



Escuela de Ingeniería Electrónica

Informe de Proyecto de Graduación

Desarrollo de un sistema de comunicación redundante para las
estaciones del AYA (Instituto Costarricense de Acueductos y
Alcantarillados)

José Gabriel Carmona Rodríguez

Cartago, Costa Rica

Mayo del 2018

ESCUELA DE INGENIERÍA ELECTRÓNICA

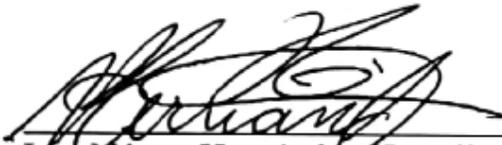
PROYECTO DE GRADUACIÓN

ACTA DE APROBACIÓN

**Defensa de Proyecto de Graduación
Requisito para optar por el título de Ingeniero en Electrónica
Grado Académico de Licenciatura
Instituto Tecnológico de Costa Rica**

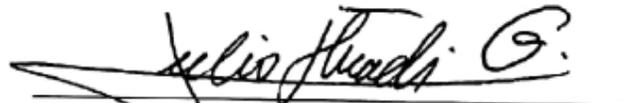
El Tribunal Evaluador aprueba la defensa del proyecto de graduación denominado Desarrollo de un sistema de comunicación redundante para las estaciones del AYA (Instituto Costarricense de Acueductos y Alcantarillados) , realizado por José Gabriel Carmona Rodríguez y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador



Ing. Néstor Hernández Hostaller

Profesor lector



Ing. Julio Stradi Granados

Profesor lector



Ing. Aníbal Coto Cortés

Profesor asesor

Declaratoria de autenticidad

Yo José Gabriel Carmona Rodríguez, cédula 6-0418-0922, declaro que el presente Proyecto de Graduación ha sido realizado en su totalidad por mi persona, utilizando y aplicando literatura referente al tema e introduciendo mis conocimientos adquiridos a lo largo de la carrera universitaria.

Respetando la propiedad intelectual, he procedido a indicar las fuentes de información en el caso de utilizarla, por medio de citas.

Por consiguiente, asumo completa responsabilidad por el contenido del informe final realizado.



José Gabriel Carmona Rodríguez

Cédula: 6-0418-0922

Resumen

Controlar y monitorear procesos a distancia es una tendencia en la industria, ya que permite a un operador tener información actualizada de uno o más procesos que se estén ejecutando de forma simultánea en sitios diferentes; pero cuando estos procesos se encuentran separados por varios kilómetros, se utilizan distintos medios para hacer llegar la información al operador, por ejemplo; enlaces de radiofrecuencia e internet.

El Instituto Costarricense de Acueductos y Alcantarillados (AYA) tiene la misión de brindar acceso al agua potable. Dicha institución cuenta con estaciones distribuidas dentro del Gran Área Metropolitana (GAM) para las cuales se utiliza un sistema de monitoreo y control para las estaciones de bombeo, rebombes, tanques de almacenamiento, plantas de tratamiento entre otros. A partir de este punto, cabe destacar que la comunicación es un pilar fundamental, y por consiguiente un sistema de comunicación robusto puede brindar mayor fiabilidad en estos importantes procesos.

En este documento muestra el diseño e implementación de un sistema de comunicación redundante, que utiliza radio e internet, el cual es transparente al sistema de comunicación actual, al utilizar un sistema de procesamiento externo, que adquiere los datos y realiza el envío de forma simultánea tanto para el canal de radio como de internet.

Palabras Claves: AYA, comunicación redundante, protocolo RS232, conexión USB a USB, telemetría, SCADA, protocolo USB, interfaz hombre máquina.

Abstract

Control and monitor remote processes is an industry trend, since make an operator get updated information from one or more running processes at the same time from different places, but when these processes are isolated a lot of kilometers between then. There are different ways to carry this information to operator. For example, radiofrequency and internet links.

The Costa Rican Institute of Aqueducts and Sewers (AYA) have the mission of bringing potable water accesses. Also, this institution counts with information station located throughout the Greater Metropolitan Area (GAM). Those station have monitor and control systems in pumped centers, water tanks, residual water treatment plants and other ones. At this point its important stand out the communication function and heavy communication systems can offer to these important processes.

This document shows the design and implementation of a redundant communication system which uses radio and internet, then it is transparent to the current communication system by using an external processing system that acquires the data and sends it simultaneously by the radio and internet channels.

Keywords: AYA, redundant communication, RS232 protocol, USB to USB connection, telemetry, SCADA, USB protocol, human machine interface.

Dedicatoria

“A mi papá José Enrique y a mi mamá Lucrecia, que me brindaron su apoyo incondicional en estos años de estudio, permitiéndome hacerme de las herramientas necesarias para luchar por un mejor futuro.

También a mi hermano Carlos Andrés quien siempre ha estado conmigo brindándome su apoyo de manera incondicional.”

Agradecimiento

Primeramente, dar gracias a Dios por brindarme la sabiduría y la iluminación para salir adelante con los proyectos y metas que me he propuesto hasta el día de hoy.

Un agradecimiento a mi familia por brindarme apoyo en estos años de esfuerzo y preparación académica.

También a mis compañeros del TEC con los cuales fueron parte importante de mi formación en estos años con valores importantes como respeto, compañerismo y paciencia.

Además, agradecer al personal del AYA que me brindaron la oportunidad y el apoyo para realizar este proyecto también al Ing. Douglas González Alfaro y al Ing. Raúl Barrantes Domínguez, por la ayuda brindada durante el desarrollo de este proyecto.

Índice General

Capítulo 1: Introducción.....	12
1.1 Entorno del proyecto	12
1.2 Definición del Problema	13
1.2.1 Generalidades	13
1.2.2 Síntesis del Problema	15
1.3 Enfoque de la Solución	15
Capítulo 2: Meta y Objetivos.....	17
2.1 Meta.....	17
2.2 Objetivo General	17
2.3 Objetivos Específicos.....	17
Capítulo 3: Marco Teórico	18
3.1 Protocolo USB.....	18
3.2 Protocolo RS232	18
3.3 MODBUS	19
3.4 UART	20
3.5 PL-2303HX	21
3.6 Controlador Lógico Programable (PLC)	22
3.7 Interfaz Hombre Máquina (HMI).....	23
3.8 Sistema Empotrado.....	25
3.9 Telemetría.....	25
Capítulo 4: Diseño y Desarrollo de la Solución.....	26
4.1 Análisis del sistema actual	26
4.2 Análisis y Selección de la Solución	29
4.2.1 Solución 1: Realizar el procesamiento de los datos en el servidor y en el PLC.....	29
4.2.2 Solución 2: Realizar el Procesamiento de los datos en un sistema externo al servidor y en el PLC.....	31
4.2.3 Solución 3: Realizar el procesamiento de los datos en un sistema externo al servidor y externo al PLC	32
4.3 Desarrollo del Diseño.....	33

4.3.1 Criterios de diseño raspServer y raspCliente.....	35
4.3.2 Diseño raspServer y raspCliente	36
4.3.3 Etapa de procesamiento de los datos.....	37
4.3.4 Etapa de Envío	38
4.3.5 Etapa de Recepción.....	38
4.3.6 Etapa de comparación y escritura.....	39
4.3.7 Etapa de comprobación de red	39
4.3.8 Etapa de comprobación de errores.....	39
Capítulo 5: Resultados y Análisis	41
5.1 Comprobación del sistema con simulador	41
5.2 Comprobación de sistema	43
5.2.1 Lectura y escritura de información en el servidor	43
5.2.2 Envío y recepción sistema redundante	46
Capítulo 6: Conclusiones	49
Capítulo 7: Recomendaciones	50
Referencias	51
Apéndice 1	53
Tabla Completa de Frecuencias Estaciones AYA.....	53
Apéndice 2	55
Apéndice 3	57
Apéndice 4	58

Índice de Figuras

Figura 3.1: Diagrama de bloques de una UART.....	20
Figura 3.2: Diagrama de bloques para el PL2303HX	21
Figura 3.3: Diagrama de Conexión USB a USB.....	22
Figura 3.4: Estructura Básica de un PLC	23
Figura 3.5: Esquema básico de un sistema SCADA	24
Figura 4.1: Diagrama de bloques transmisión y recepción de datos CCO	27

Figura 4.2: Diagrama de bloques recepción y transmisión de datos desde la Estación	27
Figura 4.3: Configuración de la HMI para la Zona 1	28
Figura 4.4: Diagrama de Bloques del nuevo sistema de transmisión y recepción CCO	33
Figura 4.5: Diagrama de bloques del nuevo sistema de recepción y transmisión de datos desde la Estación	34
Figura 4.6: Trama de Envío de Datos	37
Figura 5.1: Simulación de datos enviados y recibidos por el maestro	42
Figura 5.2: Simulación de datos recibidos y enviados por el esclavo.....	42
Figura 5.3: Monitor de comunicaciones DAQFactory estaciones	44
Figura 5.4: Datos adquiridos en la raspServer desde el servidor	45
Figura 5.5: Datos adquiridos en la raspServer hacia el servidor	46
Figura 5.6: Monitor de comunicaciones DAQFactory con prioridad en la estación Pavas	47
Figura 5.7: Datos codificados para envío por radio e internet	47
Figura 5.8: Datos enviados desde la raspCliente	48
Figura A2.1: PLC RFScada 4	55
Figura A2.2: PLC RFScada 16	55
Figura A2.3: PLC RFScada 32	56
Figura A2.4: PLC RFScada 16V3.1	56
Figura A3.1: Notificación de error en Radio.....	57
Figura A3.2: Notificación de error en Internet.....	57
Figura A4.1: Estación Ipís	58
Figura A4.2: Estación Curridabat	58
Figura A4.3: Estación La Finca	59
Figura A4.4: Estación Pavas	59

Índice de Tablas

Tabla 1.1: Frecuencias de algunas estaciones AYA	14
Tabla 3.1: Distribución de Pines DB9.....	19
Tabla 4.1: Frecuencias de los radios maestros en el CCO.	26
Tabla 4.2: Estaciones y frecuencias de la Zona 1	27
Tabla 5.1: Respuesta del sistema ante fallos en alguno de los enlaces de comunicación	43
Tabla A1.1: Frecuencias de Estaciones AYA	54

Capítulo 1: Introducción

1.1 Entorno del proyecto

El departamento de Radiocomunicación y Telemetría ubicado en el plantel del Instituto Costarricense de Acueductos y Alcantarillados (AYA), calle 44 la Uruca San José Costa Rica, es encargado de monitorear y tomar acciones en el funcionamiento de pozos, tanques, plantas de tratamiento de aguas residuales y potables, bombeos y rebombeos, las cuales utilizan un sistema de control conformado por un PLC, actuadores, solenoides, sensores entre otros, y un sistema de comunicaciones el cual se encuentra conformado por un sistema de radio. Actualmente se encuentran 82 de estas estaciones, las cuales se encuentran distribuidas por el Gran Área Metropolitana (GAM).

Algunas de las estaciones se encuentran en zonas alejadas de la ciudad, en lugares de difícil acceso y en los que no se cuenta con suministro eléctrico, por lo que es necesario utilizar celdas solares para alimentar los sistemas eléctricos, lo que plantea una limitante en el consumo de potencia. Otras estaciones se encuentran ubicadas en zonas con alto índice de delincuencia por lo que se tiene que instalar los equipos en torres para que no los hurten.

La comunicación con las estaciones se realiza con señales de radio, con una comunicación full dúplex, la cual como se conoce permite transmitir y recibir información simultáneamente. Este tipo de equipos de radio son sensibles ante interferencias generadas tanto por el medio ambiente como por otros equipos de radio. Este enlace estaba en la banda de 450-470MHz, pero por motivos del apagón analógico, que consiste en un reacomodo del espectro radioeléctrico, la banda que estaban utilizando fue asignada a otro propósito lo que generó interferencia con los equipos de comunicación, viéndose obligados a cambiar a la banda de 420-450MHz para poder seguir en operación.

Los PLC utilizados en las estaciones son conocidos como sistemas SCADA, los cuales utilizan una Interfaz Hombre Máquina (HMI). Para realizar el monitoreo y el control de las estaciones se utiliza el software DAQFactory.

El Centro de Control Operacional (CCO) se encarga de mantener en equilibrio los niveles de los tanques del acueducto por medio de la apertura y cierre de las válvulas o el encendido y apagado de bombas, por lo que el desconocimiento de esta información puede generar desde un desabastecimiento de este importante líquido, hasta una situación que puede poner en riesgo la vida de las personas, ya que los rebalses de algunos tanques han inundado sus alrededores, dañado plantaciones y los cimientos de estructuras, generando gastos de indemnización para institución, esto sin contar con el posible daño a los equipos, ya que si una bomba permanece encendida con un nivel bajo de agua en el tanque puede dañarse.

