Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Desarrollo de un sistema de comunicación y almacenaje de los datos de masas de aceite medidos por los Micro Motions Transmitter en la refinería de aceite de INOLASA

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Luis Enrique Alpízar Ramírez

Cartago, noviembre de 2016

INSTITUTO TECNOLÓGICO DE COSTA RICA

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 y almacenaje de los datos de masas de aceite medidos por los Micro Motions Transmitter en la refinería de aceite de INOLASA, realizado por Luis Enrique Alpízar Ramírez 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. William Marín Moreno

Profesor lector

Ing. Leonardo Rivas Arce

Profesor lector

Ing. José Faustino Montes de Oca Murillo

Profesor asesor

Cartago, 21 de noviembre de 2016

Declaración de autenticidad

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 conocimientos propios.

En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Luis Enrique Alpízar Ramírez

Cédula: 604070805

Resumen

La automatización implica procesos que han venido revolucionando la industria; suplanta tareas realizadas por humanos por mecanismos que efectúen el mismo trabajo, con el uso de métodos más eficientes en la elaboración de sus productos.

La empresa Industrial Oleaginosas Americanas S.A, conocida como INOLASA, se basa en la producción y comercialización de aceites vegetales, lecitina de soya e insumos alimenticios para la nutrición animal. Actualmente, el departamento de refinería de aceite de la industria INOLASA, cuenta con una amplia variedad de procesos y equipos para su producción. Gran cantidad de estos procesos se controlan manualmente. Uno de los casos, es la monitorización de la producción de aceite que fluye por sus canales, donde, su lectura es visible únicamente en la planta de refinería que se encuentra ubicada aproximadamente 80 metros de las oficinas de producción, lugar donde el dato debe ser estudiado.

En el presente proyecto se plantea desarrollar un sistema capaz de tomar la lectura de aceites y de productos derivados, de distintos equipos utilizados en la planta y que finalmente puedan ser visualizados en un ordenador.

Para ello, se va a desarrollar la programación de un microcontrolador que permita realizar lecturas de las mediciones de los equipos de la planta y la comunicación con otros sistemas mediante la red de la empresa. Además, se va a realizar la programación para una aplicación capaz de tomar los datos de la red local y almacenarlos en una base de datos.

Palabras Claves: Comunicación serial, protocolo RS 485, Modbus RTU, protocolo de señal de flujo de corriente 4 a 20 mA, protocolo de comunicación Hart, comunicación SPI, sistema de base de datos, red local.

Abstract

Automation are processes that have been revolutionized industry, supplanting tasks performed by humans, by mechanisms that made the same work, providing more efficient methods in developing their products.

The company, Industrial Oleaginosas Americanas S.A., known as INOLASA, is a company based on the production and marketing of vegetable oils, soy lecithin and food supplies for animal nutrition. Nowadays, the department of oil refinery, of INOLASA industry has a lot of variety of processes and equipment for production. Many processes are controlled manually. One case, is the monitoring the oil production, flowing through ducts where, the reading is visible only in the oil refinery plant, which is located approximately 80 meters from the production offices, where the data is studied.

The present project aims to develop a system able to take oils measurements and derivative products, of different equipment used in the plant and finally to be viewed on a computer.

For this, it will develop a programming for a microcontroller, which be able to made readings in different equipments of measurement in the plant and communicate with other systems via the company network. Also, it will make a programming for an application, which be able to take data from the local network and store them in a database.

Keywords: Serial Communication, RS485 protocol, Modbus RTU, Signal 4 to 20 mA, HART Protocol, SPI communication, Database System, Local Area Network.

Dedicatoria

Dedico este trabajo a mi madre, María Elena Ramírez González; a mi padre José Enrique Alpízar Saborío y a mi hermana María Auxiliadora Alpízar Ramírez, por toda la ayuda brindada, inspiración y apoyo a lo largo de toda mi formación académica.

Agradecimientos

Primeramente, a mis padres, porque sin ellos no hubiera sido posible llegar al final de mi licenciatura. Al personal de INOLASA, por permitirme realizar el proyecto de graduación en esta empresa.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1: Introducción

La automatización en las industrias implica procesos que han venido renovando e innovando los distintos sistemas de producción, mediante métodos más eficientes en sus numerosos pasos para la elaboración de sus productos. Para entender mejor lo que se busca realizar, se definirá *automatización* como, "sistema de fabricación diseñado con el fin de usar la capacidad de las máquinas para llevar a cabo determinadas tareas, anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana. El término automatización también se ha utilizado para describir sistemas no destinados a la fabricación en los que los dispositivos programados o automáticos pueden funcionar de forma independiente o semi-independiente del control humano". [11]

Esta gran revolución en las industrias ha venido remplazando los antiguos procesos, debido a la facilidad que brinda a las empresas para tener buenos resultados, además de una excelente precisión en estos.

INOLASA está ubicada en la ciudad de El Roble de la Provincia de Puntarenas, 600 metros al este del Colegio Técnico de Puntarenas, sobre la carretera principal. Cuenta con numerosos procesos industriales manuales, los cuales pretenden remplazar por procesos automatizados, y a su vez, tener sistemas capaces de monitorizar sus procesos. [8]

INOLASA es una empresa basada en la producción y comercialización de aceites vegetales, lecitina de soya e insumos alimenticios para la nutrición animal. [8]

Unos de los principios fundamentales que rigen el funcionamiento de la empresa es la utilización de tecnología de punta; con ello, INOLASA se asegura de ofrecer productos de excelente calidad que satisfagan las necesidades y gustos de sus clientes. La constante inversión en nuevos equipos, la capacitación de sus colaboradores, el uso de las mejores materias primas y los estrictos controles de calidad, distinguen a las plantas de producción de INOLASA. [8]

Cuenta con equipos de alta tecnología que le permiten tener la capacidad de procesar 1200 Toneladas Métricas (TM) de frijol de soya al día. Entre los principales productos está el aceite de soya para consumo doméstico, para consumo industrial e institucional; además, produce lecitina de soya y harina de soya para consumo animal. [8]

INOLASA ha sido reconocida por la Asociación Americana de Soya como una de las plantas procesadoras de frijol de soya más eficientes y modernas en toda América Latina. Su Departamento de Control de Calidad recibió desde el año 2001 el "status" de *Químico Aprobado*, el cual es extendido por la A.O.C.S (American Oil Chemists Society). ello garantiza la certificación a nivel internacional de sus laboratorios de control de calidad; también cuentan con la certificación FSSC22000. [8]

El presente proyecto se realiza en la planta de refinería de aceite de la empresa INOLASA, debido a la necesidad de elaborar un sistema automático capaz de tomar datos de distintos equipos, que puedan ser procesados y almacenados para posteriores estudios de producción.

1.1 Meta

La meta principal por cumplir en el proyecto es que, al sustituir el método de la lectura manual de las masas de aceite en su proceso de refinación, se debe brindar una mejora a la empresa INOLASA, de modo que constituya un proceso más eficiente y cómodo para trabajar.

También se tiene como meta modernizar los métodos de lectura de parámetros y lograr otro avance para el mejoramiento tecnológico en INOLASA.

1.2 Objetivo General

1. Desarrollar un sistema preciso y eficaz, capaz de comunicar, monitorizar y almacenar, las masas de aceite dadas por los medidores de flujos de aceite (Micro Motions Transmitter), tanto en la entrada como en la salida del proceso de refinería de aceite de INOLASA, para la supervisión de este.

1.3 Objetivos Específicos

1. Realizar comunicación, con cada medidor de flujo de la refinería de aceite de INOLASA (Micro Motions Transmitter).

2. Diseñar un sistema microprocesado, capaz de recibir datos de los 10 Micro Motions Transmitter.

3. Implementar un sistema capaz de comunicar los datos unificados de los Micro Motions Transmitter a la red local de Inolasa.

4. Diseñar e implementar un programa ejecutable capaz de obtener los datos de los Micro Motion Transmitter, enviados por el microprocesador a la red local y almacenarlos en una base de datos en un servidor.

5. Desarrollar sistema que permita capturar los datos almacenados en el servidor y desplegarlos en las computadoras del departamento de producción de Inolasa.

Capítulo 2: Marco Teórico

En este apartado se van a desarrollar los temas analizados durante la elaboración del proyecto, de modo que se logre comprender la solución del proyecto de graduación. Dichos temas comprenden: comunicación serial, protocolo RS 485, Modbus RTU, protocolo 4 a 20 mA, protocolo de comunicación Hart, comunicación SPI, comunicación Ethernet, base de datos, red local.

2.1 Comunicación Serial

La comunicación serial o comunicación serie es un tipo de comunicación secuencial que transmite la información bit a bit, uno tras otro, por un único canal; por lo tanto, se envía un solo bit en cada momento, respetando los tiempos entre ellos. [15]

La ventaja de la comunicación serial es su número pequeño de líneas de transmisión, pues en una línea se puede enviar el total de bits que compone la información. [15]

2.2 RS 485

RS 485 es un protocolo de comunicación estandarizado, también conocido como EIA-485 y se utiliza para interconectar distintos de instrumentos o dispositivos, con su característica de conectar a gran distancia y en ambientes ruidosos. La velocidad de transmisión de RS485 es inversamente proporcional a la distancia de transmisión, donde la distancia puede alcanzar hasta 1200 metros. [13]

Algunas características importantes, es que permite conectar varios dispositivos en las mismas líneas; además, permite protocolos de comunicación de dispositivos, como lo es Modbus.

2.3 Modbus RTU

Modbus RTU es un protocolo de comunicación digital serial, utilizado para una comunicación elaborada entre maestro y esclavo o bien, cliente y servidor. Puede utilizar estándares de capa física como RS-232, RS-422 o RS-485 para la interconexión de los distintos dispositivos. El código de función de Modbus RTU es de 32 bits de punto flotante y es utilizado de formato entero. [12]

Su formato de comunicación entre maestro y esclavo empieza cuando el maestro envía una consulta a los esclavos involucrados, donde los esclavos responden a la petición realizada por el maestro, de acuerdo con el código de la consulta. En este formato, solo debe de existir un maestro y puede haber hasta 247 esclavos. Cuando el maestro envía una petición, todos los esclavos involucrados la reciben. Sin embargo, el único esclavo que responde, es aquel al que el maestro le indica; por lo tanto, cada esclavo debe tener una única dirección, la cual puede ser del 1 hasta el 247. En la figura 2.1 se observa la estructura del paquete del dato enviada por el maestro y de esta misma manera la información que el esclavo le responde. [14]

Figura 2.1 Formato de la comunicación de Modbus RTU. [1]

Como se indica en la figura 1, el formato de la comunicación Modbus RTU se estructura por una dirección, la función, el conjunto de datos y un control de error, para formar un total de 8 bytes. La dirección está formada de 1 byte (8 bits); la función también está formada por 1 byte y pueden tener distintas acciones según el código de la función. En la tabla 2.1 se observa cuál acción corresponde al código de la función. Luego de la función sigue la traza de datos, que se conforma por 4 bytes (32 bits), donde los primeros 2 bytes asignan la dirección del registro que se desea leer y los otros 2 bytes asignan la cantidad de registros que se van a leer. Por ultimo está el control de error, comprendido por 2 bytes, el cual es un valor que comprueba si el dato se envió correctamente.

