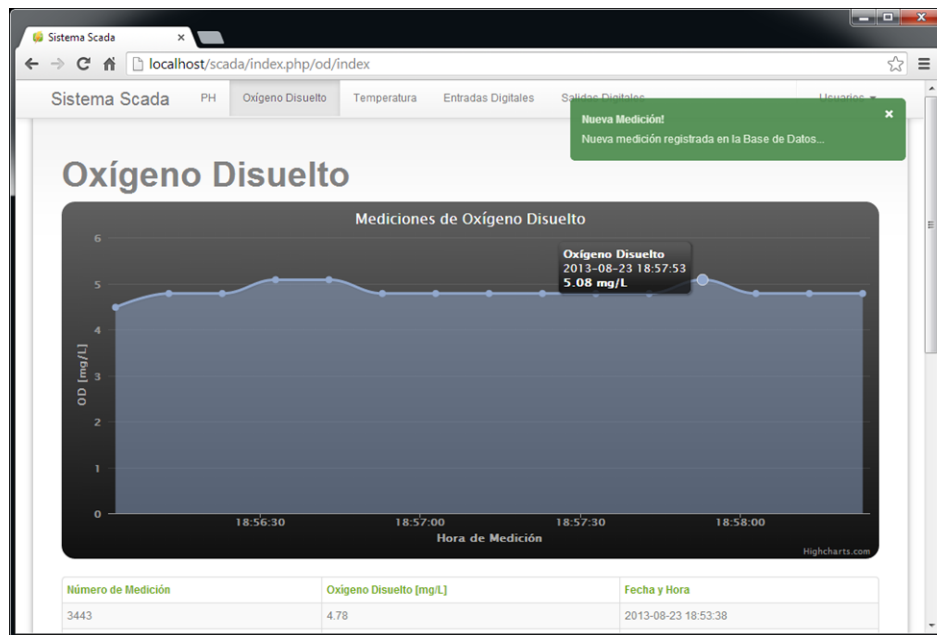


(b) Gráfica de mediciones para la muestra con un valor de pH de 7.

**Figura 4.27.** Presentación de mediciones de pH en la aplicación Web.

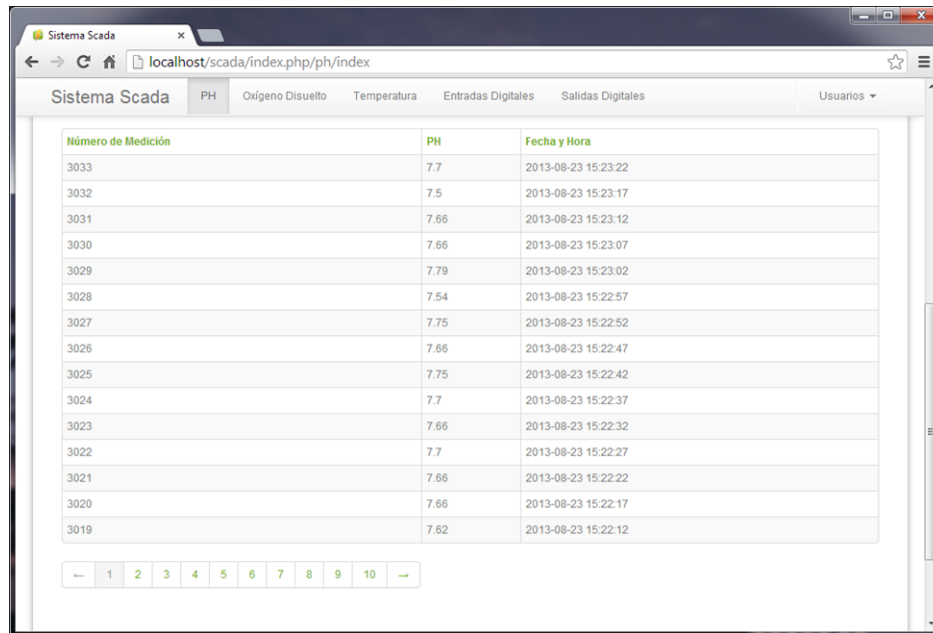


(a) Gráfica de mediciones de temperatura.



(b) Gráfica de mediciones de OD.

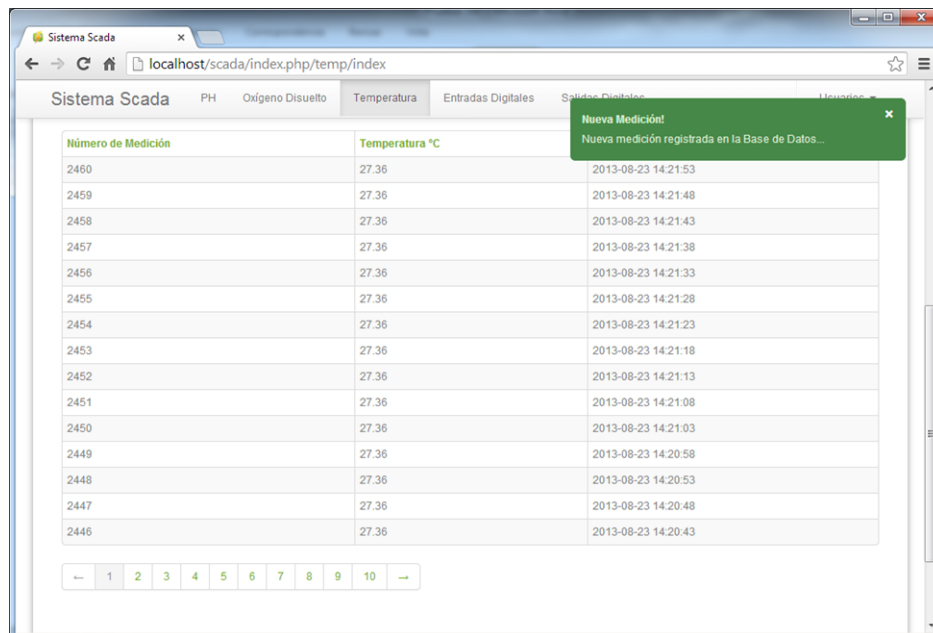
**Figura 4.28.** Presentación de mediciones de temperatura y OD en la aplicación Web.



The screenshot shows a web browser window titled 'Sistema Scada' with the URL 'localhost/scada/index.php/ph/index'. The page has a navigation menu with 'PH' selected. The main content is a table with three columns: 'Número de Medición', 'PH', and 'Fecha y Hora'. The table contains 15 rows of data. Below the table is a pagination control showing page 1 of 10.

Número de Medición	PH	Fecha y Hora
3033	7.7	2013-08-23 15:23:22
3032	7.5	2013-08-23 15:23:17
3031	7.66	2013-08-23 15:23:12
3030	7.66	2013-08-23 15:23:07
3029	7.79	2013-08-23 15:23:02
3028	7.54	2013-08-23 15:22:57
3027	7.75	2013-08-23 15:22:52
3026	7.66	2013-08-23 15:22:47
3025	7.75	2013-08-23 15:22:42
3024	7.7	2013-08-23 15:22:37
3023	7.66	2013-08-23 15:22:32
3022	7.7	2013-08-23 15:22:27
3021	7.66	2013-08-23 15:22:22
3020	7.66	2013-08-23 15:22:17
3019	7.62	2013-08-23 15:22:12

(a) Tabla de mediciones de pH.



The screenshot shows a web browser window titled 'Sistema Scada' with the URL 'localhost/scada/index.php/temp/index'. The page has a navigation menu with 'Temperatura' selected. The main content is a table with three columns: 'Número de Medición', 'Temperatura °C', and 'Fecha y Hora'. The table contains 15 rows of data. A green notification box is overlaid on the table, stating 'Nueva Medición! Nueva medición registrada en la Base de Datos...'. Below the table is a pagination control showing page 1 of 10.