1.2 Definición del Problema

1.2.1 Generalidades

Las estaciones de control se encuentran compuestas por dos módulos, de control y captura de datos con un Controlador Lógico Programable (PLC) y de comunicaciones con un radio. El PLC recibe la información proveniente de la HMI enviada por el CCO por medio del módulo de comunicaciones para activar o desactivar las bombas y enviar las señales provenientes de los sensores instalados en las estaciones. Esto pone la comunicación como el pilar central, ya que sin las instrucciones enviadas por el CCO el PLC no realizaría ninguna acción. Esto se debe a que utilizan el protocolo MODBUS esclavo, lo que implica que espera a que el máster en este caso el CCO le envíe la orden para realizar una acción ya sea para capturar o escribir los datos.

El módulo de comunicaciones emplea señales de radio, utilizando una frecuencia para transmitir y otra para recibir la información; además se divide en 3 zonas las cuales son definidas de acuerdo con la distancia que se encuentran del CCO. En la tabla 1 se muestran algunas de las distintas estaciones, así como las frecuencias y las zonas de estas.

Nombre del Sitio	RF-1.2-TX	RF-1.1-RX
Bello Horizonte	429,08750 MHz	424,08750 MHz
Pavas	429,08750 MHz	424,08750 MHz
Tanques del Sur	429,08750 MHz	424,08750 MHz
Maiquetía	429,08750 MHz	424,08750 MHz
San Juan de Dios	429,08750 MHz	424,08750 MHz
Tanque La Pelota	429,08750 MHz	424,08750 MHz
Bombeo La Pelota	429,08750 MHz	424,08750 MHz
Tanque Higuito	429,08750 MHz	424,08750 MHz
Tanque Los Guidos	429,08750 MHz	424,08750 MHz
Tanque Curio	429,08750 MHz	424,08750 MHz
Tanque Lajas	429,08750 MHz	424,08750 MHz
Tanque Villacon	429,08750 MHz	424,08750 MHz
Tanque Pozos	429,08750 MHz	424,08750 MHz
Tanque Cerro Minas	429,08750 MHz	424,08750 MHz
Repetidor Sacramento-Uruca	429,08750 MHz	424,08750 MHz
Repetidor Sacramento-Escazú	424,62500 MHz	429,62500 MHz
Matra	429,62500 MHz	424,62500 MHz
Pozo Belén 01	429,62500 MHz	424,62500 MHz
Zoológico	429,62500 MHz	424,62500 MHz
Ciudad Colon	429,62500 MHz	424,62500 MHz
Tanque Salitral	429,62500 MHz	424,62500 MHz
Tanque Santa Ana	429,62500 MHz	424,62500 MHz
Tanque Piedades	429,62500 MHz	424,62500 MHz
Potrerosillos	429,62500 MHz	424,62500 MHz
CNP 1-2	429,62500 MHz	424,62500 MHz
CNP 3-4	429,62500 MHz	424,62500 MHz
CNP 5-6	429,62500 MHz	424,62500 MHz
CNP 7-8	429,62500 MHz	424,62500 MHz
CCO AYA	424,08750 MHz	429,08750 MHz
CCO AYA	424,23750 MHz	429,23750 MHz
CCO AYA	424,45000 MHz	429,45000 MHz
Planta Sitios	429,45000 MHz	424,45000 MHz
Tanque Sitios	429,45000 MHz	424,45000 MHz
La Finca	429,45000 MHz	424,45000 MHz

Tabla 1.1: Frecuencias de algunas estaciones AYA

En el Apéndice 1 se puede encontrar la tabla completa con las frecuencias de todas las estaciones.

Este tipo de señales son sensibles ante interferencias que pueden ser generadas por diversos factores tanto climáticos, geográficos (dependiendo de la zona en la que se encuentre la estación) y generados por otros sistemas de telecomunicación, por lo que implementar un sistema de comunicación redundante permite tener más caminos para realizar la comunicación, utilizando el enlace de red actual en conjunto con un enlace de internet.

1.2.2 Síntesis del Problema

Los pozos, tanques, plantas de tratamiento de aguas residuales y potables, bombes y rebombes del AYA, utilizan una HMI para activar o desactivar bombas, abrir o cerrar válvulas y visualizar las variables físicas como caudales, presiones y niveles. La adquisición de los datos y el control se realiza con un PLC, la transmisión de esta información se realiza utilizando un enlace de radio, pero por interferencias el sistema puede perder la comunicación o entregar datos erróneos.

1.3 Enfoque de la Solución

Implementar un sistema de comunicación redundante que envíe la información por radio y por internet entre las estaciones remotas y la estación central del AYA. Se pretende aumentar la fiabilidad del sistema y se le agrega la capacidad de enviar una notificación si existiese un fallo en el enlace de comunicación ya sea del radio o de internet.

Al enviarse la información por ambos canales, es necesario que esta se prepare asignándole un identificador para que en la etapa de recepción el sistema

sepa si la información ya había llegado o no y también para identificar por cuál de los canales fue recibida la información, lo que permite saber que los canales se encuentran en funcionamiento.

La HMI envía la información a los puertos serie (COM) del servidor utilizando un puerto por cada una de las zonas definidas y luego esta información es enviada a los radios maestros. En el punto de recepción el radio esclavo recibe la información y la escribe directo en el PLC, esta comunicación entre el PLC y el servidor es transparente.

En lo que respecta a la conexión inalámbrica de Internet se va a utilizar la conexión a la red brindada por el AYA que está operativa, al igual que los enlaces de radio.

Al ser 82 estaciones distribuidas por todo el GAM y por motivos de la duración del proyecto se limita a realizar el diseño de un prototipo funcional para una estación.

Capítulo 2: Meta y Objetivos

2.1 Meta

Desarrollar un sistema de comunicación que permita al departamento de Radiofrecuencia y Telemetría del AYA asegurar que la comunicación entre las estaciones y el Centro de Control Operacional sea fiable.

2.2 Objetivo General

Desarrollar un sistema de comunicación redundante libre de errores.

2.3 Objetivos Específicos

Diseñar módulo que realice una comunicación con redundancia y transparencia para el sistema actual.

Diseñar un módulo que reciba los datos provenientes de los dos enlaces de comunicación almacenando el que llegue primero para realizar su ejecución.

Diseñar módulo que verifique la existencia de algún problema en los canales de comunicación y envíe una notificación de dicha falla.

Capítulo 3: Marco Teórico

3.1 Protocolo USB

El protocolo USB (Universal Serial Bus) es un protocolo que nace con la idea de simplificar la cantidad de conectores existentes en una PC, simplificando la conexión y configuración de los dispositivos. [1]

Al ser un bus de punto a punto, cuenta con un lugar de inicio: el Host, dispositivo maestro que inicia la comunicación, en este caso la PC y el punto de destino el cual puede ser un periférico o un hub, dispositivo con uno o más conectores o conexiones internas hacia otros dispositivos[2] . Al ser este protocolo de arquitectura maestro esclavo, no es posible realizar una comunicación entre dos hosts o dos periféricos.

3.2 Protocolo RS232

Es una forma del protocolo serie que utiliza tensiones de $\pm 15V$ y además posee otras líneas de control de flujo (Hand-shake) que son de uso opcional dependiendo del dispositivo. [4]

Para poder realizar la comunicación entre 2 puertos, estos deben tener las mismas características como lo es la velocidad de transmisión (Bits por segundo que transfieren, baud rate), bits de datos (cantidad de bits en el paquete de transmisión), bits de parada (indica el fin de comunicación de un solo paquete) y paridad (permite verificar si hay errores en la transmisión). [3]

Utiliza el conector DB9, el cual cuenta con la distribución de pines que se muestra en la tabla 3.1. [4]

Señal	No. Pin
Carrier Detect	1
Receive Data	2
Transmit Data	3
Data Terminal Ready	4
Ground	5
Data Set Ready	6
Request to Send	7
Clear to Send	8
Ring Indicator	9

Tabla 3.1: Distribución de Pines DB9 [4]

3.3 MODBUS

Es un protocolo de solicitud-respuesta, el cual utiliza la relación maestro esclavo, para este tipo de relación la comunicación se produce en pares, lo que implica que un dispositivo tiene que iniciar una solicitud (el dispositivo maestro usualmente una HMI) y luego esperar una respuesta del dispositivo esclavo (usualmente el PLC). [5]

3.4 UART

Universal Asynchronous Receiver/Transmitter (Receptor/Transmisor Asíncrono Universal) es el dispositivo en un sistema de comunicaciones serial, su función principal es convertir los datos serie a paralelo en el caso de la recepción y convertir los datos de paralelo a serie para la transmisión tal y como se observa en la figura 3.1.

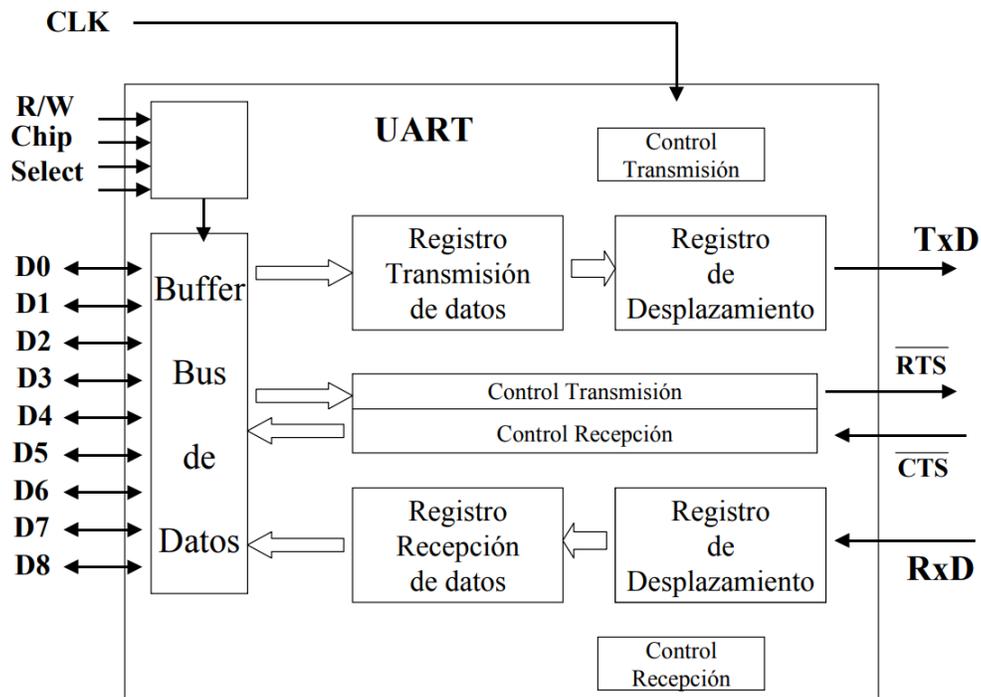


Figura 3.1: Diagrama de bloques de una UART.

3.5 PL-2303HX

Este chip tiene la capacidad de realizar la conversión de protocolo serial a USB y viceversa, por lo que permite tener un comportamiento de UART en cualquier puerto COM. Además, aprovecha el modo de transferencia masiva USB, los grandes buffers de datos y el control de flujo automático, por lo que se pueden mejorar las velocidades de transmisión. En la figura 3.2 se puede observar el diagrama de bloques para el PL-2303HX.

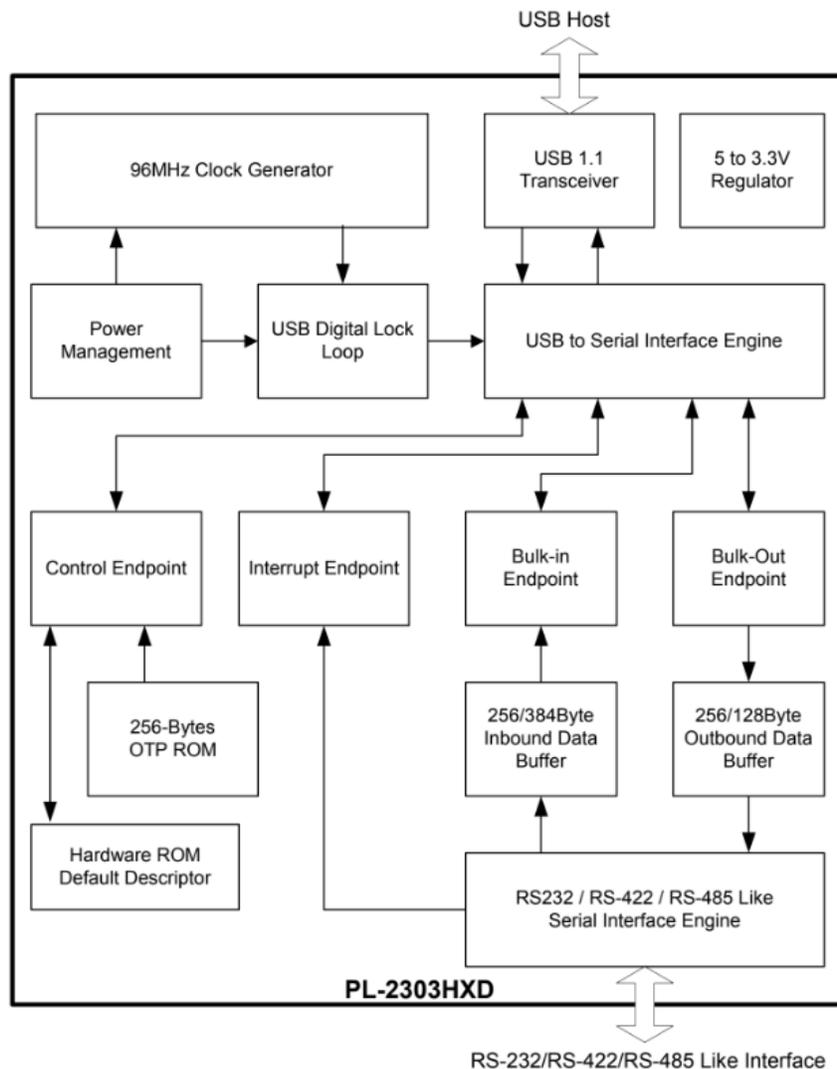


Figura 3.2: Diagrama de bloques para el PL2303HX [7]

Si se quisiera realizar una conexión entre dos computadoras por medio de USB se tendrían conflictos con el protocolo tal como se explica en el punto 3.1, si se utiliza un chip como el PL-2303HX se logra que cada puerto USB de la computadora pueda realizar una comunicación directa como una UART, de este modo para obtener una comunicación entre ambos se realiza la conexión que se muestra en la figura 3.3.