Tabla 2.1 Especificación de códigos para las acciones de las distintas funciones del protocolo de Modbus.

Código	Acción	Significado
01	Leer Bobinas (0:xxxx)	Obtiene el estado actual ON/OFF de un grupo de bobinas lógicas.
02	Leer Entradas (1:xxxx)	Obtiene el estado actual ON/OFF de un grupo de entradas lógicas.
03	Leer Registros (4:xxxx)	Obtiene el valor binario de uno o más registros de almacenamiento.
04	Leer Registros (3:xxxx)	Obtiene el valor binario de uno o más registros de entrada.
05	Escribir Bobina (0:xxxx)	Fuerza el estado de una bobina.
06	Escribir Registro (4:xxxx)	Escribe el valor binario de un registro de almacenamiento.
15	Escribir Bobinas (0:xxxx)	Fuerza el estado de un grupo de bobinas.
16	Escribir Registros (4:xxxx)	Escribe el valor binario de un grupo de registros de almacenamiento.

2.4 Comunicación Hart

El protocolo de comunicación Hart, conocido así por sus siglas en inglés (Highway Addressable Remote Transducer), desarrollado a final de los años 1980s, fue diseñado para aplicaciones industriales. Posee la habilidad de realizar una combinación de comunicación analógica de 4 a 20 mA, mientras comunica información agregada sobre una señal digital, por lo que es llamado protocolo híbrido. [6]

La señal digital es capaz se ser bidireccional, por lo tanto, puede viajar en ambas direcciones de la comunicación; sin embargo, la señal analógica solo puede enviar la información en una sola dirección. [6]

2.5 Protocolo 4 a 20 mA

El protocolo de señal analógica de 4 a 20 mA es un método de transmisión de corriente, la cual emplea una señal de corriente en cd con un rango de 4 mA para el límite inferior y 20 mA para el límite superior y tiene como característica el ser una respuesta lineal. Como se observa en el ejemplo de la figura 2.2, si se desea medir un porcentaje de 0% a 100% y se le define el rango de 4 a 20 mA respectivamente, tendrá una relación de 12 mA al 50% ó de 16.8 al 80%. La señal de corriente puede llegar a transmitir hasta distancias de 3 Km, lo cual permite tomar medidas de equipos remotos en las plantas de grandes extensiones. [16]

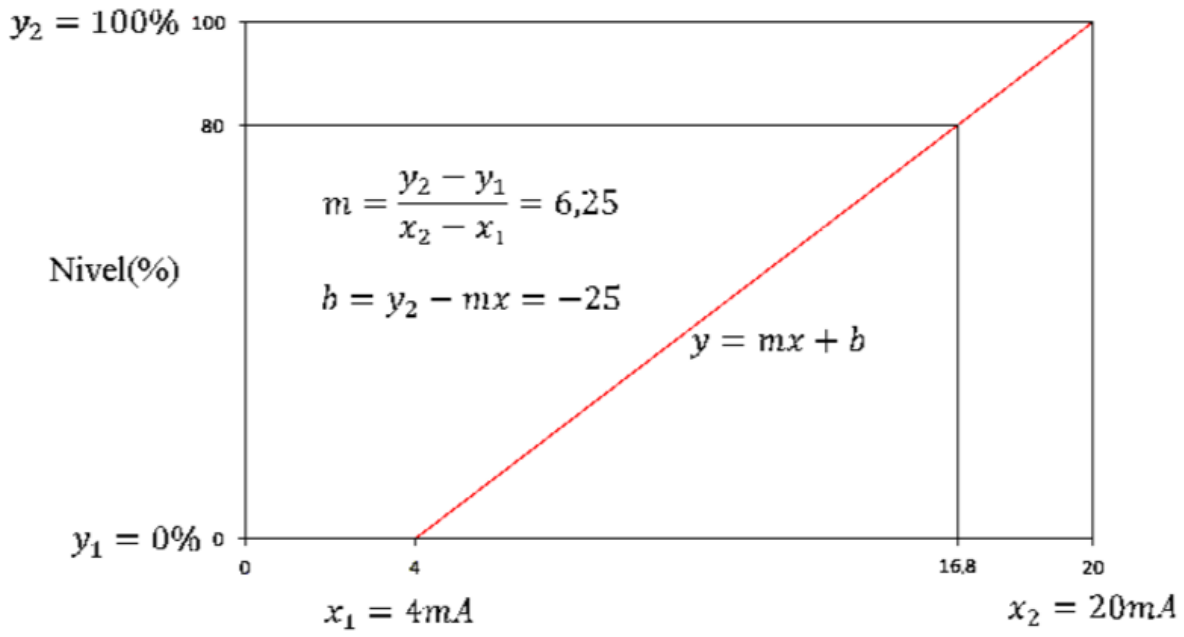


Figura 2.2 Representación de gráfica de la salida del transmisor del protocolo 4 a 20 mA.

2.6 Comunicación SPI

La comunicación SPI, por sus siglas en inglés, Serial Peripheral Interface, es un protocolo serial, sincrónico y full-duplex y se utiliza en la comunicación entre microcontroladores y periféricos. Permite controlar cualquier dispositivo electrónico digital que acepte un flujo de bits en serie, sincronizado, o regulado por un reloj. [15]

La comunicación se da entre maestro y esclavo; el maestro inicia la transmisión de datos y genera la señal de reloj y el esclavo solo responde. [15]

El protocolo SPI requiere de cuatro líneas para realizar la comunicación. Una de las líneas es llamada “SCK”; se encarga de enviar a todos los dispositivos la señal de reloj generada por el maestro. Otra de las líneas es llamada “SS”, con ella el maestro se encarga de escoger con cuál esclavo va a establecer la comunicación. Otra es llamada MOSI por sus siglas en inglés (master out, slave in); se utiliza para enviar los datos desde el maestro hacia el esclavo elegido por “SS”. La otra línea es MISO por sus siglas en inglés (master in, slave out) y se utiliza para que el esclavo pueda enviarle la respuesta al maestro. [15]

En la figura 2.3 se representa el esquema de las líneas de comunicación entre un maestro y esclavo y entre un maestro y tres esclavos.

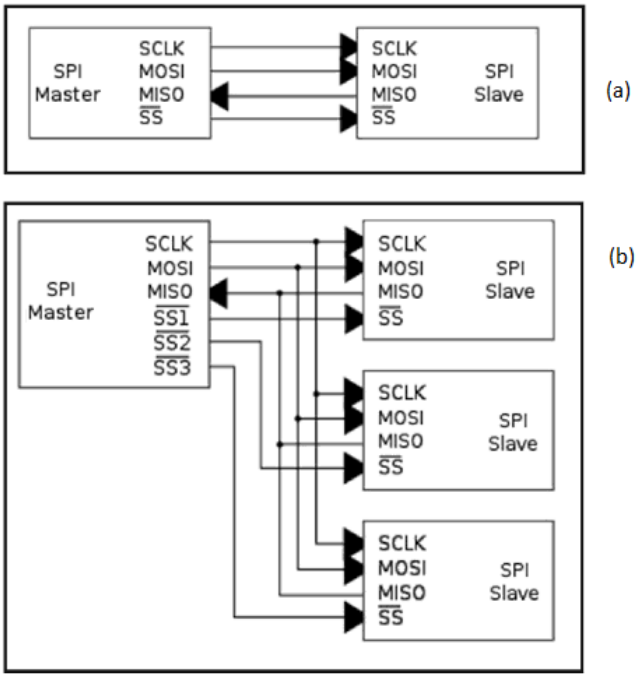


Figura 2.3 (a) Esquema de línea de comunicación entre maestro y un esclavo. (b) Esquema de línea de comunicación entre maestro y tres esclavos. [15]

2.7 Red de área local

Una red de área local, también conocida como LAN por sus siglas en inglés (Local Area Network) es una red formada por dos o más ordenadores, limitados por un espacio físico, el cual puede ser una habitación, casa, edificio, u otro. [2]

Las redes de áreas locales se utilizan, sobre todo, para la conexión de ordenadores en áreas de trabajo, pues al estar conectadas entre sí, se pueden compartir recursos o intercambiar datos.

Entre las características más importantes de una LAN están, por ejemplo: tienen limitantes en su extensión; poseen una tasa baja de errores, lo que las convierte en una

fuente de comunicación confiable; son de uso privado, es decir, son redes de una misma empresa, hogar u otro y tienen acceso restringido, a diferencia del internet, que es una red pública y todos tienen acceso. Además, tienen capacidades altas de transmisión, pueden llegar a decenas de gigabits por segundo, para las redes cableadas más rápidas; se les puede cambiar hardware y software con facilidad; se pueden conectar varias LAN entre sí. [2]

Se requieren equipos para poder realizar una red de área local, como lo son los medios de transmisión; pueden ser alámbricos o inalámbricos. También se necesitan adaptadores de red o tarjetas de interfaz de red. Los dispositivos de interconexión dependen del número de ordenadores y secciones que vaya a tener la LAN. Los dispositivos de interconexión pueden ser: concentradores (hub), conmutadores (switch), encaminadores (routers). [2]

2.8 Sistema de Base de Datos

Un sistema de base de datos, básicamente es un sistema computarizado para guardar registros y cuya finalidad general es almacenar información y permitir que usuarios puedan recuperar y actualizar dicha información. [14]

Por otra parte, una base de datos es un conjunto de datos persistentes, utilizados por los sistemas de aplicación de alguna empresa dada. [14]

En la figura 2.4 se observa un sistema de base de datos; sus principales componentes son: datos, hardware, software y usuarios.