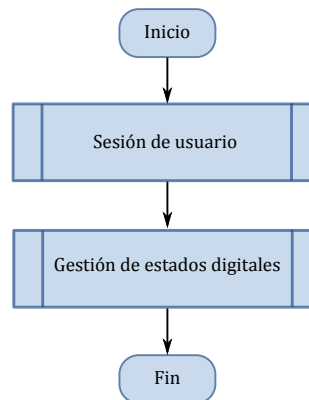
Número de Medición	Temperatura °C	Fecha y Hora
2460	27.36	2013-08-23 14:21:53
2459	27.36	2013-08-23 14:21:48
2458	27.36	2013-08-23 14:21:43
2457	27.36	2013-08-23 14:21:38
2456	27.36	2013-08-23 14:21:33
2455	27.36	2013-08-23 14:21:28
2454	27.36	2013-08-23 14:21:23
2453	27.36	2013-08-23 14:21:18
2452	27.36	2013-08-23 14:21:13
2451	27.36	2013-08-23 14:21:08
2450	27.36	2013-08-23 14:21:03
2449	27.36	2013-08-23 14:20:58
2448	27.36	2013-08-23 14:20:53
2447	27.36	2013-08-23 14:20:48
2446	27.36	2013-08-23 14:20:43

(b) Tabla de mediciones de temperatura.

**Figura 4.29.** Presentación de los registros de mediciones en forma tabular.

### 4.3. Pruebas de Caja Negra

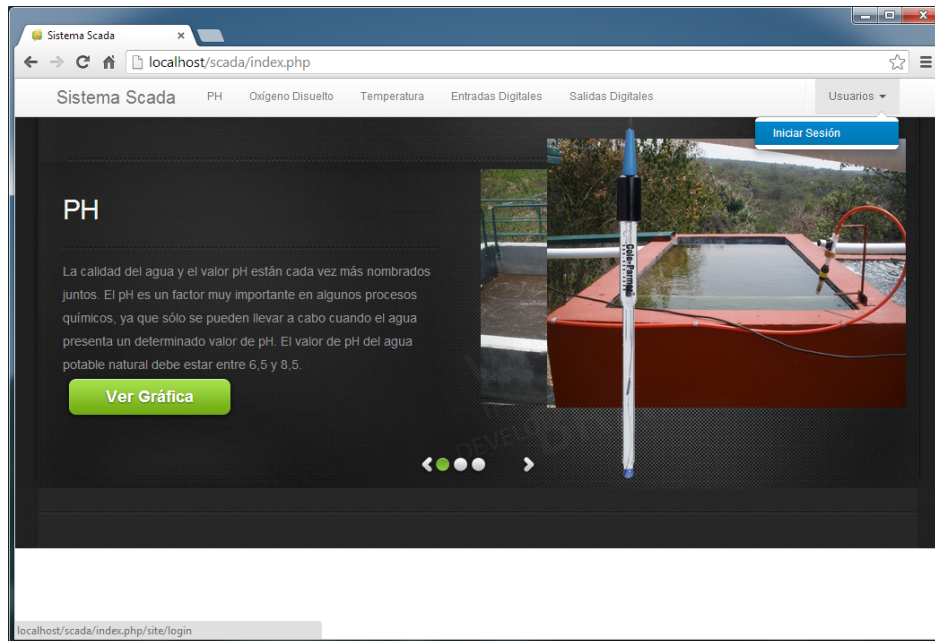
Las pruebas de caja negra consisten en verificar el funcionamiento de los módulos de sesión de usuario y la gestión de entradas y salidas digitales. Con estas pruebas se verifica que la aplicación Web gestiona de manera adecuada la seguridad de la misma y se verifica que la presentación y gestión de estados digitales se realice de manera exitosa. Las pruebas de caja negra se ilustran en el diagrama de actividades de la figura 4.30.



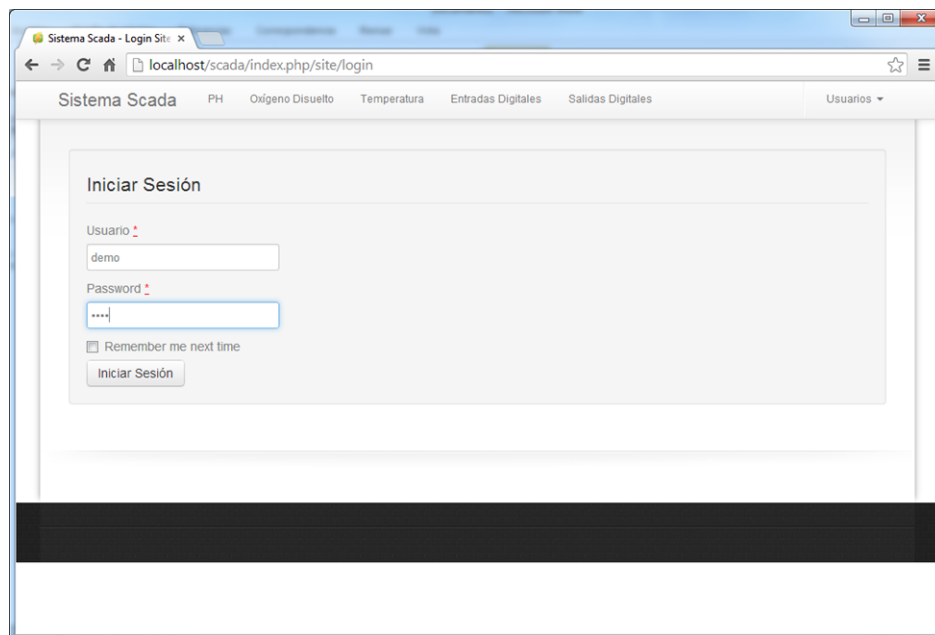
**Figura 4.30.** Diagrama de actividades para las pruebas de caja negra.

#### 4.3.1. Prueba de gestión de sesión de usuario

Esta prueba se realiza para verificar que la gestión de usuario se realiza de manera adecuada, es decir, que solo el administrador pueda tener acceso a los módulos de registros de mediciones. No obstante, cualquier usuario podrá acceder a la ruta principal de la aplicación Web, dicha ruta solo muestra una vista de bienvenida (Figura 3.52). Pero, al intentar acceder a algún módulo se debe solicitar el inicio de sesión del Administrador, aunque se puede iniciar sesión desde el menú de usuario. En a figura 4.31 se muestra la opción del menú de usuario y el formulario para iniciar sesión.



(a) Menú de usuario para iniciar sesión.



(b) Formulario para iniciar sesión.

**Figura 4.31.** Gestión de usuarios en la aplicación Web.

### 4.3.2. Prueba de gestión de estados digitales

En esta prueba se verifica que la aplicación Web presenta los registros de estados digitales de manera correcta. En la figura 4.32a se muestra la forma en que se presentan los registros de estados digitales (entradas<sup>11</sup> y salidas) y en la figura 4.32b se muestra el formulario para que el Administrador establezca los estados de las salidas digitales.

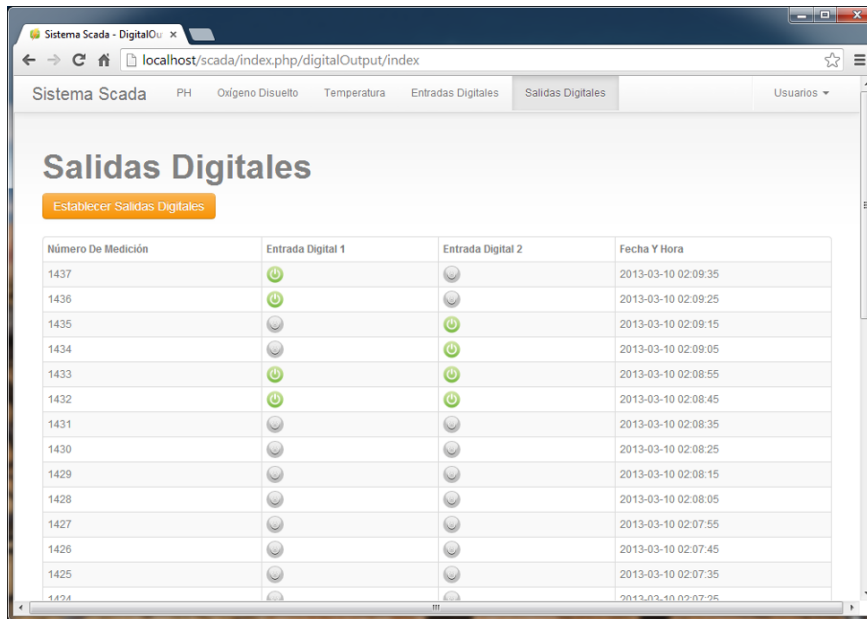
Como se observa en la figura 4.32a, el icono en color rojo indica un estado activo (ON) y el icono en color gris indica un estado inactivo (OFF). Ahora, Si el formulario es aceptado, se construye una ADU de solicitud y es enviada a la MTU, la cual, por medio del servidor Web (Implementado con un *socket*) recibe la ADU y la transfiere a la RTU a través de la interfaz serie.