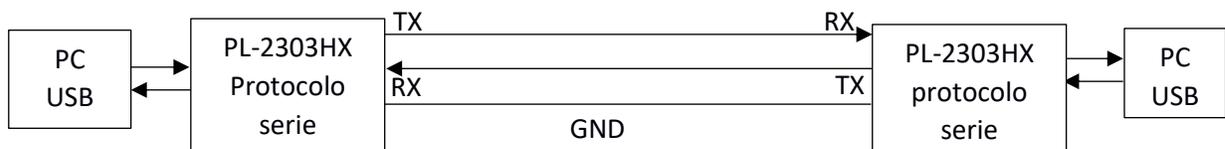


Figura 3.3: Diagrama de Conexión USB a USB

De esta forma se logra de manera se logra simplificar la forma en la que se comparte información de computadora a computadora utilizando los puertos USB. Utilizando un adaptador USB a DB9, según la tabla 3.1 se utilizan los pines los pines: TXD (Transmit Data), RXD (Receive Data) y GND (Ground).

3.6 Controlador Lógico Programable (PLC)

Estos controladores son utilizados en ambientes industriales donde la decisión y la acción deben ser tomadas en forma muy rápida, para responder en tiempo real. Los PLC son utilizados donde se requieran tanto controles lógicos como secuenciales o ambos a la vez [10]. En la figura 3.4, se puede observar la estructura básica de un PLC.

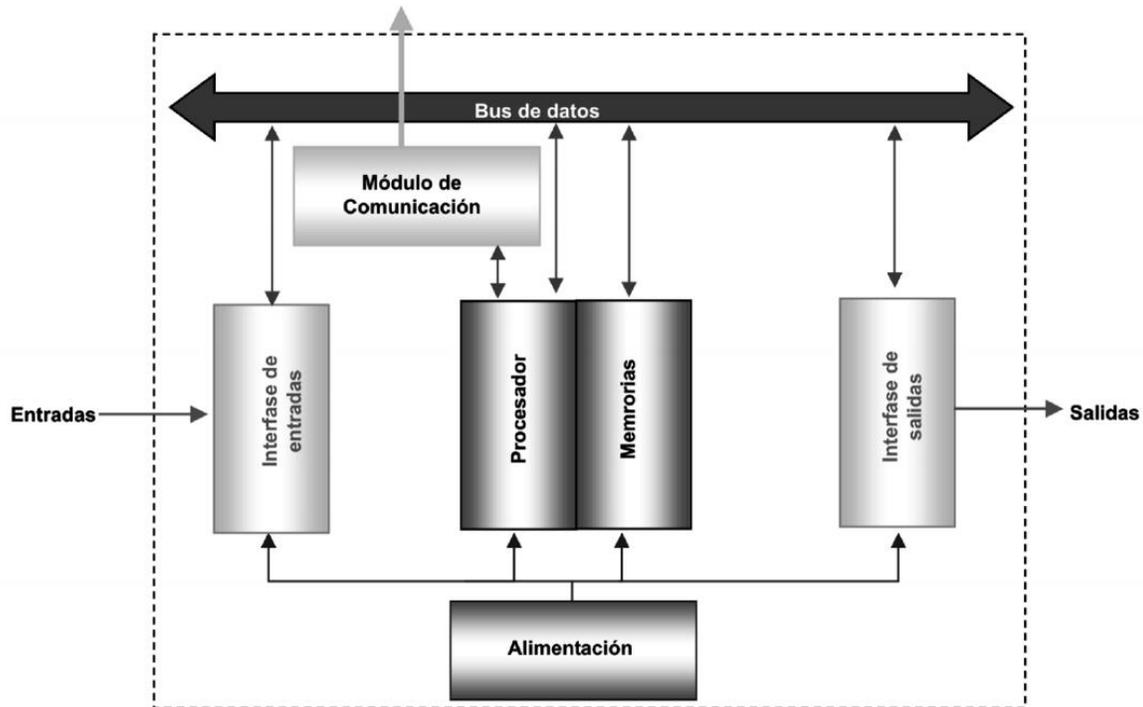


Figura 3.4: Estructura Básica de un PLC [10]

Los PLC utilizan protocolo de comunicación MODBUS con interfaces: RS232, RS485 o TCP/IP dependiendo del tipo de PLC. Las interfaces de entrada pueden ser analógicas o digitales, al igual que las de salida, que pueden ser PWM (Analógicas) o relés (Digitales).

3.7 Interfaz Hombre Máquina (HMI)

Es una interfaz de usuario en un sistema de control de fabricación o algún proceso, la cual proporciona una visualización gráfica del mismo [12]. Principalmente realizan la función de traducir las variables complejas de un proceso en información útil y aprovechable [13].

Una forma de utilizar la HMI es el sistema de Supervisión, Control y Adquisición de Datos (SCADA), que es el conjunto de software y hardware que sirven para poder comunicar, controlar y supervisar diversos dispositivos de campo, así como controlar de forma remota todo el proceso. Además, se suele incorporar

una HMI que proporciona un control del proceso mucho más intuitivo y rápido tomando en cuenta las prestaciones que una interfaz tiene. Todo esto, tiene como fin ayudar a los operarios y supervisores, otorgándoles un mejor control y la posibilidad de realizar cambios de forma prácticamente inmediata [14].

Entre las principales características de un sistema SCADA se encuentran [14]:

- Adquisición y Almacenamiento de datos.
- Representación gráfica de las variables.
- Ejecución de acciones de control para modificar.
- Conectividad con otras aplicaciones y base de datos.
- Arquitectura abierta y flexible.
- Supervisión.
- Transmisión de información

La figura 3.7 muestra un esquema básico de la composición de un sistema SCADA.

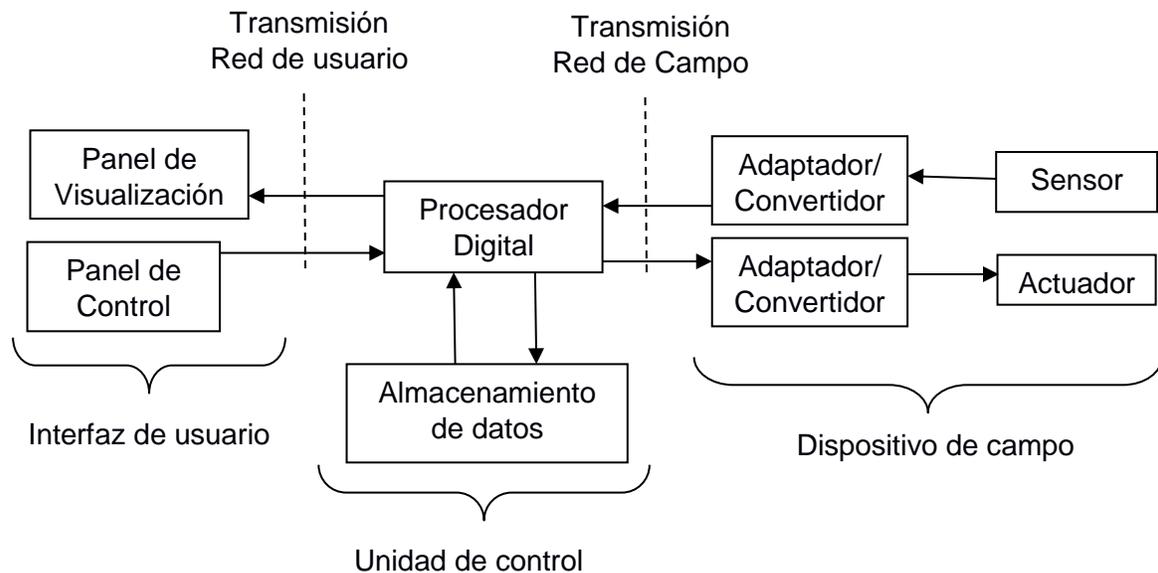


Figura 3.5: Esquema básico de un sistema SCADA [15]

3.8 Sistema Empotrado

Circuito electrónico digital capaz de realizar operaciones de computación, generalmente en tiempo real, que sirven para cumplir una tarea específica, los cuales, a diferencia de los sistemas de cómputo, suelen tener recursos mucho más limitados y aplicaciones específicas [16]. La arquitectura de un sistema embebido contiene un microprocesador dedicado capaz de ejecutar instrucciones a una determinada velocidad, controlada por la señal de reloj, así como almacenamiento limitado [16]. Entre sus principales características es el bajo consumo de energía [16].

3.9 Telemetría

Es una de las áreas de la ingeniería que está orientada a la medición de cualquier cantidad física, utilizando interfaces electrónicas que conectadas a través de alguna línea de transmisión ya sea un medio guiado o no guiado permiten enviar la información a un centro de gestión [17]. La radio frecuencia en sistemas de telemetría es un ejemplo, ya que utilizan de radios de corto alcance y bajo ancho de banda, que son más económicos y tienen un menor consumo de energía [18].

Capítulo 4: Diseño y Desarrollo de la Solución

4.1 Análisis del sistema actual

El AYA cuenta con el DAQFactory, que es un sistema SCADA, instalado en un servidor en el plantel de la Uruca, el cual utilizando telemetría controla distintas estaciones distribuidas por el GAM las cuales utilizan enlaces de radiofrecuencia para lograr una conexión entre la Uruca y las estaciones.

Los tele-comandos se envían por los puertos COM del servidor a un radio maestro, el cual envía esos datos a todos los radios esclavos que se encuentren en su frecuencia, todos reciben la misma información, pero solo el que tiene el PLC al cual se le envió el tele-comando realiza la acción. En la tabla 4.1 se puede observar las frecuencias de los radios maestros.

Nombre del Sitio	RF-1.2-TX	RF-1.1-RX
CCO AYA	424,08750 MHz	429,08750 MHz
CCO AYA	424,23750 MHz	429,23750 MHz
CCO AYA	424,45000 MHz	429,45000 MHz

Tabla 4.1: Frecuencias de los radios maestros en el CCO.

Tabla de frecuencias en Apéndice 1

Es importante mencionar que existen 3 radios maestros por motivo de sectorización debido a distancias de las estaciones al CCO. Cada uno de estos radios obtiene la información de puertos COM diferentes. En la figura 4.1 se observa un diagrama de bloques del funcionamiento del transmisión y recepción de datos desde el CCO hacia cualquier estación en la Zona 1, las cuales pueden observar en la tabla 4.2. En la Figura 4.2 se observa un diagrama de bloques de la recepción y transmisión de datos desde la estación.