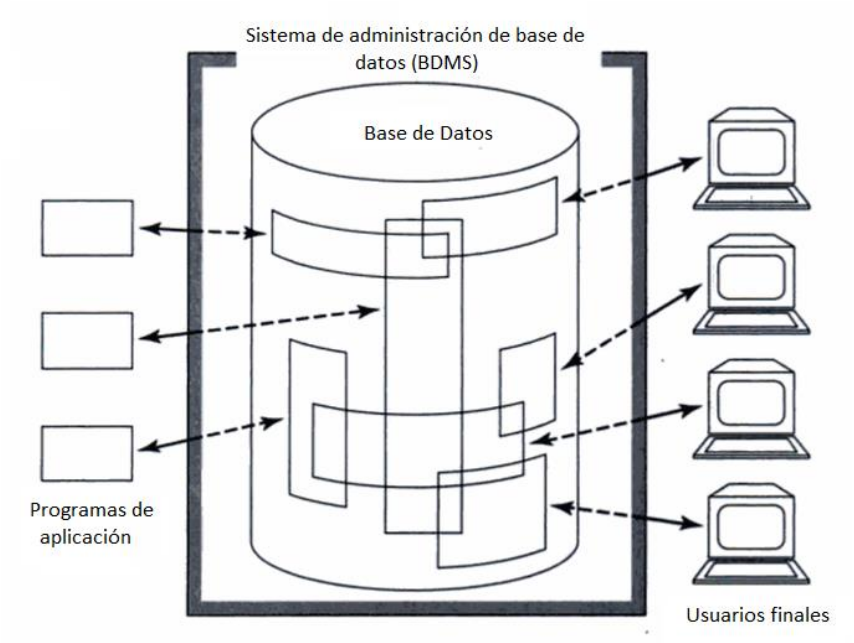


Figura 2.4 Imagen simplificada de un sistema de base de datos. [3]

Capítulo 3: Diseño y desarrollo de solución

En este capítulo se presentan tres propuestas para la solución del problema y se selecciona una de las propuestas con su debida justificación. Además, se presenta todo el proceso de diseño de la solución seleccionada.

Como punto de partida para seleccionar la solución, se definen algunos requerimientos de esta, los cuales se presentan a seguidamente:

- Diseño de un sistema físico robusto, capaz de soportar ambientes ruidosos, calientes y con presencia de vapores.

- Diseño de un sistema capaz de tomar mediciones automáticamente, sin la necesidad de ser operado ni monitorizado.

- Debe haber una conexión segura de los datos, entre la refinería de aceite y el servidor de la empresa.

- Debe ser capaz de almacenar los registros medidos.

- Debe ser escalable; en otras palabras, debe poseer la característica de tener un método simple para aumentar sus capacidades de medir más equipos, en caso de conectar más instrumentos de medición a futuro.

- Capaz de realizar cálculos necesarios para las mediciones necesarias.

3.1 Análisis y selección de solución

En cuanto a la solución del problema, se plantean tres posibles métodos para realizar el proyecto; estos se presentan con la explicación respectiva. Luego se evalúan las

diferentes alternativas, para seleccionar la solución que satisfaga los requerimientos del proyecto, y esta sea desarrollada.

3.1.1 Solución 1: Lectura de datos elaborada por un controlador lógico y transferencia de datos por radio frecuencia para almacenamiento en base de datos

Se adquieren los datos de cada medidor de flujo (Micro Motion Transmitter) de la refinería, mediante un controlador lógico, con un periodo determinado, capaz de comunicarse por distintos protocolos, debido a que unos de los medidores de flujos de aceite son de diferente modelo y no todos tienen los mismos protocolos de comunicación. Entre ellos están ModBus RTU, señal de 4 a 20 mA y señal de pulsos de frecuencia.

Luego de la adquisición de datos se necesita un módulo de radio frecuencia capaz de transferir los datos de los medidores de flujos, ubicados en la planta de refinería, a la oficina donde se encuentra el servidor de la empresa y mediante un sistema receptor RF, tomar los datos y agregarlos a una base de datos en el servidor, mediante un software realizado.

3.1.2 Solución 2: Lectura de datos por mediante software y comunicación por medio de una red local

Mediante un programa realizado por software se realizarán las lecturas de datos de cada medidor de flujo. Dicho programa deberá tener la capacidad de hacer lecturas en distintos instrumentos con diferentes protocolos de comunicación.

El software se ejecutará en el servidor de la empresa y se deberá realizar una comunicación mediante una red local, por medio de un conmutador, para poder comunicar el software con los Micro Motions Transmitter ubicados en la planta de refinería de Inolasa.

También se deberá usar un dispositivo conversor de señales, que permita realizar la conversión para poder acoplar los distintos equipos utilizados.

3.1.3 Solución 3: Lectura de datos elaborada por un controlador lógico y comunicación por medio de una red local para el almacenamiento en base de datos

Se adquieren los datos de cada Micro Motion Transmitter de la refinería, mediante un controlador lógico con un periodo determinado, capaz de comunicarse de distintas formas; como, por ejemplo: los protocolos ModBus RTU y señal de flujo de corriente de 4 a 20 mA y señal de pulsos de frecuencia, para realizar la comunicación con los distintos medidores de flujo.

Con los datos procesados en el controlador lógico, se deberá comunicar con un servidor localizado en el área de producción y administración; por ello, se utilizará la red local de la empresa para la comunicación de los datos por medio de un conmutador.

Finalmente, se necesitará un programa que se ejecuta en el servidor, capaz de tomar los datos de la red local y almacenarlos en la base de datos.

3.2 Selección de la solución

A partir de las tres soluciones planteadas, se procede a escoger la que mejor cumpla y satisfaga la necesidad del proyecto. Para ello, se evalúan y se comparan en distintos aspectos importantes, como la robustez, funcionalidad, seguridad e implementación; así se elegirá la solución por implementar.

En la tabla 3.1 se muestran las distintas soluciones; estas se evalúan, para luego ser comparadas. Para evaluarlas se calificarán del 1 al 5; 1 es la calificación mínima (muy malo) y cinco la calificación máxima (muy bueno).

Tabla 3.1 Comparación de las propuestas para la solución del proyecto.

Característica	Solución 1	Solución 2	Solución 3
Robustez	5	5	5
Funcionalidad	5	5	5
Seguridad	1	3	5
Implementación	3	5	5
Total	14	18	20

Como se observa en la tabla 3.1, todas las soluciones muestran un valor de 5 en el apartado de robustez; ello, debido a que las tres soluciones se igualan en la condición de ocupar tanto un circuito físico, como un programa realizado por software. Además, luego de ejecutar cualquiera de las tres soluciones, no se necesita un usuario externo que las esté operando.

Por otra parte, en la funcionalidad las tres soluciones también marcan el valor de *muy bueno*, debido a que todas pueden llegar a lograr el objetivo, pero de distintas maneras.

En lo que respecta la seguridad, se muestra en la tabla 3.1 que la solución 1, tiene una calificación de 1 (muy malo); esto se debe a su método de transferencia de datos de la planta de refinería al lugar de recepción, pues se da por radio frecuencia y debido a las condiciones de la planta y demás equipos utilizados en el área, se pueden ocasionar perdidas de información. La solución 2 por su parte, tiene un valor de 3, debido a que, si la red local sufre de errores, se perderá la comunicación y datos importantes serán omitidos. Por otra parte, a la solución 3 se le asigna un valor de 5, debido a que, si la red local llegase a sufrir de errores, el microcontrolador podrá guardar en variables los flujos medidos y luego podrán ser enviados nuevamente.

En cuanto a la implementación, la solución 1 tiene una calificación de 3 debido al costo de la conexión de radio frecuencia, y por otro lado, la solución 2 y 3 tiene una calificación de 5, debido al nivel de simplicidad de conexión en su circuito físico.

Por lo tanto, basados en las comparativas anteriores y de acuerdo con la tabla 3.1, se debe implementar la solución 3, para el desarrollo del sistema de comunicación y almacenaje en base de datos de los flujos de aceite de la refinería de Inolasa.

3.3 Desarrollo del Diseño

Según la propuesta seleccionada en el apartado 3.2, se escoge la solución que consiste en realizar la lectura de datos elaborada por un controlador lógico y la comunicación por medio de una red local para el almacenamiento en base de datos; esto, para la finalización del objetivo del proyecto. En la figura 3.1 se muestran, de forma general, las etapas de la solución.

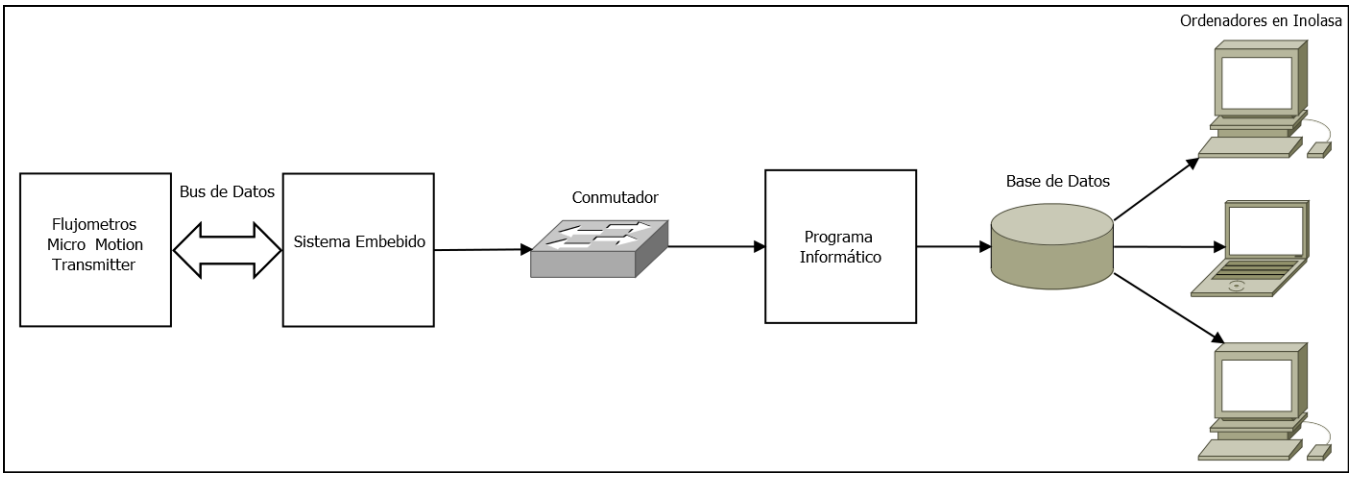


Figura 3.1 Representación de las etapas de la solución por ejecutar.

Se observa que en la figura 3.1, en la primera etapa del proyecto se encuentran los medidores de flujo; estos medidores poseen las características de mostrar datos en un monitor y tienen distintas señales de salidas con distintos protocolos de comunicación, dependiendo de su modelo. Entre los medidores de flujo que posee la empresa y los que se usarán para el proyecto, algunos de ellos poseen protocolos de comunicación Modbus, pulsos de frecuencia y señal de corriente de 4 a 20 mA. La gran parte de ellos poseen los tres protocolos; sin embargo, también se usarán algunos modelos que únicamente poseen señales de salida de 4 a 20 mA.