En la figura 4.33a se muestra cuando el *software* de la MTU recibe la ADU de solicitud que se emite desde la aplicación Web, dicha ADU es enviada a la RTU. En la figura 4.33b se muestra como establecer las salidas digitales desde el *software* de la MTU.

---

<sup>11</sup>No se muestra la captura para las entradas digitales ya que es idéntica a la presentación de las salidas, obviamente la vista de las entradas digitales no incluye el botón para establecer estados digitales.

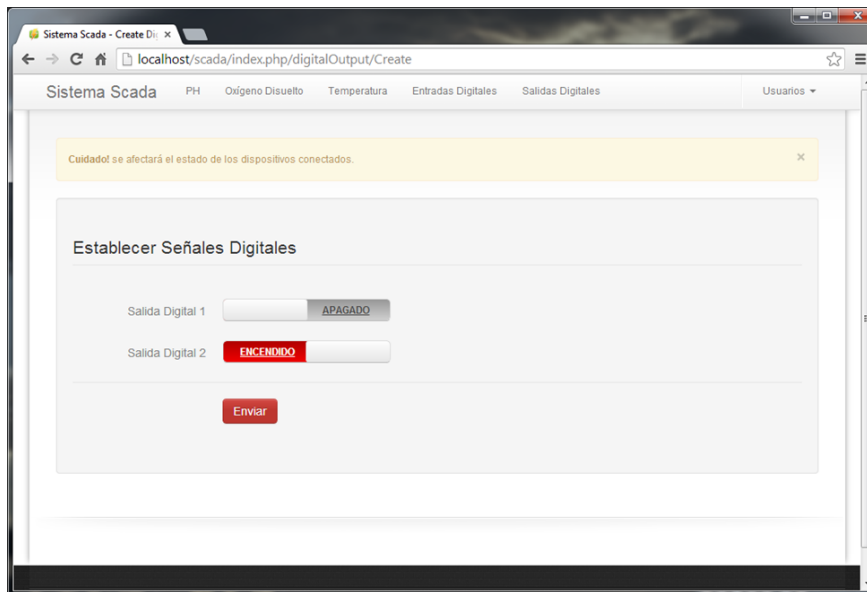




The screenshot shows a web browser window titled 'Sistema Scada - DigitalOu'. The address bar displays 'localhost/scada/index.php/digitalOutput/index'. The navigation menu includes 'PH', 'Oxígeno Disuelto', 'Temperatura', 'Entradas Digitales', 'Salidas Digitales', and 'Usuarios'. The main heading is 'Salidas Digitales' with a sub-button 'Establecer Salidas Digitales'. Below is a table with the following data:

Número De Medición	Entrada Digital 1	Entrada Digital 2	Fecha Y Hora
1437	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:09:35
1436	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:09:25
1435	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2013-03-10 02:09:15
1434	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2013-03-10 02:09:05
1433	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2013-03-10 02:08:55
1432	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2013-03-10 02:08:45
1431	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:08:35
1430	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:08:25
1429	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:08:15
1428	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:08:05
1427	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:07:55
1426	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:07:45
1425	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:07:35
1424	<input type="checkbox"/>	<input type="checkbox"/>	2013-03-10 02:07:25

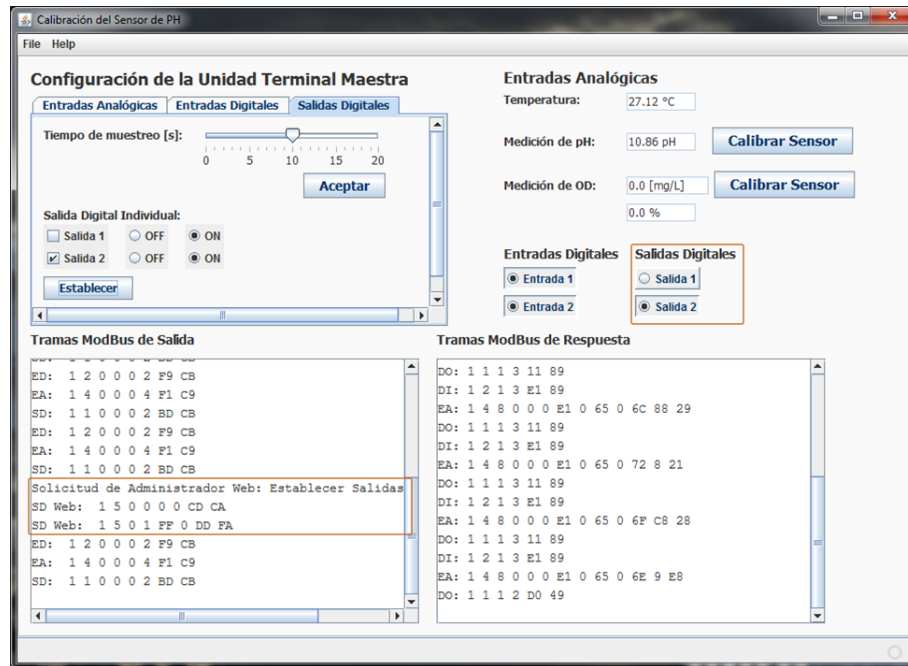
(a) Presentación de registros de estados digitales.



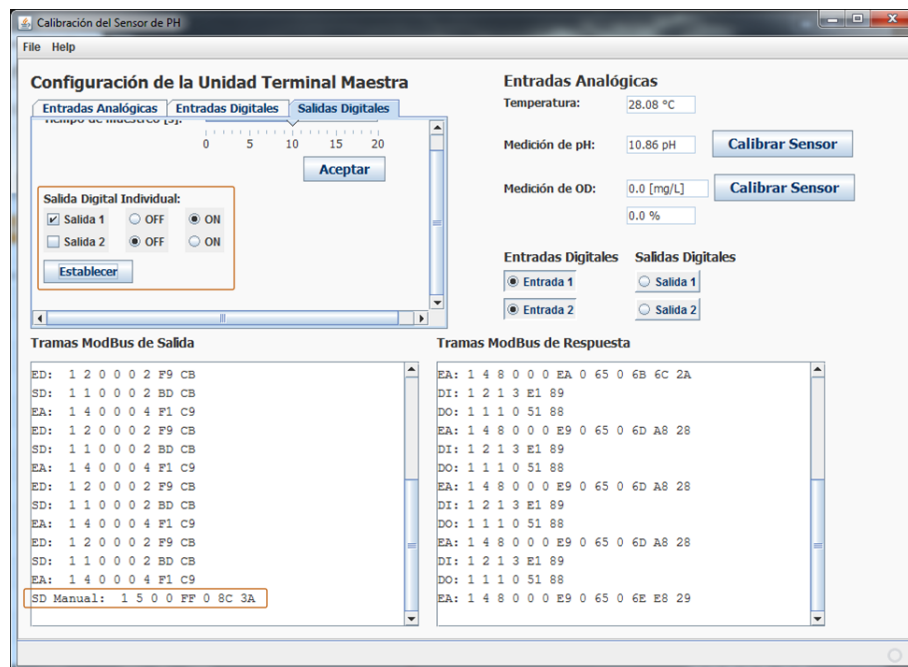
The screenshot shows a web browser window titled 'Sistema Scada - Create Di'. The address bar displays 'localhost/scada/index.php/digitalOutput/Create'. The navigation menu is the same as in (a). A yellow warning box at the top reads 'Cuidado! se afectará el estado de los dispositivos conectados.' Below is the heading 'Establecer Señales Digitales'. The form contains two rows: 'Salida Digital 1' with a button labeled 'APAGADO' and 'Salida Digital 2' with a button labeled 'ENCENDIDO'. At the bottom is a red 'Enviar' button.