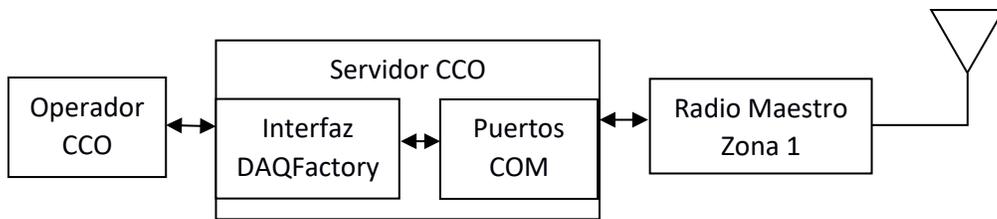


Figura 4.1: Diagrama de bloques transmisión y recepción de datos CCO

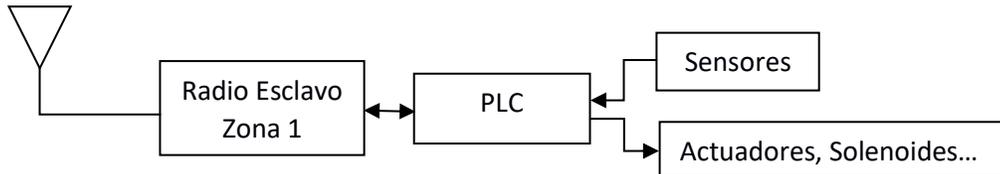


Figura 4.2: Diagrama de bloques recepción y transmisión de datos desde la Estación

Nombre del Sitio	RF-1.2-TX	RF-1.1-RX
Bello Horizonte	429,08750 MHz	424,08750 MHz
Entierrillos	429,08750 MHz	424,08750 MHz
Naranjal	429,08750 MHz	424,08750 MHz
Cruz Roja Escazú	429,08750 MHz	424,08750 MHz
Tanque Psiquiátrico	429,08750 MHz	424,08750 MHz
Tanq-Bomb El Carmen	429,08750 MHz	424,08750 MHz
Tanq-Bomb Corazón de Jesús	429,08750 MHz	424,08750 MHz
Tanque San Antonio Escazú	429,08750 MHz	424,08750 MHz
Planta Alajuelita	429,08750 MHz	424,08750 MHz
Calle Nueva	429,08750 MHz	424,08750 MHz
Bello Horizonte	429,08750 MHz	424,08750 MHz
Pavas	429,08750 MHz	424,08750 MHz
Tanques del Sur	429,08750 MHz	424,08750 MHz
Maiquetía	429,08750 MHz	424,08750 MHz
San Juan de Dios	429,08750 MHz	424,08750 MHz
Tanque La Pelota	429,08750 MHz	424,08750 MHz
Bombeo La Pelota	429,08750 MHz	424,08750 MHz
Tanque Higuito	429,08750 MHz	424,08750 MHz
Tanque Los Guidos	429,08750 MHz	424,08750 MHz
Tanque Curio	429,08750 MHz	424,08750 MHz
Tanque Lajas	429,08750 MHz	424,08750 MHz
Tanque Villacon	429,08750 MHz	424,08750 MHz
Tanque Pozos	429,08750 MHz	424,08750 MHz
Tanque Cerro Minas	429,08750 MHz	424,08750 MHz
Repetidor Sacramento-Uruca	429,08750 MHz	424,08750 MHz
CCO AYA	424,08750 MHz	429,08750 MHz

Tabla 4.2: Estaciones y frecuencias de la Zona 1

La comunicación entre la HMI y los PLC es “transparente”, esto implica que para la HMI y el PLC les es “indiferente” el medio por el que llegue la información.

Como se observa en la tabla 4.2, en la Zona 1 hay 24 estaciones, lo que implica que hay 24 radios esclavos recibiendo datos, pero tomando en cuenta que los PLC responden ante el protocolo MODBUS, solo el PLC que se le solicitó información va a responder. En la figura 4.3 se muestra cómo se encuentra la configuración de la Zona 1 en la HMI.

Site name : AyAGAMsvr Program Version: 2017
Generic Version: Generic 5.0

Zona 1 Poll
Port Zona 1:
BaudrateZona 1:

ZONA 1 **STOP** **RESTART**

Estacion	Ultima-Lectura	Tiempo-Poll	Tipo
Unit 2: Unit2	06/Mar/2018 09:09:47	0.5	Classic
Unit 3: Unit3	06/Mar/2018 09:09:49	0.5	Classic
Unit 4: Unit4	06/Mar/2018 09:08:04	0.5	Classic
Unit 7: Unit7	06/Mar/2018 09:08:05	0.5	Classic
Unit 9: Unit9	06/Mar/2018 09:08:06	0.5	Classic
Unit 10: Unit10	06/Mar/2018 09:05:48	0.5	Classic
Unit 12: Unit12	06/Mar/2018 09:09:52	0.5	Classic
Unit 13: Unit13	06/Mar/2018 09:09:54	0.5	Classic
Unit 14: Unit14	06/Mar/2018 09:09:55	0.5	Classic
Unit 16: Unit16	06/Mar/2018 04:43:41	0.5	Classic
Unit 17: Unit17	06/Mar/2018 09:09:57	0.5	Classic
Unit 19: Unit19	06/Mar/2018 09:09:58	0.5	Classic
Unit 22: Unit22	06/Mar/2018 09:10:00	0.5	Classic
Unit 23: Unit23	06/Mar/2018 09:10:01	0.5	Classic
Unit 24: Unit24	06/Mar/2018 09:03:49	0.5	Classic
Unit 25: Unit25	06/Mar/2018 09:10:03	0.5	Classic
Unit 26: Unit26	06/Mar/2018 09:08:22	0.5	Classic
Unit 28: Unit28	06/Mar/2018 09:10:05	0.5	Classic
Unit 29: Unit29	06/Mar/2018 09:10:07	0.5	Classic
Unit 30: Unit30	06/Mar/2018 09:08:27	0.5	Classic
Unit 50: Unit50	06/Mar/2018 09:10:09	0.5	Classic
Unit 52: Unit52	06/Mar/2018 09:10:10	0.5	Classic
Unit 53: Unit53	06/Mar/2018 09:10:11	0.5	Classic
Unit 100: Unit100	06/Mar/2018 09:10:19	0.5	Classic
Unit 101: Unit101	06/Mar/2018 09:10:27	0.5	Classic
Unit 103: Unit103	06/Mar/2018 09:08:53	0.5	Classic

Figura 4.3: Configuración de la HMI para la Zona 1

Como se observa en la figura 4.3 se tiene el número de estación, número el cual se le define en la programación al PLC para que responda, la última lectura y el tiempo-poll, que define el tiempo que va a durar en esa estación antes de cambiar a la siguiente, en este caso está definido en 0.5 segundos. Para las demás zonas cuenta con la misma interfaz.

4.2 Análisis y Selección de la Solución

Se plantearon 3 posibles métodos para la realizar el proyecto, de los cuales se tuvo que seleccionar aquel que satisfaga los requerimientos.

4.2.1 Solución 1: Realizar el procesamiento de los datos en el servidor y en el PLC

Los datos son tomados desde la HMI antes de ser escritos en los puertos serie y se envía a Python, el cual se encarga de realizar el envío de los datos hacia el puerto serie(Radio) y a una dirección IP específica que tendría el PLC.

En el PLC se le indicaría que tome como prioridad los datos que recibe por el puerto de Ethernet y que en caso de no estar recibiendo información por este puerto que cambie al puerto de comunicación RS232 y adquiera los datos desde ahí.

4.2.1.1 Análisis Solución 1

Se decide utilizar Python porque es un lenguaje de alto nivel y con una gran cantidad de librerías que pueden facilitar la creación del script, que realice el envío de la información por ambos puertos, Serie y Ethernet, pero el realizar este script en el servidor implica una manipulación de este y de la HMI, además se depende del servidor y en caso de un fallo se tendría que instalar la aplicación y librerías en un servidor de respaldo. No se lograría una transparencia completa porque se depende del programa en el servidor, lo que implica que una persona con conocimiento del tema tiene que realizar la calibración cada vez que se cambie el servidor para evitar inconvenientes en el funcionamiento de la HMI y en el envío y recepción de información.

Para la parte de la estación, al igual que en el servidor no se lograría una transparencia completa, ya que al ser necesaria la programación directa en el PLC se depende del mismo, el cual debe tener las capacidades de procesamiento de esta información. Un PLC de altas prestaciones tiene un valor elevado en el mercado, un ejemplo de este tipo de PLC es el Control Logix 5580 de la marca Allen-Bradley el cual tendría un precio aproximado por estación de 1600000 colones. Se tiene que tomar en cuenta que la institución no cuenta con estos equipos tan costosos incurriendo en la necesidad de una inversión elevada

Por los motivos expuestos anteriormente, los cuales no satisfacen los requerimientos de transparencia e independencia, es que no se implementó esta solución.

4.2.2 Solución 2: Realizar el Procesamiento de los datos en un sistema externo al servidor y en el PLC

Los datos se adquieren desde los puertos serie del servidor y se envían a una Raspberry Pi 3 para que realice el procesamiento de los datos utilizando Python lo que consistiría en realizar un envío por el puerto COM hacia el radio y a una dirección IP específica que tendría el PLC.

En el PLC se le indicaría que tome como prioridad los datos que recibe por el puerto de Ethernet y que en caso de no estar recibiendo información por este puerto que cambie al puerto de comunicación RS232 y adquiera los datos desde ahí.

4.2.2.1 Análisis Solución 2

Se decide utilizar un dispositivo externo al servidor que realice el procesamiento de los datos, a sabiendas que el servidor tiene la capacidad suficiente para realizar este procesamiento, pero como el dispositivo externo adquiere los datos desde los puertos serie donde originalmente los escribe la HMI, se logra una transparencia completa para el servidor porque realiza el envío de información tal y cual lo realiza en el sistema original.

Para este dispositivo externo se decide utilizar la Raspberry Pi 3 porque es un equipo que puede ser utilizado como sistema embebido con los beneficios de un sistema operativo, además que el script en Python que se había pensado para la Solución 1 puede ser ejecutado en la misma, aprovechando los beneficios del lenguaje de alto nivel.

Para la estación al igual que en el Análisis de la solución 1 se tiene una dependencia de las prestaciones del PLC por lo que en este caso no se logra la transparencia total. Al igual que el caso anterior este tipo de PLC de altas prestaciones tiene un costo elevado por lo que para lograr el desarrollo de un prototipo se tiene que incurrir en gastos elevados.

Tomando en cuenta lo expuesto anteriormente y que no cumple en totalidad los requerimientos de transparencia e independencia para el sistema actuar es el motivo por el que no se implementó esta solución.

4.2.3 Solución 3: Realizar el procesamiento de los datos en un sistema externo al servidor y externo al PLC

Los datos son adquiridos desde los puertos serie del servidor y se envían a una Raspberry Pi 3 para que realice el procesamiento de los datos en Python, luego se escriben en los puertos serie(Radio) y en un puerto de red con una IP específica para enviarse a otra Raspberry Pi 3, que se va a encargar de procesar los datos y definir de cuál de los enlaces realiza la escritura, si de los datos que recibió por el puerto serie o los que recibió por el puerto de ethernet, una vez que toma esta decisión se escribe en el PLC.

4.2.3.1 Análisis Solución 3

Tal y como se explicó en el Análisis de la solución 2 para el servidor se logra una transparencia total debido a que la HMI realiza la escritura en los puertos serie y la Raspberry Pi 3 captura estos datos y los envía tanto por el puerto serie y de Ethernet.

Para la estación al utilizarse un dispositivo externo al PLC, se elimina toda dependencia de este, permitiendo obtener una transparencia total en esta parte del proceso, ya que no es necesario tener un PLC con altas prestaciones, debido a que la Raspberry PI 3 se encargaría de realizar el procesamiento previo y enviarle la información al PLC.

Utilizar una raspberry pi 3 reduce los gastos de manera drástica, ya que implementar este sistema en una estación tendría un costo aproximado a los 150000 generando un gran ahorro a la institución, tomando en cuenta la inversión contra el beneficio.

Tomando en cuenta lo expuesto para esta solución y que satisface los requerimientos de transparencia e independencia, es que se tomó la decisión de implementar este diseño.

4.3 Desarrollo del Diseño

La solución seleccionada en el apartado 4.2.4, consiste en agregar al sistema actual dos Raspberry Pi 3, una ubicada en el servidor y la otra ubicada en la estación con el PLC. En la figura 4.4 y 4.5 se observan los diagramas del nuevo sistema de comunicación.

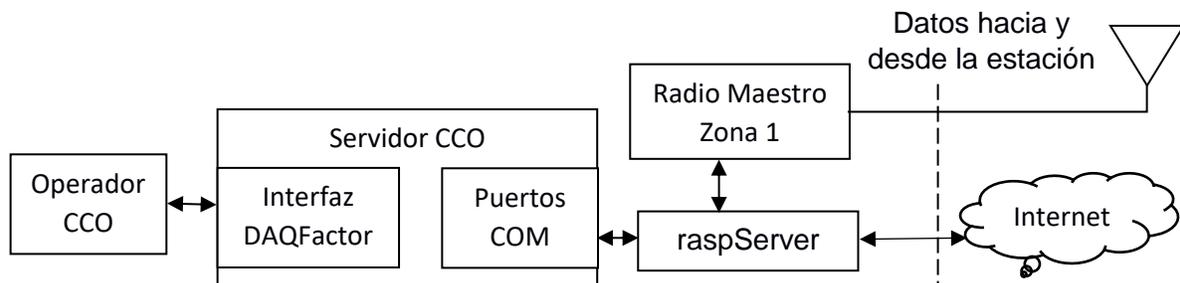


Figura 4.4: Diagrama de Bloques del nuevo sistema de transmisión y recepción CCO

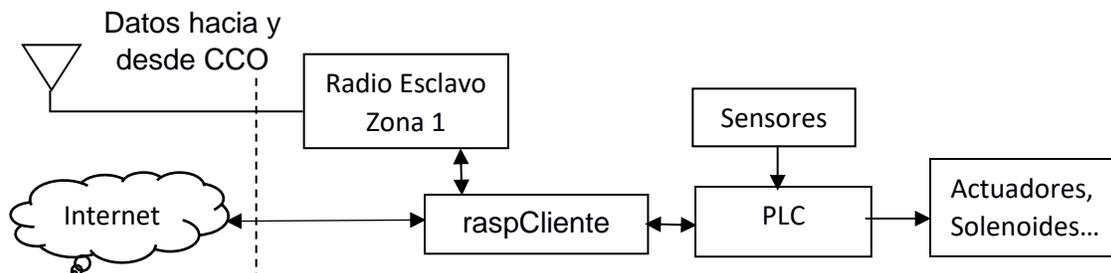


Figura 4.5: Diagrama de bloques del nuevo sistema de recepción y transmisión de datos desde la Estación

Como se aprecia en la figura 4.4 la información que se envía desde el CCO sigue la misma ruta que en la figura 4.1, hasta el punto en que llega a la primer Raspberry Pi 3, llamada raspServer que adquiere la misma y la envía por ambos canales de comunicación. Para el caso de la estación, en la figura 4.5 a diferencia de la figura 4.2 la información es recibida por la segunda Raspberry Pi 3, llamada raspCliente, la cual le envía al PLC solo la información del canal de información que llegue primero, mientras que la del otro se descarta, manteniendo la ruta de datos igual que el sistema original.