Los medidores de flujos ya se encuentran instalados en las refinerías; sin embargo, no están configurados todos sus puertos de salida. Por ejemplo, existen instalados unos medidores de flujos con salidas con el protocolo 4 a 20 mA conectados a los PLC para el

control de su proceso y las demás salidas no se encuentran configuradas y están fuera de uso.

Como se ve en la figurar 3.1, luego de los medidores de flujo se encuentra el controlador lógico programable. Este controlador lógico debe ser capaz de hacer las lecturas de los datos de cada medidor de flujo con base en el protocolo de comunicación que se esté utilizando.

Se debe resaltar que los medidores de flujo Micro Motion Transmitter, que poseen el protocolo de comunicación Modbus, tienen múltiples registros en los cuales se almacena información importante, por ejemplo: las variables de temperatura, el flujo másico, flujo volumétrico, masa total, otros. Por otro lado, el protocolo de comunicación del flujo de corriente de 4 a 20 mA, solo brinda la información del flujo másico o flujo volumétrico.

Por lo tanto, en los medidores de flujo que se deban usar, el protocolo de comunicación de flujo de corriente de 4 a 20 mA, se deberá hacer un cálculo matemático en el controlador para poder brindar la información requerida, la cual es la masa en kilogramos; caso contrario sucede con los medidores de flujo que utilice Modbus, porque al hacer la lectura del registro adecuado, se obtiene el valor de la masa.

Luego, el controlador deberá tener un módulo que permita conectarlo a una red local, para poder transmitir los datos medidos. En la figura 3.1 se observa que el controlador lógico programable, se conectará a un conmutador que permitirá realizar la conexión entre el controlador ubicado en la refinería y el servidor de Inolasa ubicado en las oficinas de producción.

En el servidor de Inolasa se ejecutará un programa informático, el cual tomará de la red local, las mediciones realizadas por los medidores de flujo en las refinerías, identifica la información de cada Micro Motion Transmitter y las envía a una base de datos en dicho servidor, para así ser almacenadas en el registro en la empresa.

Finalmente, se realiza un programa para ejecutar en las computadoras del personal, para poder obtener remotamente los datos de los medidores de flujos y realizar sus respectivos estudios de producción.

Seguidamente, se explicarán las distintas etapas por realizar para la elaboración del sistema final.

3.3.1 Configuración de los medidores de flujo

Como se ha mencionado anteriormente, los medidores de flujo poseen protocolos de comunicación, como son la salida de flujo de corriente de 4 a 20 mA, ModBus RTU, pulsos de frecuencia, entre otros.

En el presente proyecto, 7 de los medidores de flujo se comunicarán mediante el protocolo de modbus RTU y otros 3 mediante el protocolo de flujo de corriente de 4 a 20 mA. Sin embargo, estas salidas deben ser configuradas; en el caso del protocolo de salida de flujo de corriente de 4 a 20 mA, se debe definir el número de salida, calibración de puntos máximo y mínimo de lectura entre otros aspectos.

Para la configuración de los medidores de flujo se utiliza la herramienta DevCom2000, la cual es un software que permite una comunicación Hart con otros dispositivos. Al realizar esta comunicación se debe conectar mediante un conversor de USB a Hart, de la computadora, con el programa DevCom2000 a los pines de Hart que tienen los medidores de flujo, y entre los pines se deberá conectar una resistencia de carga entre 260Ω a 600Ω.

Para la configuración de los Micro Motion Transmitter que van a utilizar el protocolo de Modbus RTU, se debe asignar un número de esclavo diferente para cada medidor. Como se mencionó, Modbus RTU es un protocolo de comunicación digital serial, que puede ser utilizado para una comunicación elaborada entre maestro y esclavo. En este caso, cada medidor de flujo será asignado como un esclavo diferente del 1 al 7, los cuales son los que utilizará este protocolo.

También se configura la velocidad a la cual se le asignan 9600 baudios; por lo tanto, se debe establecer la misma velocidad en el microcontrolador. Se asigna que no posea bit de paridad ni tampoco bit de parada.

Se debe de conocer el formato del dato que responde a cada esclavo (los Micro Motion Transmitter). De acuerdo con el suplemento al manual de configuración y uso de los Micro Motion Transmitter [4], los datos de los registros tienen una estructura de punto flotante, como se observa en la tabla 3.2, donde el byte 3 y 4, son los valores alto y bajo del primer registro leído y de la misma manera los bytes 1 y 2 del segundo registro.

Tabla 3.2 Estructura de Bits de los Bytes de punto flotante. [4]

Byte	Bits	Definición
0	SEEEEEEE	S = Signo E = Exponente
1	EMMMMMMM	E = Exponente M = Mantisa
2	MMMMMMMM	M = Mantisa
3	MMMMMMMM	M = Mantisa

Por otra parte, para los otros 3 medidores de flujo, se va a utilizar la salida de corriente de 4 a 20 mA; sin embargo, con este protocolo solo se puede obtener el flujo de masa en kilogramos por minuto y no el acumulado de masa total.

Una forma de obtener el dato de la masa total, es realizando la lectura del flujo de masa en kilogramos por minuto y posteriormente a eso, realizar el cálculo de la masa acumulada en un periodo determinado.

Para realizar la medición del flujo de masa de los Micro Motion Transmitter, se realizó la configuración de que el flujo de corriente sea 4 mA cuando el flujo de masa sea 0 y la corriente sea 20 mA cuando la medida es 200 Kg/min. En la figura 3.2 se observa la respuesta de este protocolo.

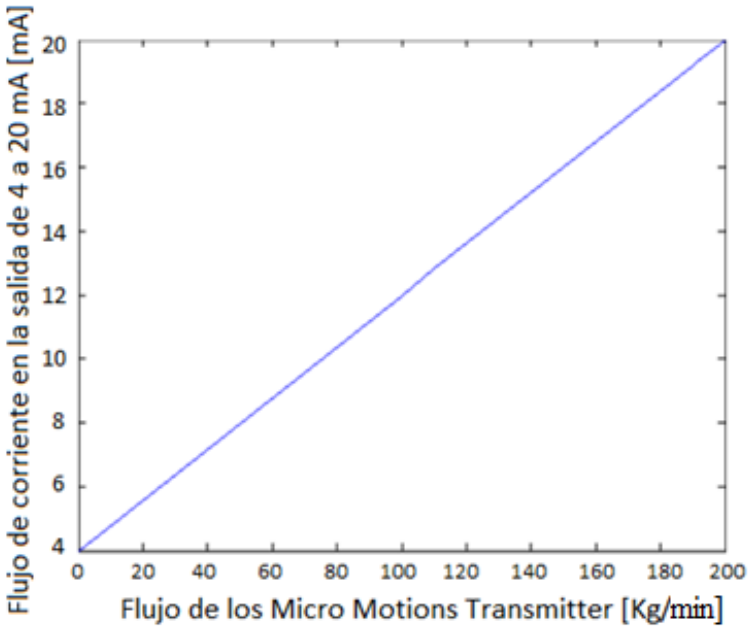


Figura 3.2 Gráfica de flujo de corriente de salida de 4 a 20 mA vs el flujo de masa medido en los Micro Motion Trasmitter.

Donde la ecuación de la recta de la gráfica de la figura 3.3 es:

$$y = 0.08x + 4 \qquad (3.1)$$

Luego de tener la salida de 4 a 20 mA configurada, se agrega una carga en la salida de 220Ω, para así tener una tensión conocida y poder realizar la medición con un microcontrolador. En la figura 3.3 se observa la tensión en la carga aplicada en la salida con respecto al flujo de corriente en la salida de los Micro Motions Transmitter.

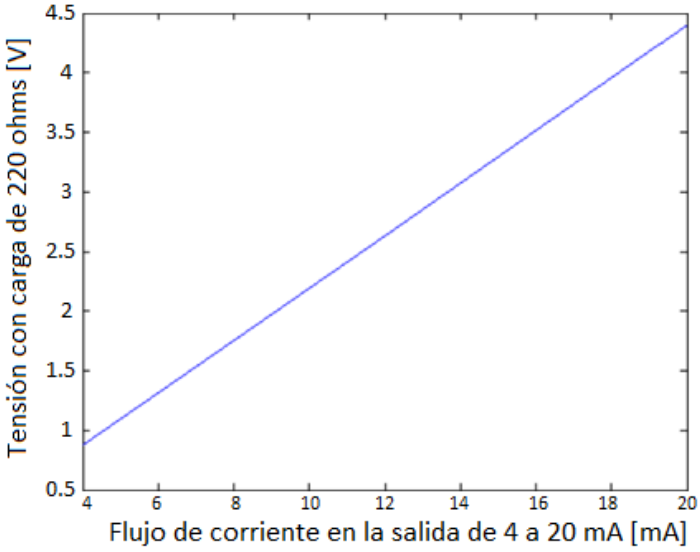


Figura 3.3 Gráfica de la tensión en una carga de 220 Ω vs corriente en la salida de los Micro Motions Transmitter.

Donde la ecuación de la recta de la gráfica de la figura 3.3 es:

$$y = 220x \qquad (3.2)$$

De acuerdo con la figura 3.4 los límites de tensión obtenidos en la carga son 0.88 V para cuando el flujo de masa sea 0 hasta 4.4 V cuando la masa sea 200 Kg/min

Conociendo la respuesta de la figura 3.3, se debe decodificar la señal de tensión medida por el microcontrolador para obtener el dato del flujo de masa. Utilizando la figura 3.2 y la figura 3.3, sea realiza la relación de flujo de masa contra la tensión de la carga en la salida. En la figura 3.4 se observa la gráfica de la relación lineal del flujo de masa y la tensión medida, donde los límites son, de 0 Kg/min para una tensión de 0.88 V y cuando la tensión en la carga de salida es 4.4 V el flujo es de 200 Kg/min.