(b) Formulario para establecer salidas digitales.

**Figura 4.32.** Presentación y gestión de salidas digitales en la aplicación Web.



(a) Recepción de la ADU emitida desde la aplicación Web.



(b) Gestión de salidas digitales desde el software de la MTU.

**Figura 4.33.** Recepción de la ADU emitida desde la aplicación Web y gestión de salidas digitales desde el software de la MTU.

# 5

## Conclusiones

Al finalizar el desarrollo del sistema de monitoreo, se han cumplido con los objetivos establecidos en el inicio de este proyecto, y se obtienen las siguientes conclusiones:

- Debido a la magnitud del proyecto, se aplicaron dos metodologías de desarrollo: *CMMI*, que se emplea para la gestión global del proyecto, ya que dicha metodología se emplea generalmente en proyectos *software* a gran escala; la metodología de *Sistemas Empotrados* se emplea para la división del proyecto en elementos *hardware* y *software* para la RTU y la MTU.
- Este proyecto contribuye a la implantación de mejores prácticas en la realización de proyectos de monitoreo y automatización remota, y por ende, dar pauta a futuros proyectos de innovación tecnológica e industrial, ya que estas prácticas se pueden aplicar a proyectos de automatización de procesos y plantas industriales.
- El sistema de monitoreo desarrollado consta de los siguientes elementos *hardware*: acondicionamiento de sensores (CAS), una RTU basada en el MCU ATmega8 y una interfaz de comunicación serial basada en el estándar RS-485. Los elementos *software* del sistema de monitoreo son: *software* de la RTU, *software* de la MTU y una aplicación Web.
- Se diseñaron e implementaron CAS para cada sensor. El CAS del sensor de pH y de temperatura se divide en tres etapas: aislamiento, filtrado y amplificación. Las etapas del CAS para el sensor de OD son: conversión de corriente a voltaje, amplificación y filtrado.
- Para el sensor de pH fue necesario incluir una etapa de adición de un voltaje *offset* ( $V_{offset}$ ). La etapa de aislamiento, filtrado y amplificación se implementaron con el *op-amp* CA3140. La etapa de aislamiento se realiza empleando dicho *op-amp* como seguidor de voltaje, en la etapa de filtrado y amplificación se emplea un circuito RC como filtro pasa-bajas y el CI CA3140 configurado como amplificador con una ganancia.

- Para el sensor de OD se diseñó un convertidor de corriente a voltaje y la etapa de filtrado consta de un filtro activo pasa-bajas inversor.
- La etapa de aislamiento del sensor de temperatura consiste en el CI CA3140 con la configuración de seguidor de voltaje y en la etapa de filtrado y amplificación se utiliza la configuración empleada para el sensor de pH en las respectivas etapas.
- La comunicación entre la RTU y la MTU se realiza a través de una interfaz serial RS-485. Dicha interfaz serial se implementa con el CI MAX489 y se construyó un transceptor RS-485/RS-232 para la comunicación con la MTU. La velocidad máxima de transferencia de la interfaz serial, bajo la configuración actual del MCU, fue de 38.4kbps.
- El software del MCU se desarrolló en el lenguaje de programación C. Este software se desarrolló en el IDE AVR Studio y se empleó la librería WinAVR que es un compilador de lenguaje C, ambas herramientas son específicamente para MCUs de la marca ATMEL.
- EL software de la MTU se desarrolló en el lenguaje de programación Java, el cual facilita el desarrollo de manera modular como se describe en el inciso 3.3.2.2. Java proporciona herramientas con las que se implementó el *Modelo del Gestor*, con ello se logra un mejor desempeño del *software* a través de la programación de múltiples hilos (*Multithreading*).
- A través de un búfer sincronizado, Java permite solucionar el problema *productor-consumidor* que se presenta al momento de compartir la interfaz serial entre los distintos subprocesos de monitoreo.
- El software de la MTU almacena registros de mediciones y estados digitales en una base de datos MySQL, y acepta solicitudes desde la aplicación Web para establecer salidas digitales.
- La aplicación Web fue desarrollada empleando el *framework Yii*. Para complementar la aplicación Web se utilizaron las extensiones (De Yii): *YiiBootstrap*, *HighCharts* y *YiiBooster*. También se utilizó la herramienta *Ambiance (jQuery Notification Plugin)* que es una herramienta de notificación basada en *jQuery*. Las gráficas dinámicas fueron implementadas usando las extensiones *YiiBooster*, *HighCharts* (en su versión gratuita) y *jQuery*<sup>1</sup>.
- Se describió detalladamente el diseño *hardware* y *software*, lo cual permite utilizar esta información en otros proyectos de investigación como: adquisición de datos, acondicionamiento de señales, filtrado, comunicación serial, aplicaciones que requieren múltiples procesos, bases de datos y aplicaciones Web.
- El sistema de monitoreo desarrollado da una pauta a seguir para implementar sistemas SCADA y de automatización remota. Este tipo de proyectos son fácilmente escalables y se pueden agregar al sistema múltiples RTUs, MTUs, bases de datos y aplicaciones Web. Por ende, se pueden gestionar múltiples plantas o procesos industriales sin comprometer el rendimiento, ya que el procesamiento se realiza de manera distribuida.

---

<sup>1</sup>Se emplean solicitudes asíncronas *AJAX*.

## 5.1. Líneas Futuras de Investigación

En base al desarrollo del presente proyecto de tesis y otros proyectos de investigación, se abre la posibilidad de evolucionar este proyecto de manera significativa, por lo que se proponen las siguientes posibilidades para trabajos futuros:

- Implementar en el MCU controles por realimentación como el control PID, controladores de lógica difusa entre otros. Con esto se podrán controlar, desde un lugar remoto, variables cruciales en procesos, plantas o cualquier sistema que así lo requiera (Por ejemplo: pH, temperatura, presión, volumen y otras variables).
- Implementar alarmas o disparadores que indiquen un estado activo o inactivo, lo cual se podrá monitorear como entradas digitales. Igualmente, se podrán implementar activadores de dispositivos analógicos (Motores, ventiladores, ventanas y otros) o de un sistema en particular a través de señales digitales (TTL), dichas señales se pueden implementar a través de las salidas digitales que se detallan en el diseño de este proyecto de tesis.
- Mejorar el dispositivo CAS para el sensor de pH utilizando CIs de propósito industrial, con esto se podrá reducir el margen de error en las lecturas. Por ejemplo, se puede utilizar el CI LM4140A-1.0 de la marca *National Semiconductor*, el cual proporciona un voltaje de referencia de baja caída y produce un voltaje preciso de 1.024V que puede ser utilizado como  $V_{offset}$ , desde luego con un arreglo divisor de voltaje. Esto reducirá las variaciones en  $V_{offset}$  y por ende las mediciones serán consistentes.
- Basándose en la referencia [4], las arquitecturas posibles de comunicación entre una RTU y la MTU que se pueden implementar, utilizando radio módem son: punto a punto y estaciones repetidoras. Dichas arquitecturas permiten abarcar distancias significativamente grandes entre la RTU y la MTU. En los enlaces punto a punto interactúan solo una RTU y una MTU, y generalmente consiste en sistemas con propagación en línea de vista (*Line-of-sight propagation*). Los enlaces con estaciones repetidoras brindan la ventaja de gestionar RTUs que están fuera de la línea de vista del radio módem de la MTU y por ende a una distancia mucho mayor que la arquitectura punto a punto, gracias a esto, permite abarcar distancias entre regiones y estados.
- Actualmente, debido a la gran demanda de la Internet, resulta conveniente implementar el protocolo de comunicación Modbus sobre TCP/IP. Con ello, se pueden utilizar estándares de redes LAN como Ethernet (Estándar IEEE 802.3) y Wi-Fi (Estándar IEEE 802.11) para la comunicación entre la RTU y la MTU en distancias cortas. Considerando que para una red interna o intranet la latencia es relativamente despreciable, es posible sustituir la interfaz serial RS-485 por cualquier estándar de red LAN.
- Se puede realizar un estudio para verificar si existe la posibilidad de usar la infraestructura de Internet que ofrece TELMEX al público en general, ya que TELMEX ofrece, al sector empresarial e industrial, servicios de Internet dedicado, dichos servicios tienen un costo fuera del alcance de un usuario común. Dicho estudio proporcionará los rangos de latencias de propagación en la información de monitoreo y se podrá determinar que tipo de

plantas o procesos industriales podrían ser monitoreados o automatizados bajo dicha infraestructura.