Como se puede observar se agrega un nuevo elemento en ambas figuras, el cual es la Intranet AYA. Este es el nuevo medio por el que se van a estar enviando los datos, en el caso del CCO corresponde a un punto de red del plantel y para la estación corresponde a un punto de red inalámbrico que utiliza un chip de telefonía móvil, el cual se encuentra vinculado a la intranet del AYA utilizando túnel, este dispositivo es configurado y otorgado por el departamento de TI del AYA, por lo que se define simplemente como un punto de conexión de intranet.

Tanto la raspServer como la raspCliente se les instala sistema operativo raspbian-stretch el cual es una distribución de Debian/GNU, los cuales tiene preinstalado Python 3.6.5 que es la versión que se va a utilizar para el desarrollo del script.

4.3.1 Criterios de diseño raspServer y raspCliente

Como se puede observar en la figura 4.4 y 4.5, es necesaria la lectura y escritura de los puertos serie, envío y recepción de datos por la Intranet. Es importante tener en cuenta que no es posible realizar una implementación inmediata de todas las estaciones. La misma tiene que ser paulatina lo que implica que el sistema tiene que mantenerse funcionando de la forma original para los sistemas que no se les ha realizado la nueva adaptación.

La lectura y escritura de los puertos serie es posible si se utiliza la librería PYSerial de Python [20]. La Raspberry Pi 3 tiene 4 puertos USB y una UART, pero son necesarias al menos 2 UART para realizar la comunicación de la información proveniente del servidor a la raspServer y de la raspServer hacia el radio maestro de la zona 1.

Para implementar esta UART extra se tienen 2 opciones:

- Realizar un circuito impreso utilizando el chip MAX14830 [19] el cual tiene 4 UART, pero esta opción fue descartada porque hay que importar el chip y el fabricante vende mínimo 2500 unidades.
- Utilizar un dispositivo, basado en el chip PL-2303HX, como se explicó en la sección 3.5. Esta es la opción seleccionada.

Al utilizar un adaptador es posible “convertir” el puerto USB en una UART permitiendo realizar la comunicación entre el servidor y la raspServer realizando la conexión como se muestra en la figura 3.3.

Para lograr el envío de información por internet o un puerto de red se utiliza la librería socket de Python [24]. Los datos enviados por el socket son cadenas de caracteres codificadas en bytes, que tienen que ser decodificadas para su manipulación.

La ejecución de funciones en paralelo es uno de los puntos más importantes, esto porque si se esperara a que se realicen todas las funciones de manera serial se corre el riesgo de corromper o perder datos. La librería threading de Python [23], permite que un lenguaje serie ejecute sentencias en paralelo.

4.3.2 Diseño raspServer y raspCliente

Para estos módulos se debe tomar como punto clave la transparencia con el sistema actual, por este motivo se realiza la adquisición de datos desde el servidor de CCO por el puerto serie hacia la raspServer y desde el PLC por el puerto serie hasta la raspCliente. Realizando una lectura de la información y una conversión de hexadecimal a cadena de caracteres para su posterior manipulación.

Identificar la estación es de vital importancia, ya que la raspCliente solo se va a encontrar en una estación y la raspServer manipula todas, por lo que si se envía mal la información solo una estación quedaría en funcionamiento. Este problema se previene, ya que cuando se identifica una estación que no tiene el sistema instalado, los datos capturados son escritos de manera inmediata en el radio maestro de la zona 1, pero si es una estación con el sistema instalado, los datos son procesados y escritos de manera simultánea para el radio maestro de zona 1 y el puerto de internet. La identificación de la unidad se realiza basándose en la trama del protocolo MODBUS, en la cual el primer byte corresponde a la identificación del dispositivo y los siguientes para los datos.

En el caso de la raspCliente, debido a que siempre necesita una instrucción enviada por el servidor para responder, no necesita esta lógica en el envío, pero en lo demás comparte las mismas características de la raspServer.

4.3.3 Etapa de procesamiento de los datos

Para el diseño de esta etapa es importante conocer el comportamiento de los métodos de envío de datos que se tienen. Para el caso de la escritura del puerto serie, utilizando PYSerial [20] se tiene que el método write() realiza una escritura del puerto serie en formato entero, para la escritura en el puerto de red se utiliza la librería socket [22]. Se tiene que el método send() realiza el envío en cadenas de caracteres codificadas en bytes. Para evitar errores en la lectura y escritura posterior de los datos, se realiza justo después de la adquisición de datos su conversión de bytes hexadecimales a cadena de caracteres, permitiendo que su manipulación no tenga mayor complicación.

Se diseña una nueva trama de datos para realizar el envío tanto por el puerto serie como por el puerto de red. En la figura 4.6 se puede observar la forma en la cual se encuentra compuesta la nueva trama de datos.

Cantidad Bytes 3	Contador 4	Medio 1	Datos n
---------------------	---------------	------------	------------

Figura 4.6: Trama de Envío de Datos

La cantidad de bytes se obtiene cuando se realiza la captura de los datos del servidor en el puerto serie, utilizando el comando inWaiting() [20], el valor de cantidad de bytes tiene asignados tres espacios, el cual puede representar un número de centenas. Es importante saber que esta trama es utilizada tanto para la raspServer como para la raspCliente.

El contador se obtiene cada vez que se realiza una captura de datos en el puerto serie y este cumple la función TAG, con el fin de reconocer los datos en etapas futuras. Cuenta con cuatro espacios que van desde 0 hasta 9999 esto para evitar una posible coincidencia en los datos.

Para los datos (n), recordando que los datos fueron convertidos en cadenas de caracteres, lo que implica que 2 espacios son 1 byte, la cantidad de espacios (letras) es igual al doble del valor de la cantidad de bytes.

4.3.4 Etapa de Envío

Cuando se realiza la captura de los datos del puerto serie del servidor, se analiza el primer byte para reconocer si es el dispositivo objetivo, cuando no sea el dispositivo en cuestión, los datos son escritos en el puerto serie del radio maestro. Pero cuando la identificación del dispositivo coincide, se realiza la codificación de acuerdo con la figura 4.6 y se envía al puerto serie del radio maestro, para el envío por internet, se observa la lista de clientes y se envían los datos.

4.3.5 Etapa de Recepción

Para esta etapa es necesaria la ejecución en paralelo del código, esto para evitar pérdida de información o corrupción de datos para lograr esto se utilizan threads[23].

Tanto para el puerto serie como para el puerto de red los datos llegan de acuerdo con la trama establecida en la figura 4.6, en la cual se identifica la estación según los valores de las posiciones 9 y 10 que representan el byte del número de unidad. De aquí es la importancia del medio en la trama figura 4.6, ya que este espacio asegura que no se ejecute una acción de decodificación a un dato erróneo.

4.3.6 Etapa de comparación y escritura

Luego de las etapas de recepción se tienen los datos provenientes del radio y los datos provenientes de internet, tiene que escribir solo uno de ellos. Esto se logra con una comparación utilizando el contador de la trama figura 4.6 y la identificación del dispositivo. Recordando que las recepciones trabajan en paralelo las mismas generan estas listas con los TAG los cuales se comparan y si el dato que se tienen en espera de ser escrito ya se encuentra en la otra lista es descartado, pero si no se encuentra en la otra lista es escrito de forma inmediata en el puerto serie que va hacia el servidor o hacia el PLC.

4.3.7 Etapa de comprobación de red

Este módulo brinda independencia entre la raspServer y la raspCliente, ya que si ocurre un error en la raspServer y es necesario reiniciar el programa la raspCliente sigue su ejecución sin ningún problema y viceversa.

4.3.8 Etapa de comprobación de errores

Este módulo se encuentra solo en la raspServer debido a que es el mejor punto para confirmar si los datos están retornando. Recordando que ambas recepciones se encuentran funcionando en paralelo, es necesario conocer el estado de estas, con el fin de saber de la existencia de una falla.

En el módulo de recepción existen dos contadores, el contador del puerto serie, que se encuentra en la recepción del puerto de red, que realiza incrementos de uno cada vez que se concreta un proceso de recepción y el contador del puerto

de red que se encuentra en la recepción del puerto serie, al igual que el anterior realiza incrementos de uno cada vez que se concreta un proceso de recepción.

Para determinar el error en el puerto serie y el puerto de red, se un utiliza un contador para el puerto serie que se reinicia cada vez que el puerto serie concreta una comunicación, pero si tiene una falla el puerto de red se encarga de realizar incrementos en este puerto. El contador del puerto de red al igual que el puerto serie, se reinicia cada vez que concreta una comunicación, pero si este puerto falla el puerto serie realiza incrementos en el mismo. Cuando alguno de estos contadores llega a 20, ejecuta una acción en la que envía un correo electrónico indicando el fallo de la unidad, ya sea con problema en el radio o en la red.

Se elige el valor de 20, como el indicador para realizar la acción del envío de la notificación de falla en alguno de los canales, ya que recordando el funcionamiento de la aplicación en DAQFactory la cual cada cierto tiempo realiza la solicitud de datos, este es un número lo bastante grande para evitar una falsa alarma, en caso de que la falla no sea de proporciones mayores, y a su vez pequeño para obtener una notificación lo más rápida posible en caso de una falla.

Capítulo 5: Resultados y Análisis

En este capítulo se muestran los resultados obtenidos en pruebas y en qué condiciones se considera que el sistema cumple con los requerimientos de transparencia y que esta información se encuentre libre de errores.

5.1 Comprobación del sistema con simulador

Utilizando los programas MODBUS Poll (maestro) y MODBUS Slave (esclavo) [27] los cuales permiten simular el comportamiento del protocolo MODBUS, se hacen pruebas para asegurar la integridad de los datos, en el caso de la estación que se escogió. Para realizar pruebas se utiliza la estación de tanque de Pavas que entrega al CCO reportes de nivel y apertura y cierre de la válvula.

Para esta prueba no se contó con un radio de comunicaciones en ese momento, por lo que se utiliza una conexión como la mostrada en la figura 3.3, que permite emular el comportamiento del radio.

El número de la estación Pavas es 13, el cual se representa en hexadecimal como 0D. El simulador se configura en el maestro para que solicite datos de la unidad 13 y el esclavo se configura como dicha unidad para que responda ante la solicitud. En la figura 5.1 y 5.2 se puede observar la respuesta del sistema cuando se establece una comunicación entre el maestro y el esclavo.

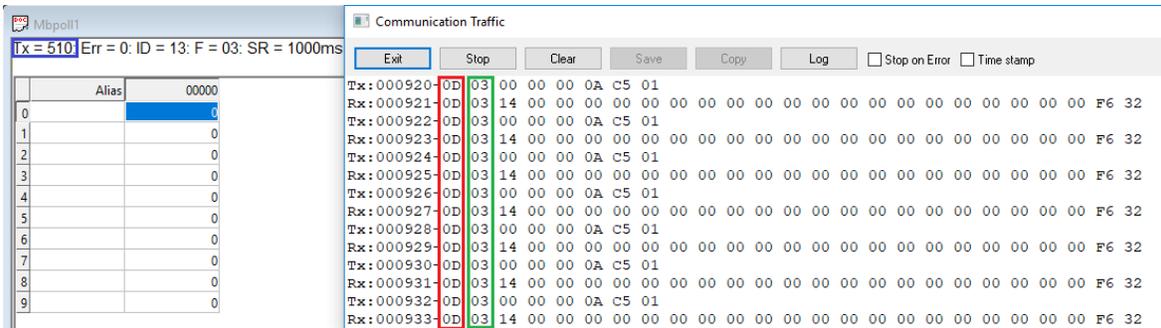


Figura 5.1: Simulación de datos enviados y recibidos por el maestro

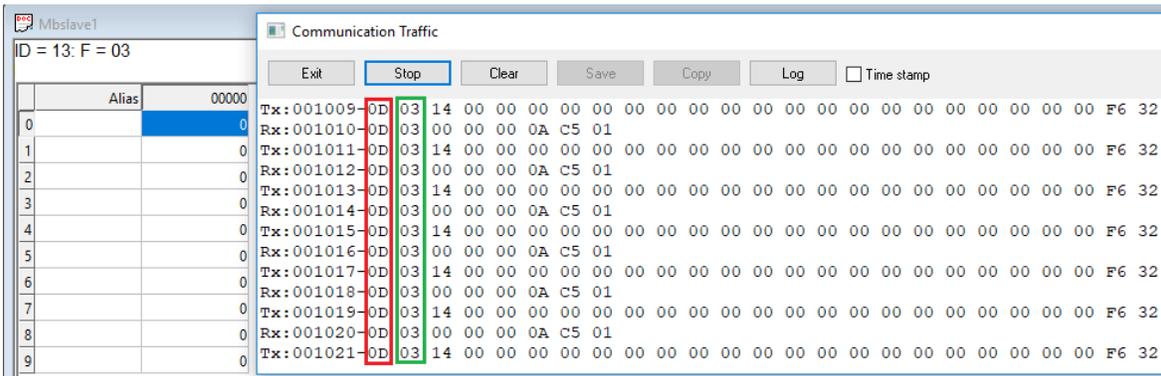


Figura 5.2: Simulación de datos recibidos y enviados por el esclavo

Tanto en la figura 5.1 como 5.2 la parte marcada en rojo corresponde al número de unidad que corresponde al 13 o 0D, le sigue el número de instrucción el cual en este caso es del tipo 3 que corresponde a los registros de entrada [5]. En la figura 5.1 se puede apreciar en la parte superior marcada en azul el número de transmisiones y errores realizados. Ya que se está utilizando una versión de prueba del software solo permite realizar 10 minutos de uso, por lo que se toma la decisión de limitar a 510 el número de transmisiones por intento.