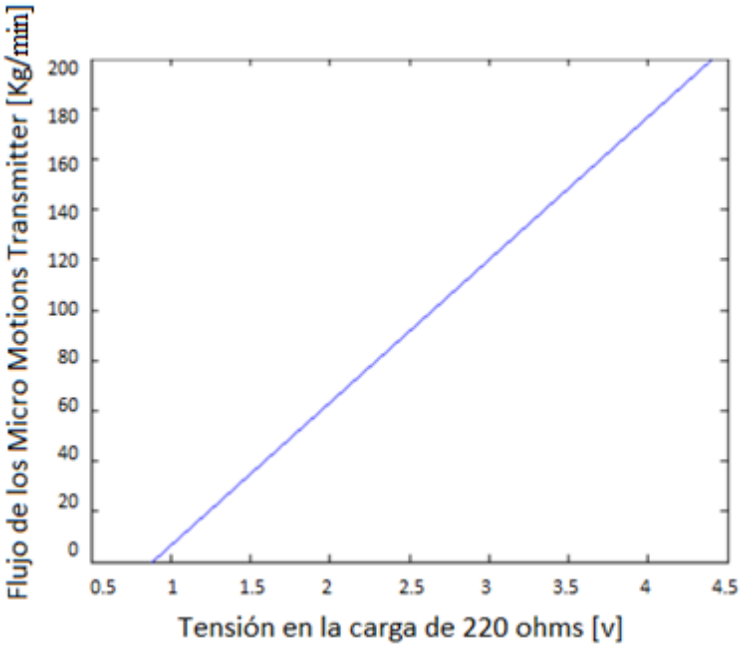


Figura 3.4 Gráfica del flujo de corriente medido de los Micro Motions Transmitter vs tensión medida en la carga de 220Ω.

De la figura 3.4 se puede obtener la ecuación de la recta, la cual es:

$$y=56.8181x - 50 \qquad (3.3)$$

Donde la ecuación (3.3) se deberá realizar en el microcontrolador para obtener el dato del flujo con respecto a la salida de corriente.

3.3.2 Diseño y Desarrollo de la programación del controlador lógico

Para el desarrollo del controlador lógico del proyecto, se debe escoger el controlador indicado, que supla la necesidad del proyecto y también posibles necesidades en trabajos futuros, en caso de llegar a recibir cambios. Por lo tanto, un aspecto importante de considerar es en la escalabilidad del controlador; que tenga múltiples entradas y salidas, entre ellas, tanto analógicas como digitales. De la misma manera, que posea diferentes herramientas y librerías para ser capaces de cumplir con las tareas por realizar.

Se utiliza un microcontrolador Arduino UNO, debido a que puede ser escalado fácilmente, además de ser capaz de realizar las tareas del proyecto y no gastar tanto recursos, como sí sucedería si se usara un PLC, Logo u otros microcontroladores.

Para la ejecución del proyecto, se debe realizar con el microcontrolador una comunicación usando el protocolo de Modbus RTU, comunicación vía Ethernet, interrupciones, lecturas de señales analógicas, tareas ejecutadas en tiempos determinados y operaciones matemáticas simples.

En la figura 3.5 se ilustra el diseño del diagrama de flujo de la ejecución del proceso, el cual se usa de guía para implementar el microcontrolador.

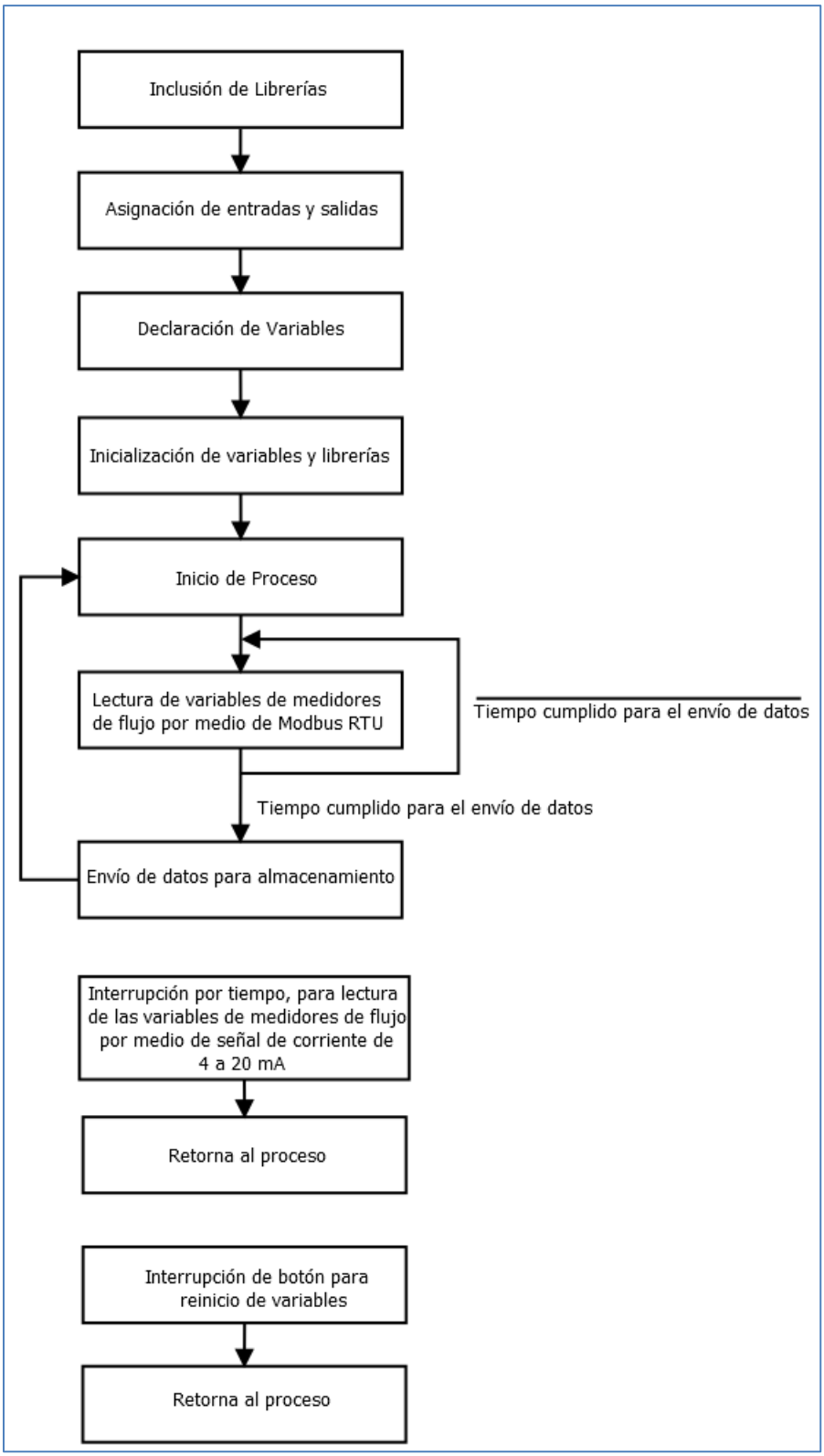


Figura 3.5 Diagrama de flujo de la ejecución del proceso del microcontrolador del sistema.

Como se observa en la figura 3.5, se inicializa el diagrama de flujo con las inclusiones de las librerías; entre ellas se encuentran la librería para la utilización del protocolo Modbus RTU, la librería para comunicación serial, librerías de temporizadores, librería de comunicación SPI y Ethernet para realizar la comunicación usando un módulo de Ethernet para el microcontrolador.

Después, se asignan las entradas y salidas que se van a utilizar. Seguidamente, se declaran las variables que se utilizarán, como por ejemplo las variables de los valores flotantes de las masas por medir. Después, se configuran librerías que se van a utilizar, como la asignación de la velocidad de la comunicación serial, o algunos tiempos para la función de los temporizadores o direcciones IP para la comunicación Ethernet.

Luego de concluir con la inicialización de las librerías, empieza la ejecución del proceso, el cual inicia con la lectura de los registros de masa total por medio de la comunicación Modbus.

Como se observa en la figura 3.5, la ejecución del microprocesador permanecerá dentro del ciclo de la comunicación de Modbus RTU, hasta cumplir un tiempo establecido de 1 minuto. En este ciclo se establece que la función es el número 03, según la tabla 2.1; también se asigna la dirección del registro que se va a leer, cuya dirección es 258 y se especifica que son 2 registros por leer, debido a que el dato de la masa se encuentra en los registros 259 y 260 tal y como se especifica en el manual del Transmisor modelo 2700 de Micro Motion ([5]). Esto se realiza repetidamente, pero cambiando la dirección del esclavo, para realizar la lectura a cada uno de los medidores de flujo.

El dato devuelto por los Micro Motion Transmitter, tienen el formato de punto flotante, de acuerdo con la tabla 3.2; el orden de los bytes de la respuesta es 3, 4, 1 y 2, ya que se lee primero el registro 259 y después 260. Se debe de separar esta información para acomodar los bytes en el orden 1, 2, 3 y 4 para así convertir el dato al valor flotante.

Después de cumplir el ciclo de lectura de Modbus, inicia un ciclo para enviar los datos por Ethernet hacia la red local, mediante dispositivo W5100, el cual es un complemento del microcontrolador con conexión Ethernet. Seguido, se inicia nuevamente el proceso.

Como se observa en la figura 3.5, existe una interrupción, la cual se encarga de tomar el dato de los medidores de flujo que utilizan las salidas de corriente de 4 a 20 mA. Como se mencionó anteriormente, en los medidores de flujo que utilizan la salida de 4 a 20 mA, lo que se obtiene es del flujo de masa por minuto; por ello, se debe calcular la masa que ha trascurrido en un tiempo determinado. Por tal razón, se crea esta interrupción que se activa cada segundo.

Para tomar el dato, se mide la tensión generada en la carga colocada en la del medidor de flujo. Se habilita las entradas analógicas del microcontrolador y se lee la tensión.

Como el dato que se toma es una tensión, utiliza ecuación (3.3) para identificar el flujo medido de los Micro Motions Transmitter respecto de la tensión medida.

Luego de tener el flujo de masa, se calcula la masa total de la siguiente manera:

$$\text{Masa} = \text{Flujo}\left[\frac{Kg}{min}\right] \times \frac{1\,min}{60\,s} \times 1\,s \tag{3.4}$$

Finalmente, en el diagrama de flujo de la ejecución del proceso, se encuentra una segunda interrupción, pero está en función de la acción de un botón. En esta interrupción se borra el valor de las variables que almacena la masa de los medidores de flujo que usan la salida de 4 a 20 mA.

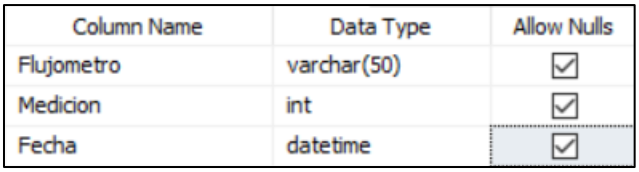
3.3.3 Creación de la base de datos

Para almacenar los datos de los medidores de flujo, se debe de crear una tabla en una base de datos en el servidor de Inolasa. La tabla se conforma de 3 columnas; una de sus columnas corresponde para identificar el medidor de flujo, la segunda almacena la masa registrada y la última columna corresponde a la fecha y la hora en que se registró el dato.