- En la actualidad, se puede implementar la RTU en MCUs que ofrecen módulos embebidos de comunicación Ethernet, tales como la familia de MCUs *PIC32MX6/7* de la marca *MICROCHIP*, los cuales soportan tasas de transferencia de datos de 10Mbps (10-BaseT), 100Mbps (10/100 Base-T) y 1000Mbps (1Gbps). En base a estos MCUs se derivan una gama de dispositivos y tarjetas de desarrollo para comunicación sobre Ethernet y Wi-Fi.
- Implementar la RTU en familias de dispositivos FPGA como: *Virtex-7*, *Spartan-6*, *Kintex-7* entre otras. Dichas familias de FPGAs proporcionan herramientas para implementar interfaces Ethernet.
- Para la aplicación Web se pueden implementar módulos de notificación vía mensajes de texto y correos electrónicos. También es posible desarrollar módulos de reportes mensuales, semanales y diarios (según se requiera). Prácticamente es imposible determinar los límites de una aplicación Web, debido a la constante evolución del software libre y privativo. Con esto se abre una nueva línea de investigación en el ámbito industrial.
- Debido a las nuevas tendencias en tecnologías de la información, es posible desarrollar aplicaciones para dispositivos móviles, lo cual abre una nueva rama de investigación en lo que se refiere a usabilidad, seguridad y gestión de usuarios.
- Finalmente, La tendencia en tecnología se dirige hacia el mercado de dispositivos móviles y computación sobre la nube (*Cloud computing*), por lo que esta área ofrece una inmensa oportunidad de innovación tecnológica.

# Bibliografía

- [1] S. Acharya and V. Pandya, "Bridge between black box and white box - gray box testing technique," *International Journal of Electronics and Computer Science Engineering*, no. ISSN 2277-1956/V2N1-175-185, january 2012.
- [2] C. Alexander and M. Sadiku, *Fundamentals of Electric Circuits*, 5th ed. McGraw-Hill Education, 2012.
- [3] Atmel Corporation, "8-bit Atmel with 8KBytes InSystem Programmable Flash," Datasheet ATmega8, ATmega8L, 2013.
- [4] D. Bailey and E. Wright, *Practical SCADA for Industry*, 1st ed. Gran Bretaña: Newnes-Elsevier, 2003.
- [5] J. Barceló Ordinas and J. Íñigo Griera, *Protocolos y aplicaciones de Internet*, 1st ed. UOC, 2008.
- [6] I. Bolaños Gómez, "Diseño espacial y estructural de una planta de tratamiento de aguas residuales para la universidad tecnológica de la mixteca," Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca, México, Tesis de licenciatura, 2006.
- [7] R. Camps Paré, L. Casillas Santillán, D. Costal Costa, M. Gibert Ginestà, C. Martín Escofet, and O. Pérez Mora, *Bases de datos*, 1st ed. UOC Formación de Posgrado, 2005.
- [8] Cisco Systems, Inc., *Internetworking Technologies Handbook*, 4th ed. Cisco Press, 2004.
- [9] R. Coughlin, F. Driscoll, and R. Gutiérrez, *Amplificadores operacionales y circuitos integrados lineales*, 5th ed. Pearson Educación, 1999.
- [10] H. Deitel and P. Deitel, *Java How to Program*, 6th ed. Prentice Hall, 2004.
- [11] ESARDA Working Groups, "Guidelines for developing unattended and remote monitoring and measurement," *ESARDA*, december 2000.
- [12] P. Espiñeira, *Las Comunicaciones en la Empresa: Normas, Redes y Servicios*. Ra-Ma, Librería y Editorial Microninfomatica, 2003.
- [13] J. Guadayol Cunill, *La automatización en la ingeniería química*, 1st ed. Univ. Politèc. de Catalunya, 2010.
- [14] W. Hayt, W. Hayt, J. Kemmerly, and S. Durbin, *Engineering Circuit Analysis*, 8th ed. MCGRAW-HILL Higher Education, 2012.

- [15] Intersil, "CA3140, CA3140A: 4.5MHz, BiMOS Operational Amplifier with MOSFET Input/Bipolar Output," Datasheet FN957.10, 2005.
- [16] Javvin Technologies Inc., *Network Protocols Handbook*, 2nd ed. Javvin, 2005.
- [17] M. Lapeña, *Tratamiento de aguas industriales: aguas de proceso y residuales*, 1st ed. Marcombo, 1989.
- [18] T. Majchrzak, *Improving Software Testing: Technical and Organizational Developments*, ser. SpringerBriefs in Information Systems. Springer, 2012.
- [19] MAXIM, "Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers," Datasheet 19-0122, 2003.
- [20] M. Á. S. Maza, *Javascript*, 1st ed., ser. Certificado de profesionalidad. Innovación y Cualificación, S.L., 2012.
- [21] Mettler Toledo, "InPro 6800 Series O2 Sensors: Instruction manual," Switzerland, Datasheet InPro 6800, 52 200 953, 2005.
- [22] MODICON, Inc., Industrial Automation Systems, "Modicon Modbus Protocol Reference Guide," Reference Guide PI-MBUS-300 Rev. J, 1996.
- [23] —, "Modbus Application Protocol Specification," Application Protocol Specification V1.1b, 2006.
- [24] —, "Modbus over Serial Line: Specification and Implementation Guide," Specification and Implementation Guide V1.02, 2006.
- [25] D. Méndez Bautista, "Sistema de comunicaciones basado en ethernet para el control de sistemas empujados," Universidad Tecnológica de la Mixteca, Huajuapán de León, Oaxaca, México, Tesis de licenciatura, 2009.
- [26] National Semiconductor Corporation, "LM35, Precision Centigrade Temperature Sensors," Datasheet TL/H/5516, December 1994.
- [27] G. O'Regan, *Introduction to Software Process Improvement*, 1st ed., ser. Undergraduate topics in computer science. Springer, 2010.
- [28] J. Park, S. Mackay, and E. Wright, *Practical Data Communications for Instrumentation and Control*, 1st ed. Newnes-Elsevier, 2003.
- [29] A. Rodríguez Penin, *Sistemas SCADA*, 2nd ed. Marcombo, 2007.
- [30] Schneider-Electric, "Solutions for water treatment plants and pumping stations, achieve energy efficiency without sacrificing performance," *Schneider Electric - Global Marketing Communications Strategy & Design*, no. ART.960526, march 2011. [Online]. Available: <http://www.schneider-electric.com.mx/documents/solutions/segment/plantas-de-tratamientos-de-agua.pdf>
- [31] M. Scholten, "LED Constant Current Source Scheme: Theory of Operation," Semiconductor Components Industries, Application Note AND8109/D, January 2003.