Como se aprecia en la figura 5.1 se realizaron las 510 transmisiones con 0 errores, estos datos se obtuvieron sin emular fallos en el sistema. Para emular estos fallos se realizan desconexiones y reconexiones aleatorias de los enlaces tanto de radio como de internet. En la tabla 5.1 se muestran 10 pruebas con la emulación de falla.

Prueba	Datos	Éxito	Fallas	Porcentaje de error
1	510	510	0	0%
2	510	510	0	0%
3	510	510	0	0%
4	510	510	0	0%
5	510	510	0	0%
6	510	510	0	0%
7	510	510	0	0%
8	510	510	0	0%
9	510	510	0	0%
10	510	510	0	0%
Total	5100	5100	0	0%

Tabla 5.1: Respuesta del sistema ante fallos en alguno de los enlaces de comunicación

Según los datos de la tabla 5.1 ante desconexiones aleatorias en alguno de los canales de comunicación, el sistema mantiene la comunicación estable, lo que produce que el simulador no visualice algún error en la comunicación, verificando el correcto funcionamiento del sistema y la transparencia de este.

5.2 Comprobación de sistema

En este apartado se muestran los resultados de las distintas etapas de la implementación, que van desde la adquisición de datos hasta el envío por radio e internet.

5.2.1 Lectura y escritura de información en el servidor

En la figura 5.3 se muestra el monitor de comunicaciones del DAQFactory, el cual contiene los datos para transmitir (TX), los cuales van a ser adquiridos por la raspServer, en la figura 5.4 se muestran los datos capturados por la raspServer.

```

Rx: \x43\xC7\x91\x00\xE1\x00\x00\x00\x00\x00\x00\xBB\x24\x91\x40\x01\x04\xE1\xF8
Tx: \x80\x01\x00\x00\x00\x0A\xA2\x1C
Rx: \x80\x01\x02\x00\x00\x85\xE2
Tx: \x80\x02\x00\x00\x00\x0C\x66\x1E
Rx: \x80\x02\x02\x09\x00\x83\xF6
Tx: \x80\x03\x00\x00\x00\x18\x5B\xD1
Rx: \x80\x03\x30\x47\xAE\x39\xFE\x7F\xF8\x00\x00\x5C\x00\x44\xE5\x60\x00\x42\x72\
Tx: \x43\xCC\x85\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x86\x27
Rx: \x43\xCC\xA5\xE0\x10\x00\x00\x00\x00\x00\x00\xB7\x25\x95\x40\x01\x04\x63\x94
Tx: \x86\x01\x00\x00\x00\x0A\xA2\x7A
Rx: \x86\x01\x02\x00\x00\x0D\xE2
Tx: \x86\x02\x00\x00\x00\x0C\x66\x78
Rx: \x86\x02\x02\x06\x00\x0E\x06
Tx: \x86\x03\x00\x00\x00\x18\x5B\xB7
Rx: \x86\x03\x30\x00\x14\x24\x10\x00\x10\x00\x0B\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
Tx: \x43\xD3\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x5D\xAB
Rx: \x43\xD3\x21\x00\x1C\x00\x00\x00\x00\x00\x00\x00\x00\xFA\x40\x00\x04\x9C\xAD
Tx: \x43\xD6\x05\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x9D\x6C
Rx: \x43\xD6\x25\x00\x72\x00\x00\x00\x00\x00\x00\x27\x00\x00\x05\xCF\x45
Tx: \x43\xD8\x06\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x10\x6B
Rx: \x43\xD8\x26\x00\x0F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x15\x3A
Tx: \x43\xDA\x08\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x9D\xA4
Rx: \x43\xDA\x28\xF0\xF5\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x78\xC1
Tx: \x43\xDB\x0B\xFC\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x0E\x9A
Tx: \x43\xDD\x0D\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xDF\xA3
Tx: \x43\xDE\x0F\xFF\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xCB\x9E
Rx: \x43\xDD\x2D\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x26\x00\x00\x05\x34\xCE
Tx: \x43\xE0\x10\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x3F\xAF
Rx: \x43\xDE\x2F\x00\x48\xA4\x00\x00\xEC\x00\x00\x00\x06\x40\x80\x05\xE6\x16\x43\
Tx: \x43\xE2\x12\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xBE\x6C
Rx: \x43\xE2\x32\x00\xEA\x00\x00\x00\x00\x00\x00\x00\x00\x00\x1F\x00\x00\x06\x83\x7A
Tx: \x43\xE4\x14\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x3E\x68
Rx: \x43\xE4\x34\x00\x00\x3B\x95\xFD\x00\x00\x00\x00\x21\x17\x00\x06\xB4\xA9
Tx: \x43\xE7\x15\xFE\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x2B\xD7
Rx: \x43\xE7\x35\x00\x00\x8A\x00\x00\x00\x00\x00\x00\x00\x00\x27\x10\x00\x05\x1E\xE6
Tx: \x43\xE8\x66\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x40\x1F
Rx: \x43\xE8\x86\xFF\xD7\x00\x00\x00\x00\x00\x00\x00\x00\x04\x40\x00\x05\x03\x95
Tx: \x69\x01\x00\x00\x00\x0A\xB4\xE5
Rx: \x69\x01\x02\x00\x00\xD8\x35
Tx: \x69\x02\x00\x00\x00\x0C\x70\xE7
Rx: \x69\x02\x02\x00\x00\xD8\x71
Tx: \x69\x03\x00\x00\x00\x18\x4D\x28
Rx: \x69\x03\x30\x00\x00\x2D\x14\x00\x13\x00\xED\x30\x30\x30\x30\x30\x30\x30\x42\
Tx: \x43\xE9\x71\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x96\x08
Rx: \x43\xE9\x91\x00\xE1\x00\x00\x00\x00\x00\x00\xBB\x24\x91\x40\x01\x04\x4F\xE4
Tx: \x80\x01\x00\x00\x00\x0A\xA2\x1C
Rx: \x80\x01\x02\x00\x00\x85\xE2
Tx: \x80\x02\x00\x00\x00\x0C\x66\x1E
Rx: \x80\x02\x02\x09\x00\x83\xF6
Tx: \x80\x03\x00\x00\x00\x18\x5B\xD1
Rx: \x80\x03\x30\x47\xAE\x3A\x12\x7F\xF8\x00\x00\x5C\x00\x44\xE5\x60\x00\x42\x72\
Tx: \x43\xEB\x85\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xE1\x3D
Rx: \x43\xEB\xA5\xE0\x10\x00\x00\x00\x00\x00\x00\xB7\x25\x93\x40\x01\x04\x04\x06
Tx: \x86\x01\x00\x00\x00\x0A\xA2\x7A
Rx: \x86\x01\x02\x00\x00\x0D\xE2
Tx: \x86\x02\x00\x00\x00\x0C\x66\x78
Rx: \x86\x02\x02\x06\x00\x0E\x06
Tx: \x86\x03\x00\x00\x00\x18\x5B\xB7
Rx: \x86\x03\x30\x00\x14\x24\x09\x00\x10\x00\x0B\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\

```

Figura 5.3: Monitor de comunicaciones DAQFactory estaciones


```

Tx: \x43\x01\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x57\x04
Rx: \x43\x01\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xBE\x68
Tx: \x43\x02\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x14\x05
Rx: \x43\x02\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xFD\x69
Tx: \x43\x05\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x53\x07
Rx: \x43\x05\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xBB\x97
Tx: \x43\x06\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x10\x06
Rx: \x43\x06\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xF9\x6A
Tx: \x43\x07\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xD1\x06
Rx: \x43\x07\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x39\x96
Tx: \x43\x08\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x9E\x02
Rx: \x43\x08\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x77\x6E
Tx: \x43\x09\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x5F\x02
Rx: \x43\x09\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xB6\x6E
Tx: \x43\x0A\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x1C\x03
Rx: \x43\x0A\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xF4\x93
Tx: \x43\x0B\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xDD\x03
Rx: \x43\x0B\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x35\x93
Tx: \x43\x0C\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x9A\x01
Rx: \x43\x0C\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x72\x91
Tx: \x43\x0D\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x5B\x01
Rx: \x43\x0D\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xB2\x29
Tx: \x43\x0E\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\x18\x00
Rx: \x43\x0E\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\xF1\x6C
Tx: \x43\x0F\x0D\xff\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x08\xD9\x00
Rx: \x43\x0F\x2D\x7F\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x04\x30\x6C

```

Figura 5.6: Monitor de comunicaciones DAQFactory con prioridad en la estación Pavas

Los datos de transmisión de la figura 5.6 tienen que prepararse con la trama de la figura 4.6, esto para poder realizar el envío de la información por ambos canales de comunicación. En la figura 5.7 se observa como quedan los datos previos a su envío.

0180935E43020dff0000000000000000000000085704	0180935R43020dff0000000000000000000000085704
0180936E43020dff0000000000000000000000081405	0180936R43020dff0000000000000000000000081405
0180937E43050dff0000000000000000000000085307	0180937R43050dff0000000000000000000000085307
0180938E43060dff0000000000000000000000081006	0180938R43060dff0000000000000000000000081006
0180939E43070dff000000000000000000000008d106	0180939R43070dff000000000000000000000008d106
0180940E43080dff0000000000000000000000089e02	0180940R43080dff0000000000000000000000089e02
0180941E43090dff0000000000000000000000085f02	0180941R43090dff0000000000000000000000085f02
0180942E430a0dff0000000000000000000000081c03	0180942R430a0dff0000000000000000000000081c03
0180943E430b0dff000000000000000000000008dd03	0180943R430b0dff000000000000000000000008dd03
0180944E430c0dff0000000000000000000000089a01	0180944R430c0dff0000000000000000000000089a01
0180945E430d0dff0000000000000000000000085b01	0180945R430d0dff0000000000000000000000085b01
0180946E430e0dff0000000000000000000000081800	0180946R430e0dff0000000000000000000000081800
0180947E430f0dff000000000000000000000008d900	0180947R430f0dff000000000000000000000008d900
0180948E43100dff0000000000000000000000088608	0180948R43100dff0000000000000000000000088608
0180949E43110dff0000000000000000000000084708	0180949R43110dff0000000000000000000000084708
0180950E43120dff0000000000000000000000080409	0180950R43120dff0000000000000000000000080409
0180951E43150dff000000000000000000000008430b	0180951R43150dff000000000000000000000008430b
0180952E43160dff000000000000000000000008000a	0180952R43160dff000000000000000000000008000a
0180953E43170dff000000000000000000000008c10a	0180953R43170dff000000000000000000000008c10a

Figura 5.7: Datos codificados para envío por radio e internet

Como se aprecia en la figura 5.7 los datos se envían de forma simultánea y de igual forma son recibidos en el puerto serie y de red de la raspCliente y esta decide cual de los dos utilizar y lo escribe en el PLC. Cuando el PLC responde la información se realiza la codificación y el envío de los datos de forma similar a la figura 5.7 y la raspServer los recibe como se muestra en la figura 5.8 y escribe solo uno de ellos.

```

0181358E43012d7f0000000000000000e6000004be68 0181358R43012d7f0000000000000000e6000004be68
0181359E43022d7f0000000000000000e6000004fd69 0181359R43022d7f0000000000000000e6000004fd69
0181360E43052d7f0000000000000000e7000004bb97 0181360R43052d7f0000000000000000e7000004bb97
0181361E43062d7f0000000000000000e6000004f96a 0181361R43062d7f0000000000000000e6000004f96a
0181362E43072d7f0000000000000000e70000043996 0181362R43072d7f0000000000000000e70000043996
0181363E43082d7f0000000000000000e6000004776e 0181363R43082d7f0000000000000000e6000004776e
0181364E43092d7f0000000000000000e6000004b66e 0181364R43092d7f0000000000000000e6000004b66e
0181365E430a2d7f0000000000000000e7000004f493 0181365R430a2d7f0000000000000000e7000004f493
0181366E430b2d7f0000000000000000e70000043593 0181366R430b2d7f0000000000000000e70000043593
0181367E430c2d7f0000000000000000e70000047291 0181367R430c2d7f0000000000000000e70000047291
0181368E430d2d7f0000000000000000e5000004b229 0181368R430d2d7f0000000000000000e5000004b229
0181369E430e2d7f0000000000000000e6000004f16c 0181369R430e2d7f0000000000000000e6000004f16c
0181370E430f2d7f0000000000000000e6000004306c 0181370R430f2d7f0000000000000000e6000004306c
0181371E43102d7f0000000000000000e60000046f64 0181371R43102d7f0000000000000000e60000046f64
0181372E43112d7f0000000000000000e8000004ac8c 0181372R43112d7f0000000000000000e8000004ac8c
0181373E43122d7f0000000000000000e6000004ed65 0181373R43122d7f0000000000000000e6000004ed65
0181374E43152d7f0000000000000000e6000004aa67 0181374R43152d7f0000000000000000e6000004aa67
0181375E43162d7f0000000000000000e6000004e966 0181375R43162d7f0000000000000000e6000004e966
0181376E43172d7f0000000000000000e7000004299a 0181376R43172d7f0000000000000000e7000004299a

```

Figura 5.8: Datos enviados desde la raspCliente

La comprobación de una falla se realiza en esta etapa, ya que cuando los datos ingresan solo por uno de los canales y se llega al valor límite establecido, se ejecuta la acción que envía un correo electrónico con el texto:

- “Fallo en la comunicación por Radio Tanque Pavas”
- “Fallo en la conexión de internet Tanque Pavas”

En el Apéndice 3 se pueden observar las imágenes del correo electrónico que envía el sistema

En el Apéndice 4 se observan imágenes del DAQFactory operando con el sistema.