Para crear la base de datos, se utiliza el programa de Microsoft SQL Server Management Studio, el cual es un software que se encuentra instalado en el servidor de Inolasa.

Como primer paso, se debe crear la base de datos en el programa Microsoft SQL Server Management Studio, lo cual se observa paso a paso en el apéndice A.1. Posteriormente se debe realizar la tabla con las especificaciones anteriores para almacenar los datos.

En la figura 3.6 se muestra el diseño de la tabla y el tipo de datos correspondiente de cada columna, donde la columna “Flujómetro” es de tipo “varchar” para almacenar cadenas e identificar el medidor de flujo, la columna medición es de tipo “int” para almacenar números enteros, y columna “Fecha” va a poseer un formato “datetime” para almacenar fecha y hora.



Column Name	Data Type	Allow Nulls
Flujometro	varchar(50)	☑
Medicion	int	☑
Fecha	datetime	☑

Figura 3.6 Asignación de las columnas, el tipo de dato y opción de permitir el dato nulo de la tabla.

Con la tabla creada, se puede ingresar al editor de líneas del programa, en donde se podrán ver los datos almacenados o también ingresar nuevos. En la figura 3.7 se observa la estructura de la tabla creada con datos de prueba.

Flujometro	Medicion	Fecha
1	100	2016-01-08 13:30:00.000
2	90	2016-12-25 09:22:46.000
3	123	2016-09-29 15:27:27.320

Figura 3.7 Tabla realizada en la Base de Datos hecha en Microsoft SQL Server Management Studio.

3.3.4 Desarrollo del programa informático

Con la base de datos creada y el programa del microcontrolador implementado, se empieza el diseño del programa que enlazará las dos partes. Para ello, se utiliza el programa Microsoft Visual Studio 2010, software que la empresa posee, y se escoge el lenguaje C# para el desarrollo de la aplicación.

En la figura 3.8 se muestra el diagrama de flujo de la ejecución del programa encargado de recibir los datos enviados por el microcontrolador y almacenarlos en la base de datos creada anteriormente.

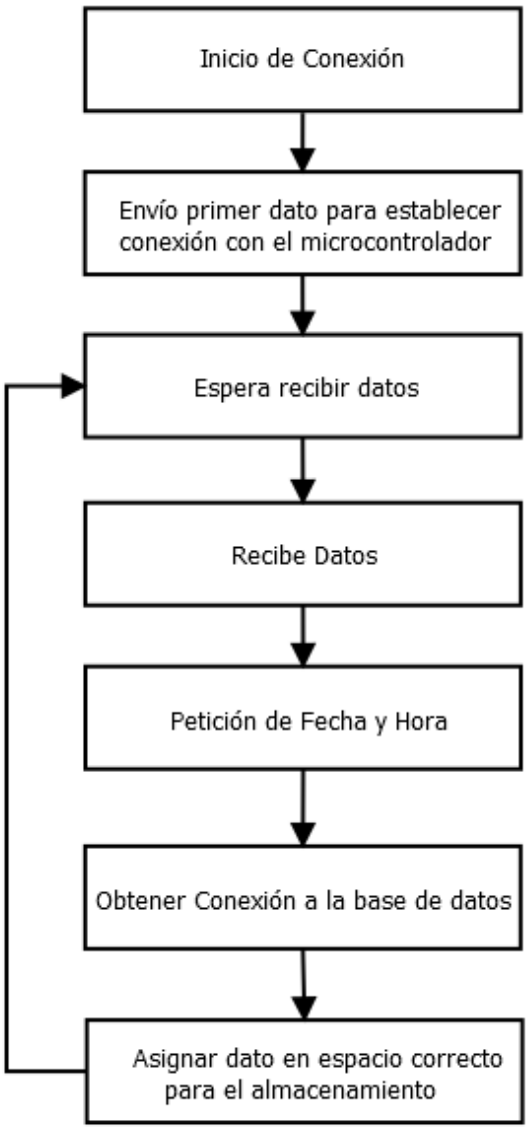


Figura 3.8 Diagrama de flujo de la ejecución de programa en C#.

Luego de abrir un nuevo proyecto de C# en Microsoft Visual Studio 2010, el programa crea el formulario y su respectiva ventana del diseño. Después de tener la ventana del diseño del formulario, se le agregan dos editores de texto, uno para ingresar la dirección IP del microcontrolador y otro para asignar el puerto de comunicación; cada uno se encuentra con su respectiva etiqueta. Además, a la ventana de diseño del formulario se agrega un botón, el cual funcionara para empezar la conexión con los datos declarados en

los editores de texto. Por último, lleva una ventana de vista de texto para poder monitorizar mensajes de la aplicación.

En la figura 3.9 se muestra la ventana de diseño del formulario del programa, con sus respectivos elementos mencionados.

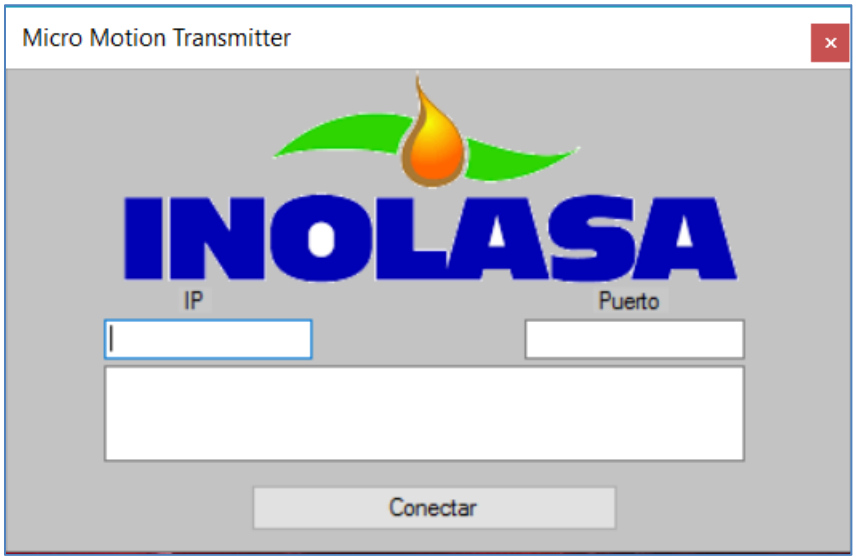


Figura 3.9 Ventana de diseño del formulario del programa en Microsoft Visual Studio 2010.

Por otro lado, se comienza el código del programa, el cual debe tener librerías que satisfagan la necesidad de la aplicación, como lo es la comunicación en una red local y base de datos SQL. En el código de la ventana principal del programa, se realizan las declaraciones de las funciones principales en el formulario del programa, las cuales son: dos funciones llamadas “BackgroundWorker”, donde estas se encargan de ejecutar operaciones en hilos separados y proporciona la ejecución de tareas que consumen mucho tiempo en un subproceso en segundo plano, y la tercera función corresponde a un accionamiento del botón agregado en la ventana de diseño del formulario. En el apéndice 3, en la figura A.3.1 se observa el código de la ventana principal del programa.

Para la implementación de la conexión en la red local, se utiliza la función que se encuentra accionada por el botón de la aplicación, en ella se inicializa una nueva instancia de la clase TcpClient, la cual proporciona conexiones de cliente para servicios de red TCP. Toma los valores asignados de la dirección IP del dispositivo con el que se va a comunicar y el puerto por el cual va a establecer la comunicación para iniciar una nueva instancia de la clase TcpClient y enlazarla con el extremo local especificado. Después de enlazar la comunicación, se crean variables de lectura y escritura para esta comunicación. Código obtenido y acondicionado de [10]. En el apéndice 3, en la figura A.3.2 se observa el código implementado en la función realizada por el accionamiento del botón.

Luego de ser presionado el botón de la aplicación y de enlazar la comunicación, se realiza un llamado a la función "backgroundWorker2_DoWork", usada para enviar datos por medio de la red local. Esta función realiza el envío de un texto en caso de que el cliente se encuentre conectado; en caso contrario, se envía una alerta. Código obtenido y acondicionado de [10]. En el apéndice 3, en la figura A.3.3 se observa el código utilizado para enviar información desde la aplicación.

La función "backgroundWorker1_DoWork", es utilizada para recibir datos del microcontrolador por medio de la red local. En esta función se inicializa un ciclo mientras que la aplicación se encuentre conectada; en él, serán seleccionados los datos de cada medidor de flujo para almacenar en sus respectivas variables.

Luego de obtener los valores de cada medidor de flujo, se realiza una petición de la fecha y la hora del ordenador donde se encuentre la aplicación ejecutada. Después se realiza un ciclo donde se crea un objeto llamado "Micromotion1" y se asigna los valores de cada medidor de flujo junto con el dato de la fecha y la hora, para ser almacenados a la base de datos; para ello, se usa una clase por separado para asignar el tipo de variable de los valores y finalmente se hace un llamado a la clase llamada "MicromotionDAL". En el apéndice 3, en la figura A.3.4 se observa el código del llamado del objeto "Micromotion1", asignación de cada valor y llamado de la clase "MicromotionDAL".

En la clase llamada “MicromotionDAL”, se crea un método llamado “Agregar”, donde se realiza la conexión a la base de datos, utilizando una clase llamada “BDComun”. Después de ello, se utiliza el comando “SqlCommand Comando” para agregar la información en la base de datos en sus respectivas columnas de la tabla. Finalmente, con el comando “execute” retorna un 1 en caso de que la conexión se haya hecho correctamente. Dicho código se puede visualizar en el apéndice 3, en la figura A.3.5.

Para realizar la conexión con la base de datos, se utiliza la clase llamada “BDComun” para especificar la base de datos con la que se hará la conexión. Para ello, se crea un método llamado “ObtenerConexion” y dentro del método un objeto llamado Conn, donde se especifican los datos para acceder a la base de datos. Se puede visualizar el código implementado para la clase BDComun en el apéndice 3, en la figura A.3.6.

Capítulo 4: Validación y Análisis de Resultados

En este capítulo se mostrarán los resultados obtenidos para validar la elaboración de cada etapa explicada en el capítulo 3. Para ello, se realizaron pruebas de cada etapa por separado, para la verificación de su funcionalidad. Por último, se validó el funcionamiento del resultado final para el desarrollo de un sistema de comunicación y almacenaje en base de datos, de los flujos de aceite medidos por los Micro Motions Transmitter en la refinería de aceite de INOLASA.