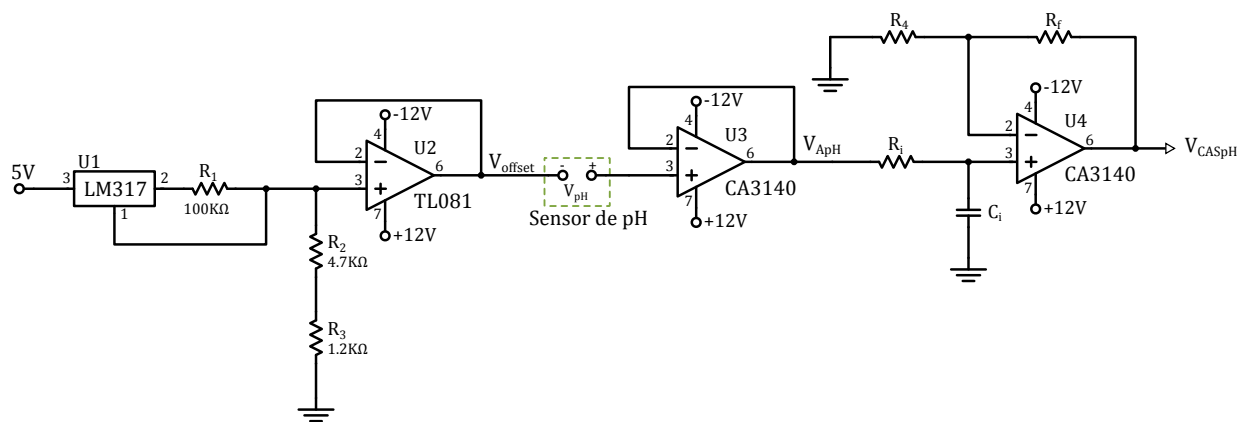


- [32] Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT), “Norma oficial mexicana nom-001-ecol-1996,” <http://biblioteca.semarnat.gob.mx/janium/Documentos/Ciga/agenda/DOFsr/60197.pdf>, 1996, pág. 70.
- [33] J. Seitz, “Application Note AN-1852: Designing with pH Electrodes,” National Semiconductor Corporation, Application Note Application Note 1852, september 2008.
- [34] R. Sheldon and G. Moes, *Beginning MySQL*, 1st ed., ser. Programmer to programmer. Wiley, 2005.
- [35] A. SILBERSCHATZ, H. KORTH, and S. SUDARSHAN, *Fundamentos de bases de datos*, 4th ed. McGraw Hill, 2002.
- [36] I. Sommerville and M. Galipienso, *Ingeniería del software*, ser. Fuera de colección Out of series. Pearson Educación, 2005.
- [37] G. Tchobanoglous, F. Burton, M. . Eddy, and H. Stensel, *Wastewater Engineering: Treatment and Reuse*, ser. McGraw-Hill Series in Civil and Environmental Engineering. McGraw-Hill Education, 2004. [Online]. Available: <http://books.google.com.mx/books?id=U9OmPwAACAAJ>
- [38] Texas Instruments, “DAC0808 8-Bit D/A Converter,” Datasheet SNAS539A, 2001.
- [39] —, “MAX232, MAX232I: DUAL EIA-232 DRIVERS/RECEIVERS,” Datasheet SLLS047I, 2004.
- [40] —, “TL081, JFET-input operational amplifier,” Datasheet SLOS081G, 2004.
- [41] P. Venkataram and S. S. Manvi, *Communication Protocol Engineering*, 1st ed. Prentice Hall, 2005.
- [42] Wikipedia, “Atmospheric pressure,” [http://en.wikipedia.org/wiki/Atmospheric\\_pressure](http://en.wikipedia.org/wiki/Atmospheric_pressure), Philips Lumileds Lighting Company, september 2013.
- [43] J. Winesett, *Web Application Development with Yii and PHP*, 2nd ed. Packt, 2012.
- [44] F. Xhafa and S. Caballé, *El Paradigma de la Computación Distribuida*, 1st ed. Publicaciones Delta, 2008, ch. 1, pp. 13–44.
- [45] Q. Xue and X. Wei Zhuo, *The Definitive Guide to Yii 1.1*, 1st ed. Yii Framework, 2012.
- [46] P. Zhang, *Industrial Control Technology*, 1st ed. William Andrew, 2008.
- [47] R. Zurawski, *Embedded Systems Handbook, Second Edition: Networked Embedded Systems*, 2nd ed., ser. Embedded Systems Handbook. Taylor & Francis, 2009.
- [48] J. Íñigo Griera and J. Barceló Ordinas, *Estructura de redes de computadores*, 1st ed. UOC, 2008.

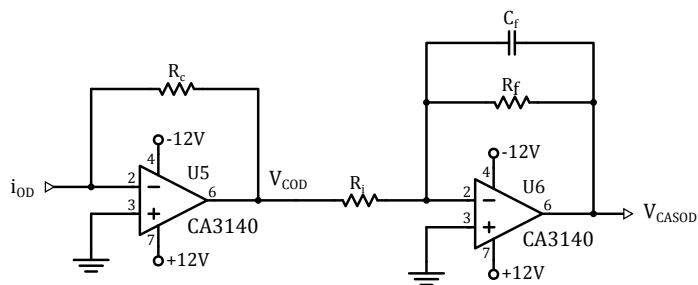


# A

## Diagramas Esquemáticos



**Figura A.1.** Diagrama esquemático del dispositivo CAS para el sensor de pH.

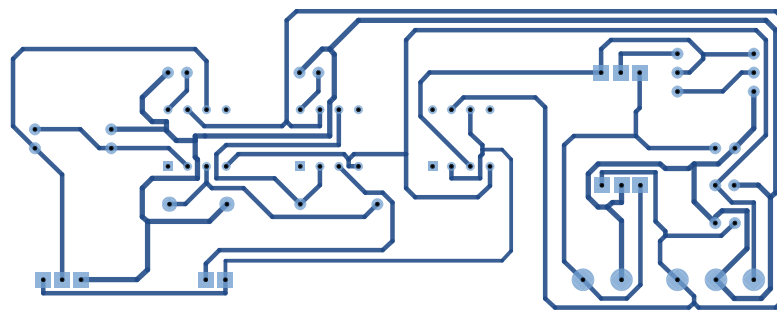


**Figura A.2.** Diagrama esquemático del dispositivo CAS para el sensor de OD.

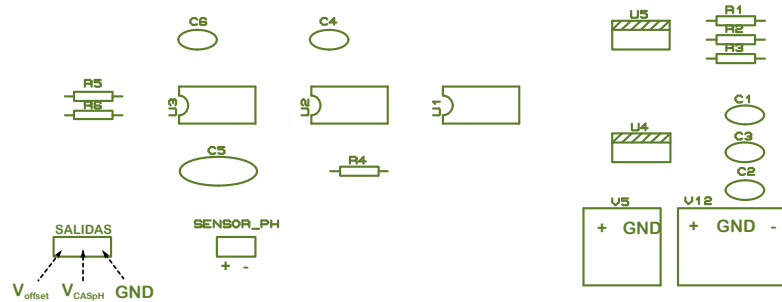


# B

## PCB

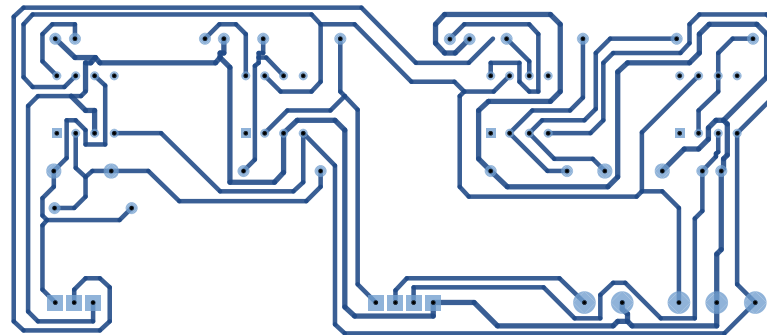


(a) Diseño de pistas PCB de capa inferior.

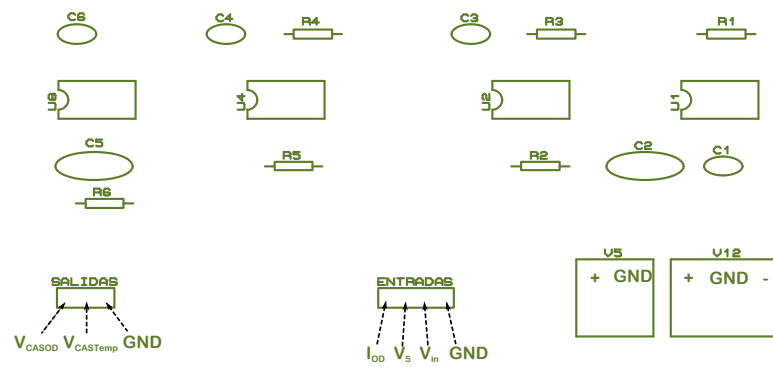


(b) Ordenamiento de CIs.