Capítulo 6: Conclusiones

Se agregó un enlace redundante vía internet, el cual satisface los requerimientos de un sistema transparente para el sistema original, utilizando un equipo externo y de fácil instalación.

El sistema realiza una recepción simultánea de los datos, en la que el TAG se utiliza para saber si los datos ya fueron recibidos para evitar volver a escribir los mismos en el puerto serie generando corrupción de datos.

Como se conoce por cuál de los canales se están recibiendo los datos, es posible saber cuándo uno de estos deja de funcionar, permitiendo así enviar una notificación por correo electrónico el cual no altera el funcionamiento original del sistema, manteniendo la transparencia de este.

El sistema tiene un impacto económico favorable para la institución, en la que se logra un ahorro aproximado por estación de 1420000 colones, lo que permite utilizar estos recursos en otras actividades como realizar nuevos proyectos, más capacitaciones para el personal, mejoramiento de equipos.

Capítulo 7: Recomendaciones

Utilizar un enlace de internet inalámbrico para las estaciones con mayor estabilidad, ya que dependiendo del momento la red de internet es mucho más lenta que el enlace de radio.

Cerciorarse que la temperatura de las Raspberry Pi 3 se mantenga inferior a los 80° C ya que después de este punto la misma corre el riesgo de sufrir daños, además que realiza comportamientos inesperados.

Utilizar un hub de USB con alimentación externa cuando se quiera implementar todas las estaciones, ya que son necesarios 6 puertos USB y la Raspberry Pi 3 solo tiene 4.

Referencias

- [1] Martín Gómez, P. (s.f.). USB. Recuperado 8 abril, 2018, de <http://www.sase.com.ar/2013/files/2013/09/SASE2013-USB-P-Gomez.pdf>
- [2] López Pérez, E. (s.f.). INGENIERIA EN MICROCONTROLADORES Protocolo USB. Recuperado 28 marzo, 2018, de <http://www.i-micro.com/pdf/articulos/usb.pdf>
- [3] National Instruments. (2006, 6 junio). Comunicación Serial: Conceptos Generales. Recuperado 8 abril, 2018, de <http://digital.ni.com/public.nsf/allkb/039001258CEF8FB686256E0F005888D1>
- [4] Tripot. (s.f.). RS 232. Recuperado 8 abril, 2018, de <http://rddatos.tripod.com/rs232.htm>
- [5] National Instruments. (2014, 16 octubre). Información Detallada sobre el Protocolo Modbus. Recuperado 9 abril, 2018, de <http://www.ni.com/white-paper/52134/es/>
- [6] Marín, F. (2014, 16 octubre). Información Detallada sobre el Protocolo Modbus. Recuperado 9 abril, 2018, de http://www.el.uma.es/marin/Practica4_UART.pdf
- [7] Prolific. (2013, 20 marzo). PL-2303HX Edition (Chip Rev D) USB to Serial Bridge Controller. Recuperado 14 abril, 2018, de http://www.prolific.com.tw/UserFiles/files/ds_pl2303HXD_v1_4_4.pdf
- [8] Vargas L, M. (2016, 1 marzo). Prepárese desde ahora para el apagón de la TV analógica. Recuperado 14 octubre, 2017, de <https://www.nacion.com/tecnologia/innovaciones/preparesse-desde-ahora-para-el-apagon-de-la-tv-analogica/Y4GQPNX7WJA6PLSNGUF6TDP7BI/story/>
- [9] Méndez Montero, A. (2017, 1 septiembre). Cuestionan preparación de Costa Rica para encendido de la televisión digital. Recuperado 14 febrero, 2018, de <https://www.ucr.ac.cr/noticias/2017/09/01/cuestionan-preparacion-de-costa-rica-para-encendido-de-la-television-digital.html>
- [10] Automación Micromecánica s.a.i.c. (s.f.). CONTROLADOR LÓGICO PROGRAMABLE. Recuperado 18 mayo, 2018, de <http://www.microautomacion.com/capacitacion/Manual061ControladorLgicoProgramablePLC.pdf>
- [11] RASPBERRY PI FOUNDATION. (s.f.). WHAT IS A RASPBERRY PI? Recuperado 20 ma, 2018, de <https://www.raspberrypi.org/help/faqs/#introWhatIs>
- [12] PCmag. (s.f.). Definition of HMI. Recuperado 16 mayo, 2018, de <https://www.pcmag.com/encyclopedia/term/44300/hmi>
- [13] Wonderware. (s.f.). Interfaz Hombre Máquina. Recuperado 16 mayo, 2018, de <https://www.wonderware.com/es-es/hmi-scada/what-is-hmi/>
- [14] AUTRACEN SA. (s.f.). Sistemas SCADA. Recuperado 16 mayo, 2018, de <http://www.autracen.com/sistemas-scada/>
- [15] Castro Lozano, C., & Romero Morales, C. (s.f.). Introducción a SCADA. Recuperado 16 mayo, 2018, de <http://www.uco.es/grupos/eatco/automatica/ihm/descargar/scada.pdf>

- [16] Salas Arriarán, S. (2018, 6 febrero). Todo sobre sistemas embebidos [Capítulo 1]. Recuperado 16 mayo, 2018, de <https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/579143/Todo+sobre+sistemas+embebidos++1er+cap.pdf;jsessionid=CAA666AC1D7F1C802162F31FFA7C2966?sequence=1>
- [17] Herrera Quintero, L. (2015, 6 abril). Telemetría y telegestión en procesos industriales mediante canales inalámbricos Wi Fi utilizando instrumentación virtual y dispositivos PDA (Personal Digital Assitant). Recuperado 16 mayo, 2018, de https://www.researchgate.net/publication/268412727_Telemetria_y_telegestion_en_procesos_industriales_mediante_canales_inalambricos_Wi_Fi_utilizando_instrumentacion_virtual_y_dispositivos_PDA_Personal_Digital_Assitant
- [18] SISTEMA DE TELEMETRÍA BASADO EN RADIO FRECUENCIA. (s.f.). Recuperado 16 mayo, 2018, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/158/A7.pdf?sequence=7>
- [19] Maxim Integrated. (s.f.). MAX14830. Recuperado 23 marzo, 2018, de <https://www.maximintegrated.com/en/products/interface/controllers-expanders/MAX14830.html>
- [20] Liechti, C. (2017, 23 septiembre). pySerial Documentation. Recuperado 14 abril, 2018, de <https://media.readthedocs.org/pdf/pyserial/latest/pyserial.pdf>
- [21] Broadcom Corporation. (2012). BCM2835 ARM Peripherals. Recuperado 22 mayo, 2018, de <https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>
- [22] Python Software Foundation. (s.f.). Socket. Recuperado 15 abril, 2018, de <https://docs.python.org/3/library/socket.html>
- [23] Python Software Foundation. (s.f.). threading — Thread-based parallelism. Recuperado 15 abril, 2018, de <https://docs.python.org/3/library/threading.html>
- [24] Modbus tools. (s.f.). Modbus Poll. Recuperado 10 abril, 2018, de http://www.modbustools.com/modbus_poll.html
- [25] Modbus tools. (s.f.). Modbus Poll. Recuperado 10 abril, 2018, de http://www.modbustools.com/modbus_slave.html

Apéndice 1

Tabla Completa de Frecuencias Estaciones AYA

Nombre del Sitio	RF-1.2-TX	RF-1.1-RX
Bello Horizonte	429,08750 MHz	424,08750 MHz
Entierrillos	429,08750 MHz	424,08750 MHz
Naranjal	429,08750 MHz	424,08750 MHz
Cruz Roja Escazú	429,08750 MHz	424,08750 MHz
Tanque Psiquiátrico	429,08750 MHz	424,08750 MHz
Tanq-Bomb El Carmen	429,08750 MHz	424,08750 MHz
Tanq-Bomb Corazón de Jesús	429,08750 MHz	424,08750 MHz
Tanque San Antonio Escazú	429,08750 MHz	424,08750 MHz
Planta Alajuelita	429,08750 MHz	424,08750 MHz
Calle Nueva	429,08750 MHz	424,08750 MHz
Bello Horizonte	429,08750 MHz	424,08750 MHz
Pavas	429,08750 MHz	424,08750 MHz
Tanques del Sur	429,08750 MHz	424,08750 MHz
Maiquetía	429,08750 MHz	424,08750 MHz
San Juan de Dios	429,08750 MHz	424,08750 MHz
Tanque La Pelota	429,08750 MHz	424,08750 MHz
Bombeo La Pelota	429,08750 MHz	424,08750 MHz
Tanque Higuito	429,08750 MHz	424,08750 MHz
Tanque Los Guidos	429,08750 MHz	424,08750 MHz
Tanque Curio	429,08750 MHz	424,08750 MHz
Tanque Lajas	429,08750 MHz	424,08750 MHz
Tanque Villacon	429,08750 MHz	424,08750 MHz
Tanque Pozos	429,08750 MHz	424,08750 MHz
Tanque Cerro Minas	429,08750 MHz	424,08750 MHz
Repetidor Sacramento-Uruca	429,08750 MHz	424,08750 MHz
Repetidor Sacramento-Escazú	424,62500 MHz	429,62500 MHz
Matra	429,62500 MHz	424,62500 MHz
Pozo Belén 01	429,62500 MHz	424,62500 MHz
Zoológico	429,62500 MHz	424,62500 MHz
Ciudad Colon	429,62500 MHz	424,62500 MHz
Tanque Salitral	429,62500 MHz	424,62500 MHz
Tanque Santa Ana	429,62500 MHz	424,62500 MHz
Tanque Piedades	429,62500 MHz	424,62500 MHz
Potrerrillos	429,62500 MHz	424,62500 MHz
CNP 1-2	429,62500 MHz	424,62500 MHz
CNP 3-4	429,62500 MHz	424,62500 MHz
CNP 5-6	429,62500 MHz	424,62500 MHz
CNP 7-8	429,62500 MHz	424,62500 MHz
CCO AYA	424,08750 MHz	429,08750 MHz
CCO AYA	424,23750 MHz	429,23750 MHz
CCO AYA	424,45000 MHz	429,45000 MHz
Planta Sitios	429,45000 MHz	424,45000 MHz
Tanque Sitios	429,45000 MHz	424,45000 MHz