4.1 Validación de la comunicación de los medidores de flujo

Para validar la comunicación de los Micro Motions Transmitter con el microcontrolador, primero se realizaron pruebas por separado de los dos tipos de protocolos de comunicación que se van a utilizar, así como, el protocolo Modbus RTU y de igual manera la comunicación de 4 a 20 mA. En la sección 4.1.1 y 4.1.2 se explican los resultados obtenidos de estos dos protocolos de comunicación.

4.1.1 Validación de la comunicación Modbus RTU

En la verificación del funcionamiento de la comunicación Modbus RTU de los Micro Motions Trasmitter, se utilizó un programa llamado ModScan32, el cual es un software elaborado para realizar comunicaciones Modbus. En el software se le asigna la dirección de esclavo, el número del registro que se desea leer, la cantidad de registros que se desean leer y la función por realizar, y mediante un conversor bidireccional de USB a RS-485 se realiza la comunicación.

Como se muestra en la figura 2.1, el maestro debe enviar el mensaje estructurado de la siguiente forma: dirección, función, dato y control de error. El control de error (CRC) es un sistema de detección de distintos posibles errores en el mensaje.

En la figura 4.1 se observa qué se asignó, en el programa ModScan32, el esclavo con dirección 8, la función 3 que de acuerdo con la tabla 2.1 corresponde a la lectura de registros. Se asigna el registro 259 debido a que según [5] los registros que le corresponden a la masa total son 259 y 260; además, se asigna leer 2 registros.

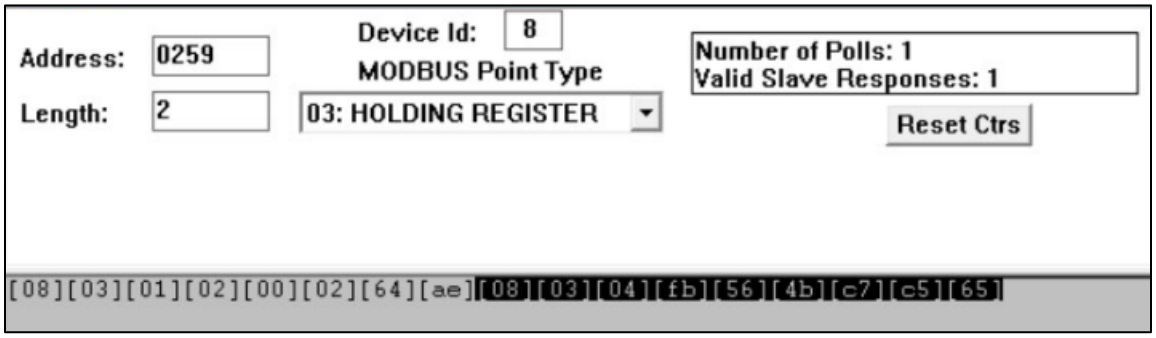


Figura 4.1 Comunicación Modbus RTU mediante software ModScan32.

Como se ve en la figura 4.1, el mensaje que se envía corresponde a un 08030102000264AE en hexadecimal. El primer byte del mensaje en hexadecimal es 08, lo que corresponde a la dirección 8. Continúa con los datos observados en la figura 4.1, el siguiente byte es un 03 en hexadecimal, que corresponde al número de la función. Seguidamente, continúan dos bytes que son 0102 en hexadecimal, los cuales son la dirección del registro por leer, donde corresponde a 258 en decimal. Los dos bytes siguientes son 0002 en hexadecimal, lo cual corresponde al número de registro que se va a leer; en este caso corresponde a dos registros y finalmente los dos últimos bytes corresponden al CRC.

De la misma manera, en la figura 4.1 se observa la respuesta que envía el esclavo, la cual es 080304FB564BC7C565 en hexadecimal, donde el mensaje se estructura de la siguiente manera: el primer byte es 08 y corresponde al esclavo, el segundo byte es 03 y corresponde la función, el tercer byte es 04 y corresponde a la cantidad de bytes que fueron leídos, los cuatro bytes siguientes son FB564BC7 y corresponden al dato que se encontraba en los dos registros leídos y finalmente los dos últimos bytes C565 corresponden al CRC.

Como se observa en la figura 4.1, el dato que se encuentra en los registros leídos es FB564BC7, donde “FB” es el dato alto del registro 259, “56” es el dato bajo del registro 259, “4B” es el dato alto del registro 260 y “C7” es el dato bajo del registro 260.

Para la interpretación del valor de la masa medida por los Micro Motion Transmitter, se debe acomodar el dato recibido, de manera que el registro 260 se encuentre por delante del registro 259. [4]

Reacomodando los bytes devueltos por el esclavo, el dato correcto se ordena 4BC7FB56. Para realizar la interpretación de la masa se debe tener el dato en formato binario, para realizar la conversión, según se explica en la tabla 3.2.

En la tabla 4.1 se obtiene el dato del registro 260 y 259 de manera ordenada y en formato binario.

Tabla 4.1 Respuesta del esclavo en el protocolo Modbus RTU en formato binario

0100	1011	1100	0111	1111	1011	0101	0110

Como se observa en la tabla 4.1, el primer bit obtenido en un “0”, lo cual representa que el signo del dato es positivo. Los siguientes 8 bits, representan el exponente, el cual tiene un valor de 151.

Los bits restantes conforman la Mantisa. De acuerdo con el formato de punto flotante o coma flotante, según la IEEE 754, la mantisa se debe colocar después un uno y una coma (1,) en formato binario, por lo que queda de manera 1,10001111111101101010110.

Finalmente, según la IEEE 754, la coma se debe correr al lado derecho, el valor del residuo del exponente menos 127. Por lo tanto, el valor se debe correr 24 espacios hacia la derecha y de ser necesario, llenar de “1” los espacios vacíos. El dato final en el formato binario es 11000111111111011010101101.

Realizando la conversión del dato de formato binario a decimal, se indica que la masa medida en kilogramos es 26212013; ello concuerda con el valor mostrado por la

pantalla indicadora del medidor de flujo, y con eso se valida el funcionamiento de la comunicación con el protocolo Modbus RTU.

4.1.2 Validación de la comunicación por 4 a 20 mA

Para determinar el correcto funcionamiento de la comunicación de 4 a 20 mA, se utiliza el simulador de señal que tiene el Micro Motion transmitter internamente. En él se asigna que brinde una salida de 4 mA y mediante un multímetro se toma la tensión en la carga de 220 Ω y se compara con la gráfica de la figura 3.4. En este caso, la tensión medida con el multímetro es de 0.87 V, por lo que existe un error de 1.136%.

De la misma manera se realiza una simulación, asignando en la salida del Micro Motion Transmitter, 20 mA. En esta simulación, el multímetro indica una tensión de 4.33 V, por lo que se genera un error de 1.59%.

Por lo tanto, existen errores relativos en los límites inferior y superior, del rango de tensión, de 1.136% y 1.59% respectivamente. En los dos casos los errores son aceptables debido a que es un error muy bajo. Sin embargo, se realiza un ajuste en la ecuación para el cálculo del flujo de masa por segundo con los datos experimentales.

Se realizó también la simulación para los otros dos Micro Motion Transmitter que utilizan la comunicación de 4 a 20 mA y con los cálculos obtenidos se completa la tabla 4.2.

Tabla 4.2 Mediciones experimentales para límites en la comunicación de 4 a 20 mA y sus respectivos errores.

Medidor de Flujo	Límite inferior	Error	Límite Superior	Error
Micro Motion Transmiter 8	0.87 V	1.136%	4.33 V	1.59%
Micro Motion Transmiter 9	0.87 V	1.136%	4.38 V	0.45%
Micro Motion Transmiter 10	0.86 V	2.27%	4.29 V	2.5%

En la tabla 4.2 se observa que los tres Micro Motion Transmitter que utiliza la comunicación de 4 a 20 mA obtienen un error bajo; sin embargo, para una mayor precisión a cada uno se ajusta los datos para obtener un flujo más exacto. Por ello, se obtienen nuevas ecuaciones para el cálculo del flujo de masa con respecto la tensión medida en la carga de la salida de 4 a 20 mA, para cada medidor de flujo. En la tabla 4.3 se observan las nuevas ecuaciones con un mejor ajuste de parámetros.

Tabla 4.3 Ecuaciones ajustadas para obtener el flujo de masa a partir de la tensión de la carga de salida de cada Micro Motion Transmitter.

Medidor de Flujo	Ecuación
Micro Motion Transmiter 8	$y=57.8034x - 50.2887$ (4.1)
Micro Motion Transmiter 9	$y=56.9800x - 49.5726$ (4.2)
Micro Motion Transmiter 10	$y=58.3090x - 50.1457$ (4.3)

Con los ajuntes en las ecuaciones para el cálculo del flujo de masa con respecto a la tensión, se obtienen las ecuaciones (4.1), (4.2) y (4.3), las cuales se utilizan en el microcontrolador y una mayor precisión en la toma del dato.

4.2 Validación del funcionamiento del microcontrolador

Para la validación del microcontrolador, se conectan los 10 Micro Motion Transmitter con el microcontrolador. En el diagrama de flujo del proceso del microcontrolador, visto en la figura 3.6, se observa que después de iniciar la ejecución correctamente, entra en un ciclo de lectura de las variables, usando el protocolo Modbus RTU durante 1 minuto. Como se observa en la figura 3.6, en este lapso se realiza la interrupción cada 1 segundo para realizar la lectura de los Micro Motion Transmitter por medio de la señal analógica de 4 a 2 mA.

Utilizando la comunicación serial, se logra monitorizar el proceso del microcontrolador conectándolo a una computadora y así verificar su funcionamiento.

En la figura 4.2 se muestra las mediciones, mediante el microcontrolador, de los Micro Motion Transmitter 1, 2, 3, 4, 5, 6, 7, las cuales fueron tomadas por medio de Modbus RTU; entre ellas se encuentran las mediciones de los Micro Motion Transmitter 8, 9, 10, las cuales fueron tomadas mediante la señal de 4 a 20 mA. De esta manera, se obtiene la validación del funcionamiento de las lecturas de masa utilizando el microcontrolador.