**Figura B.1.** PCB del CAS para el sensor de pH.

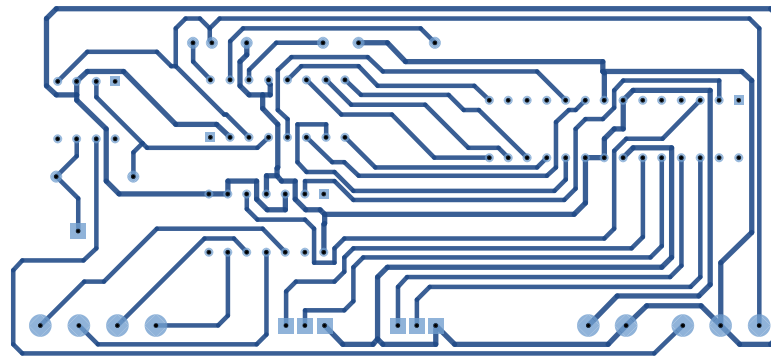


(a) Diseño de pistas PCB de capa inferior.

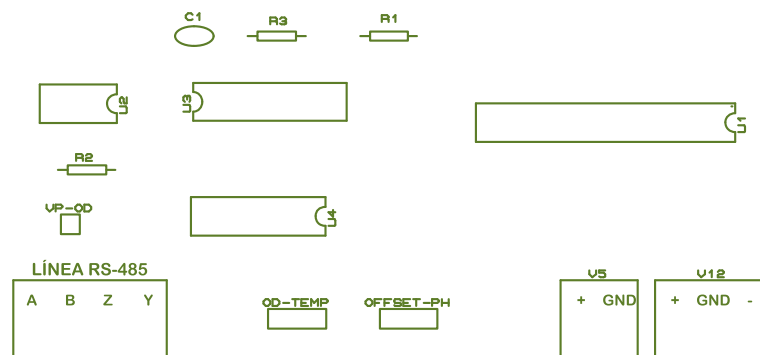


(b) Distribución de CIs.

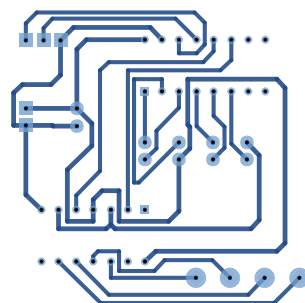
Figura B.2. PCB del CAS para el sensor de OD y temperatura.



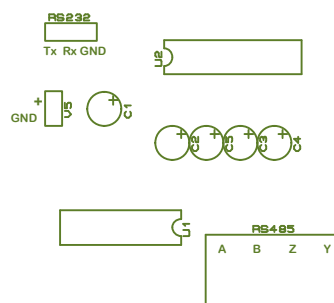
(a) Diseño de pistas PCB de capa inferior.



(b) Distribución de CIs.

**Figura B.3.** PCB para el módulo del MCU.

(a) Diseño de pistas PCB de capa inferior.



(b) Distribución de CIs.

**Figura B.4.** PCB del transceptor RS-232/RS-485.





# C

## Diseño Software

### C.1. Clases

Para una mejor comprensión de la división de subprocesos del software de la MTU (Sección 3.3.2.2) los diagramas de clase se dividen en cinco secciones: monitoreo de entradas analógicas (Figura C.1), monitoreo de entradas digitales (Figura C.2), diagrama de monitoreo y control de salidas digitales (Figura C.3) y calibración de sensores (Figura C.4).

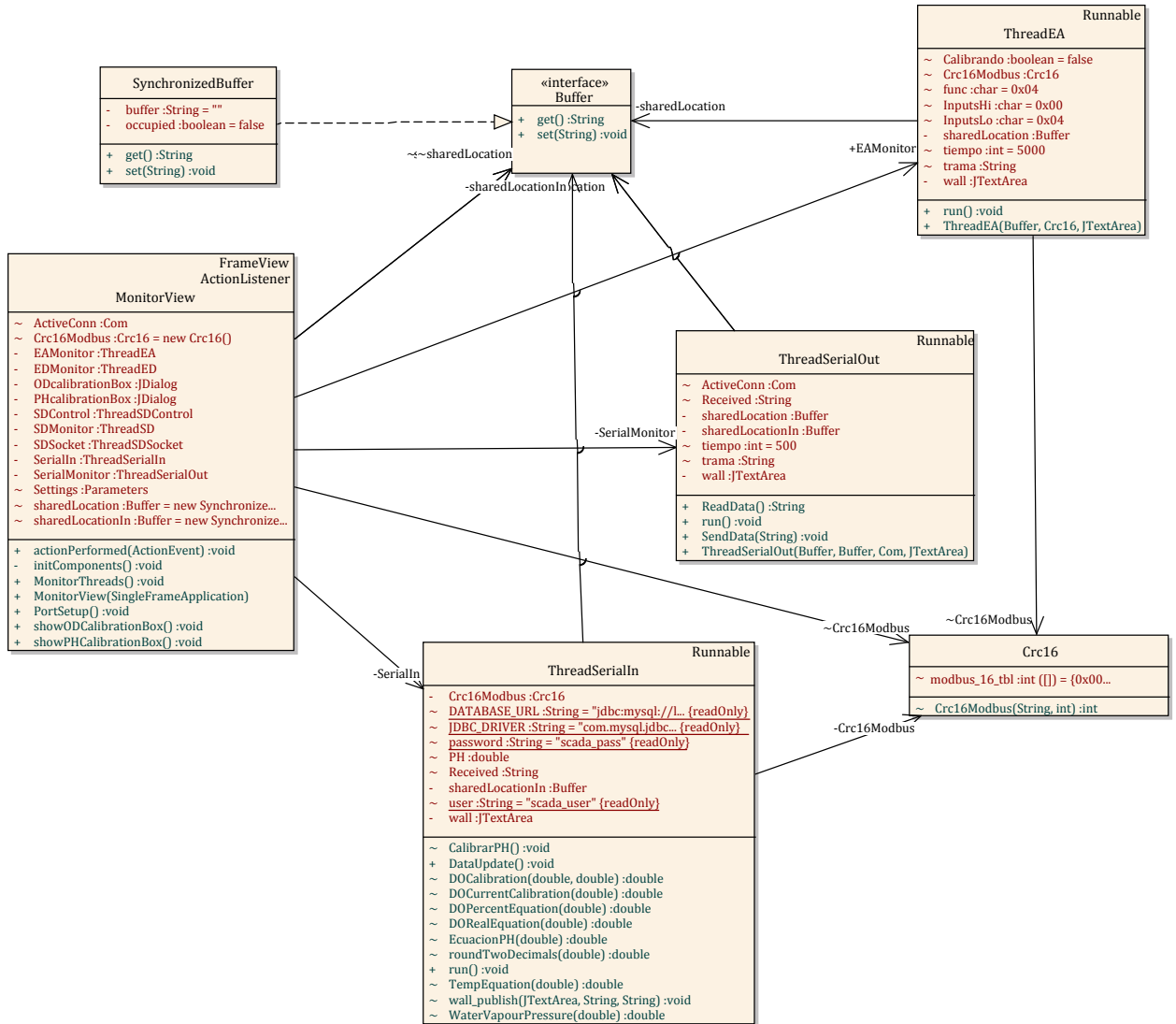


Figura C.1. Diagrama de clases para monitoreo de entradas analógicas.







### C.2. Diagramas de actividades

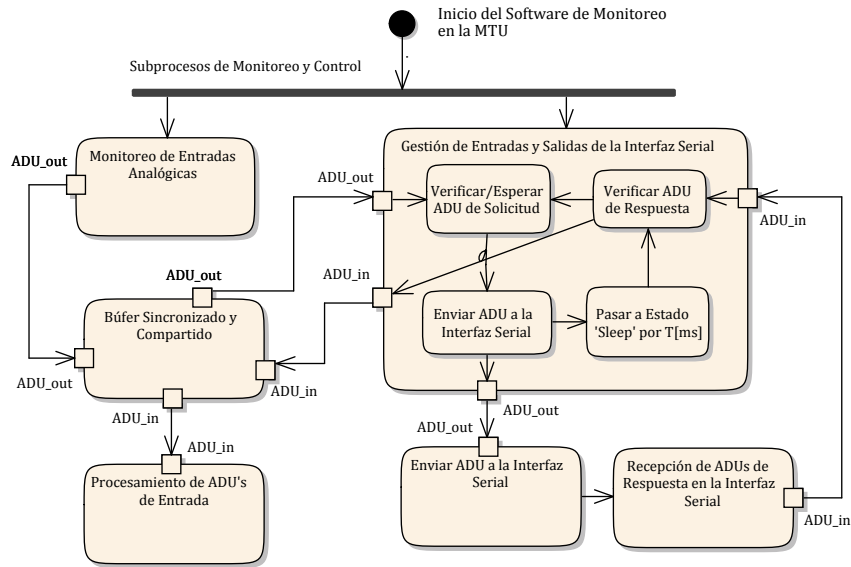


Figura C.5. Diagrama de actividades de la gestión de la interfaz serial.