La Finca	429,45000 MHz	424,45000 MHz
Bombeo Ipís	429,45000 MHz	424,45000 MHz
Planta Los Cuadros	429,45000 MHz	424,45000 MHz
Planta Guadalupe	429,45000 MHz	424,45000 MHz
Tanque Mata Plátano	429,45000 MHz	424,45000 MHz
Tanque Tibás	429,45000 MHz	424,45000 MHz
Tanque Sabanilla	429,45000 MHz	424,45000 MHz
Tanque Pizote	429,45000 MHz	424,45000 MHz
Tanque Concepción	429,45000 MHz	424,45000 MHz
Tanque Granadilla	429,45000 MHz	424,45000 MHz
Tanque Guayabos	429,45000 MHz	424,45000 MHz
Tanque Cipreses	429,45000 MHz	424,45000 MHz
Tanque Montufar	429,45000 MHz	424,45000 MHz
Tanque Curridabat	429,45000 MHz	424,45000 MHz
Tanque San Blas	429,45000 MHz	424,45000 MHz
Tanque Coronado	429,45000 MHz	424,45000 MHz
Tanque Platanares	429,45000 MHz	424,45000 MHz
Tanque Las Cruces	429,45000 MHz	424,45000 MHz
Planta San Jerónimo	429,45000 MHz	424,45000 MHz
Planta Baja Repetidor	429,45000 MHz	424,45000 MHz
Planta Alta	429,45000 MHz	424,45000 MHz
Coris	429,45000 MHz	424,45000 MHz
Planta Guatuso	429,45000 MHz	424,45000 MHz
Pozo Registro	429,45000 MHz	424,45000 MHz
Pozo Periféricas	429,45000 MHz	424,45000 MHz
Pozo W1	429,2375 MHz	424,23750 MHz
Pozo W2	429,2375 MHz	424,23750 MHz
Pozo W3	429,2375 MHz	424,23750 MHz
Pozo W4	429,2375 MHz	424,23750 MHz
Pozo W5	429,2375 MHz	424,23750 MHz
Pozo W6	429,2375 MHz	424,23750 MHz
Pozo W7	429,2375 MHz	424,23750 MHz
Pozo W8	429,2375 MHz	424,23750 MHz
Pozo W9	429,2375 MHz	424,23750 MHz
Pozo W10	429,2375 MHz	424,23750 MHz
Pozo W12	429,2375 MHz	424,23750 MHz
Pozo W13	429,2375 MHz	424,23750 MHz
Pozo W14	429,2375 MHz	424,23750 MHz
Pozo W15	429,2375 MHz	424,23750 MHz
La Libertad	429,2375 MHz	424,23750 MHz
Tanque la Meseta	429,2375 MHz	424,23750 MHz
Tanque de San Pablo	429,2375 MHz	424,23750 MHz
Pozo Ricon Ricardo #2	429,2375 MHz	424,23750 MHz
Tanque Uruca	429,2375 MHz	424,23750 MHz
Pozo San Pablo	429,2375 MHz	424,23750 MHz

Tabla A1.1: Frecuencias de Estaciones AYA

Apéndice 2

Tipos de PLC disponibles en el AYA



Figura A2.1: PLC RFSkada 4



Figura A2.2: PLC RFSkada 16



Figura A2.3: PLC RFScada 32

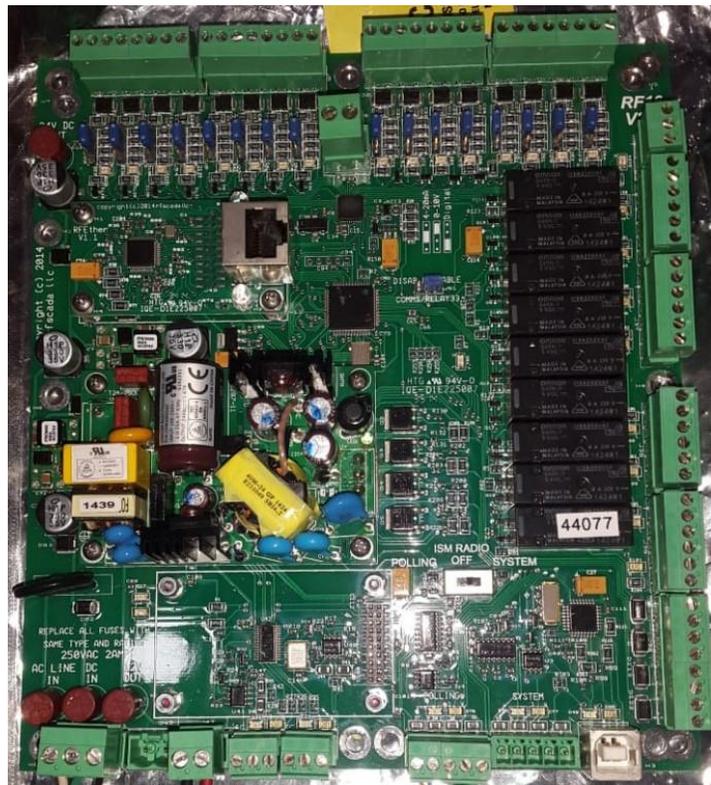


Figura A2.4: PLC RFScada 16V3.1

Apéndice 3

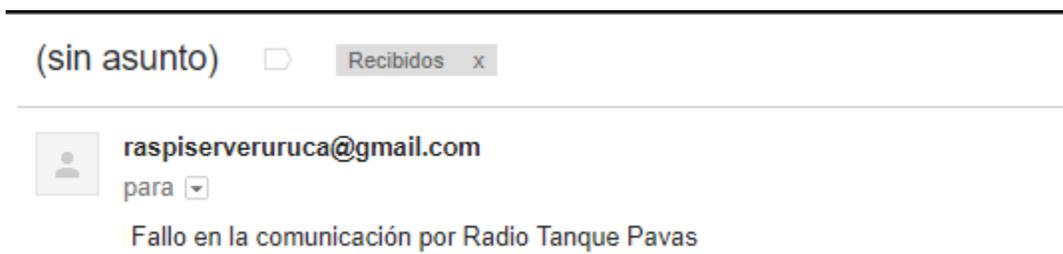


Figura A3.1: Notificación de error en Radio

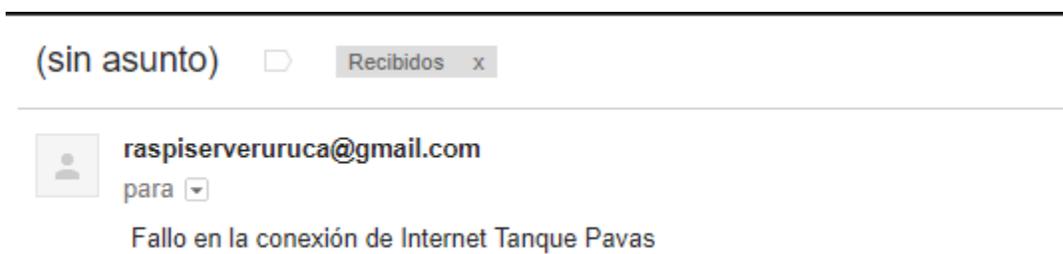


Figura A3.2: Notificación de error en Internet

Apéndice 4

En las siguientes imágenes se muestra el comportamiento del DAQFactory junto con el sistema. El sistema se instaló 23 de mayo del 2018 y estas imágenes se capturar prácticamente una semana después.

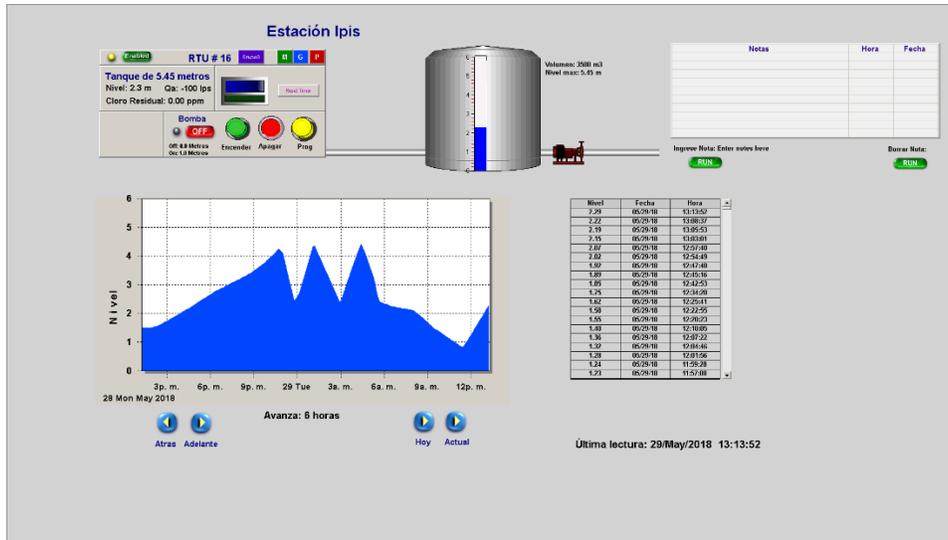


Figura A4.1: Estación Ipís

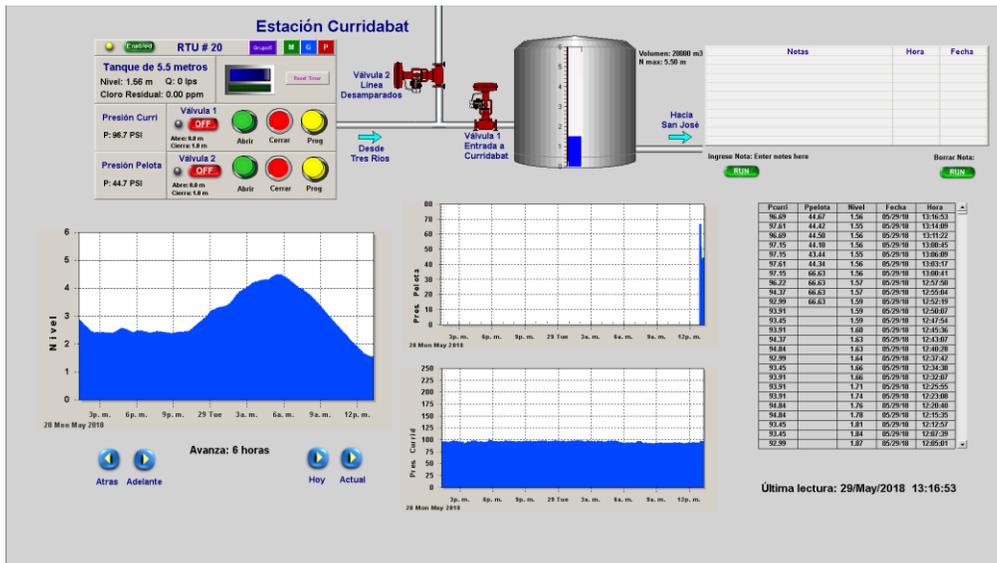


Figura A4.2: Estación Curridabat

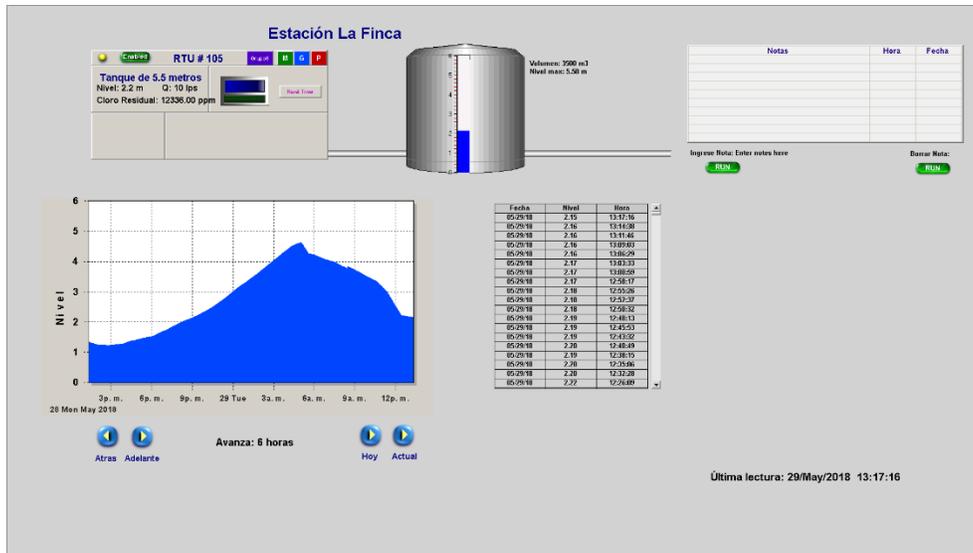


Figura A4.3: Estación La Finca

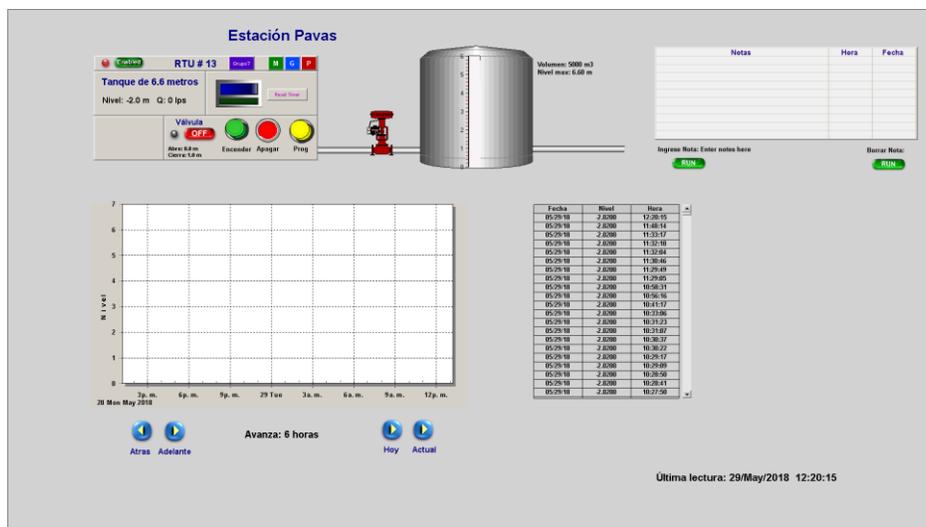


Figura A4.4: Estación Pavas

La estación Pavas presentaba problemas con el transductor que da el nivel del tanque.