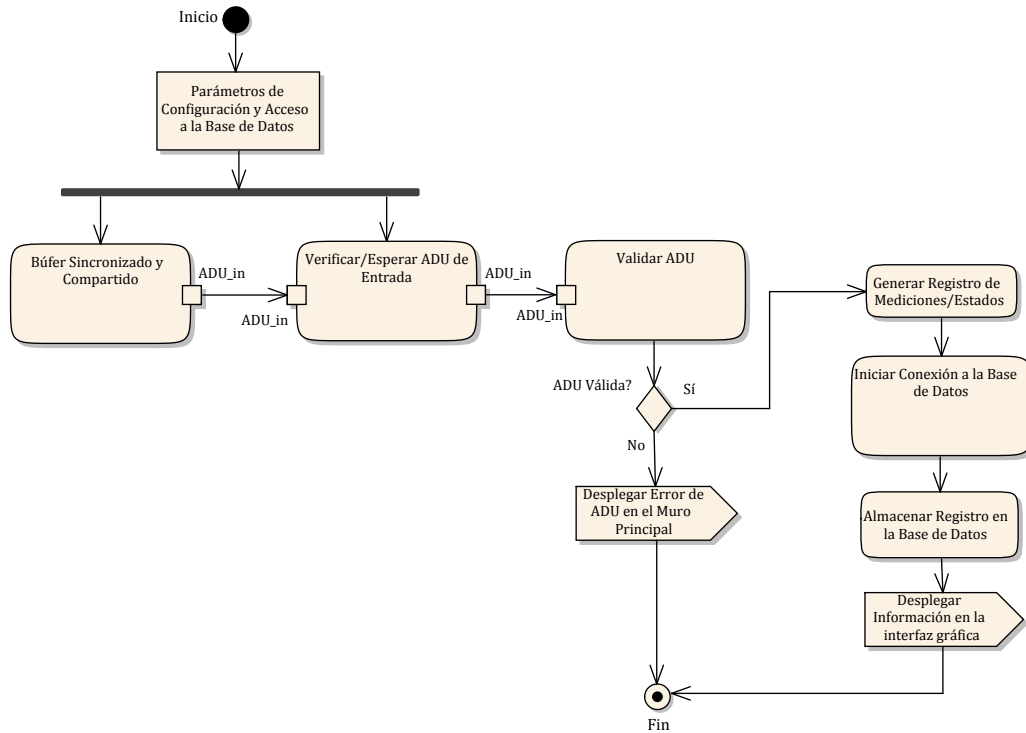


Figura C.6. Diagrama de actividades del procesamiento de ADUs.

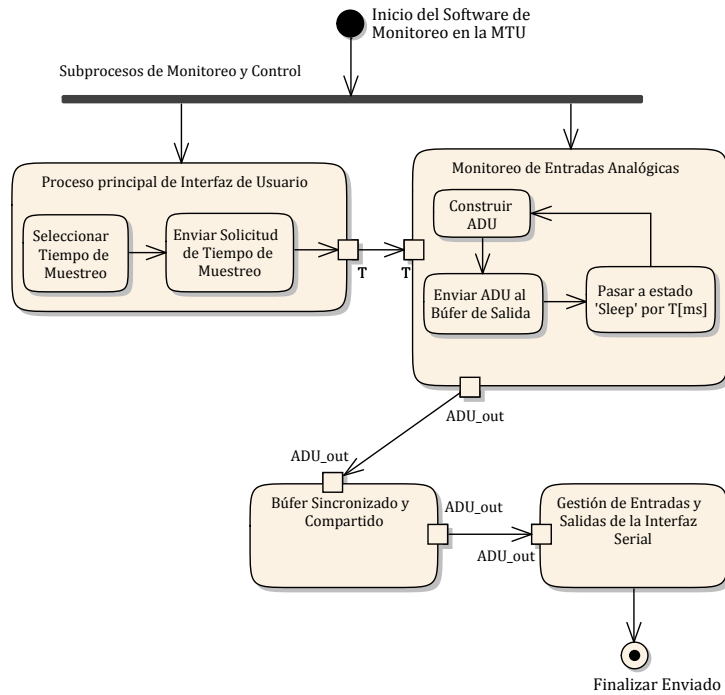


Figura C.7. Diagrama de actividades para establecer el tiempo de muestreo.

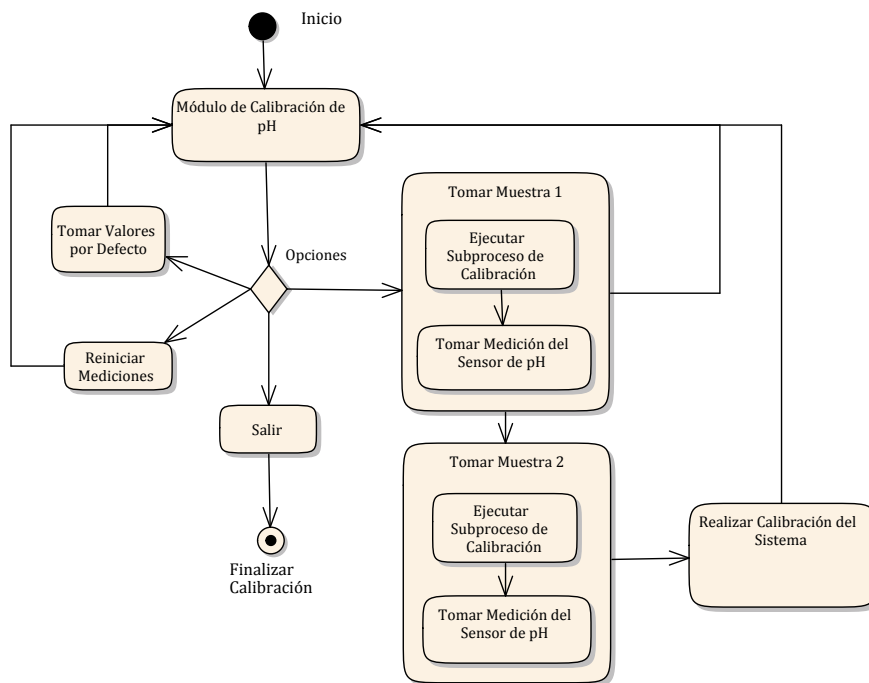
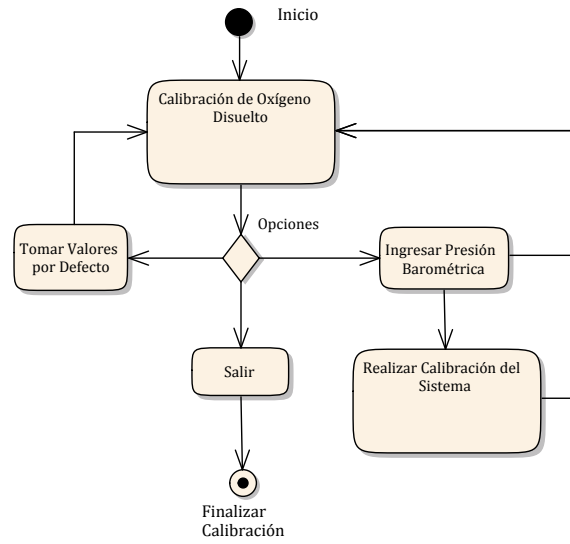


Figura C.8. Diagrama de actividades para calibrar el sensor de pH.



**Figura C.9.** Diagrama de actividades para calibrar el sensor de OD.