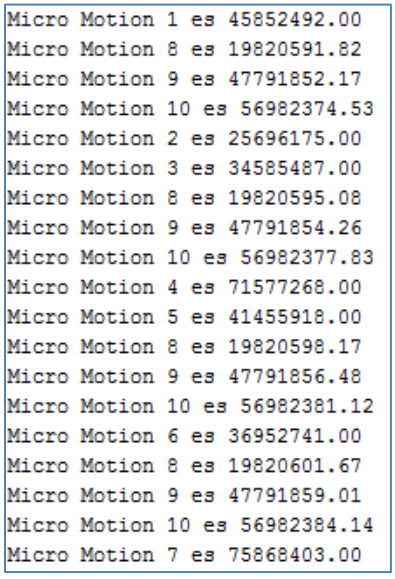


Figura 4.2 Medición de los diez medidores de flujo con el microcontrolador, usando los protocolos Modbus RTU y señal de 4-20 mA.

En la figura 4.2 se observa que en los datos de los Micro Motion Transmitter 1, 2, 3, 4, 5, 6 y 7, donde se utiliza la comunicación Modbus RTU, se encuentra el valor de la masa en kilogramos, en formato decimal. Con ello se valida el funcionamiento en el microcontrolador de la conversión realizada en el dato de formato de coma flotante a formato decimal.

Por otra parte, se verifica el funcionamiento del reseteo de las variables de los medidores de flujo comunicados por la señal de 4 a 20 mA. En la figura 4.3 se observa donde los Micro Motion Transmitter 8, 9 y 10 tienen un valor de 19820942.74, 47792159.81 y 56982604.19 respectivamente y después los valores decrecen a 3.06, 2.34 y 3.26. Esto es debido a que se inicializó de 0, los valores de los respectivos Micro Motion Transmitter, mediante la interrupción accionada por el botón.

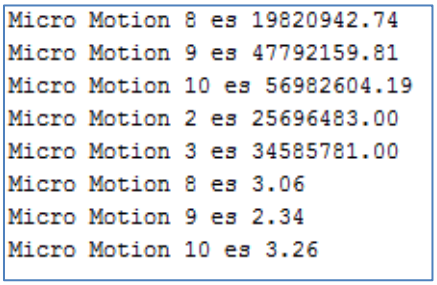


Figura 4.3 Comprobación del funcionamiento del botón.

Cuando finaliza el minuto, tiempo de la toma de datos por medio de Modbus, continúa al siguiente ciclo del envío de datos a la red de la empresa; esta validación se muestra en la sección 4.3.

4.3 Validación de la comunicación del microcontrolador con la red local

Para la validación de la comunicación del microcontrolador con la red local, se utilizó el programa llamado HyperTerminal ejecutado en una computadora, la cual mediante un cable de red se conectó a la misma red donde fue conectado el microcontrolador. Para este caso se usa Hyperterminal; sin embargo, se pudo haber utilizado cualquier software con la capacitad de realizar una conexión TCP/IP.

En el programa HyperTerminal se asigna la dirección IP del microcontrolador y el puerto por el cual se conecta la computadora a la red, para después enlazar el programa al microcontrolador y obtener los datos que se envían por ese medio.

En la figura 4.4 se observa donde el programa HyperTerminal se encuentra enlazado al microcontrolador. De esta manera, recibe los datos enviados por el microcontrolador a la red, los cuales son los datos tomados de los 10 Micro Motion Transmitter.

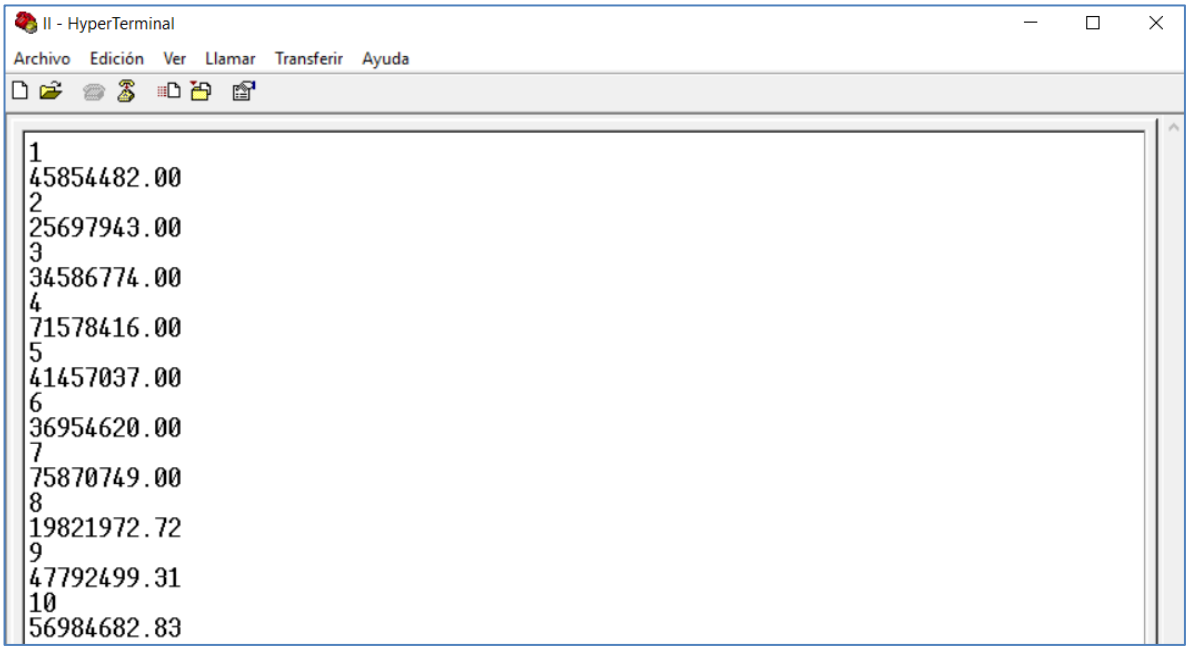


Figura 4.4 Programa HyperTerminal enlazado al microcontrolador.

Con esta verificación se valida el funcionamiento del envío de los datos a la red de Inolasa por medio del microcontrolador, y el correcto recibimiento de un programa enlazado a este.

4.4 Validación del funcionamiento del programa en C#

Para verificar el funcionamiento de la comunicación del microcontrolador y el programa desarrollado en Microsoft Visual Studio, se ejecuta el programa creado, llamado Micro Motion Transmitter, el cual se visualiza en la figura 4.5.

En el programa "Micro Motion Transmitter" se asigna la dirección IP del microcontrolador y el puerto por el cual se comunica la computadora con la red de la empresa; 'para así enlazar el programa al microcontrolador y obtener los datos que se envían por ese medio.

En la figura 4.5 se observa el programa "Micro Motion Transmitter" donde a él se asigna la dirección IP del microcontrolador y el puerto.

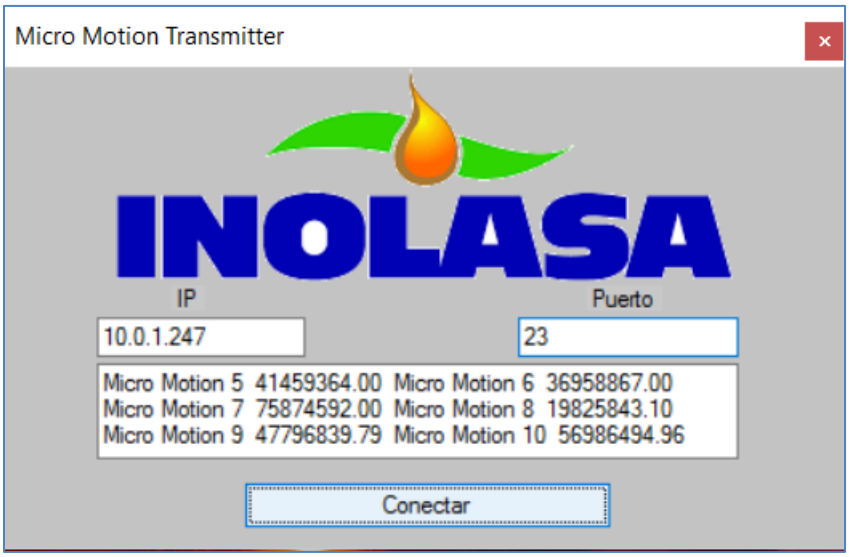


Figura 4.5 Programa realizado para la obtención de datos de la red local y almacenarlos en la base de datos.

En la figura 4.5 en la ventana de texto de la aplicación, se pueden visualizar los datos recibido de los medidores de flujo. Con ello, se verifica y valida el funcionamiento del recibimiento de los datos en la aplicación. Para verificar el almacenaje de la

información en la base de datos por medio de esta aplicación, se realiza la sección 4.5, donde se valida el funcionamiento de este.

4.5 Validación del almacenamiento en la base de datos

En la validación del almacenamiento en la base de datos de los datos medidos por los Micro Motion Transmitter, se ingresa al programa "Microsoft SQL Server Management Studio" y se realiza la verificación.

En la figura 4.6 se observan algunos de los datos almacenados el programa Microsoft SQL Server Management Studio en sus respectivas columnas, las cuales son: la columna "Flujómetro" para la identificación del Micro Motion Transmiter, la columna "Medición" que almacena el dato de la masa de cada medidor de flujo y por último la columna "Fecha" que almacena el día y la hora en que fue tomado almacenado el dato.

Se pueden observar en la columna de fecha mediciones realizadas a las 8:46 de la mañana y luego se registra otra toma de mediciones un minuto más tarde del mismo día, donde ellas registran un valor de masa superior a las primeras mediciones.

Flujometro	Medicion	Fecha
5	41459174	2016-10-21 08:46:24.000
6	36958695	2016-10-21 08:46:24.000
7	75874569	2016-10-21 08:46:24.000
8	19825687	2016-10-21 08:46:24.000
9	47796658	2016-10-21 08:46:24.000
10	56986313	2016-10-21 08:46:24.000
1	45857115	2016-10-21 08:47:44.000
2	25699124	2016-10-21 08:47:44.000
3	34589826	2016-10-21 08:47:44.000
4	71581598	2016-10-21 08:47:44.000
5	41459364	2016-10-21 08:47:44.000
6	36958867	2016-10-21 08:47:44.000
7	75874592	2016-10-21 08:47:44.000
8	19825843	2016-10-21 08:47:44.000
9	47796839	2016-10-21 08:47:44.000
10	56986494	2016-10-21 08:47:44.000

Figura 4.6 Datos almacenados en sus respectivas columnas en Microsoft SQL Server Management Studio.

Con esta verificación se valida el funcionamiento del almacenaje de los datos enviados desde el microcontrolador en la base de datos en el programa Microsoft SQL Server Management Studio.

En la sección 4.6 se verifica la exportación de los datos almacenados en Microsoft SQL Server Management Studio en un programa en Microsoft Excel y con ello poder solventar la necesidad de la empresa.

4.6 Validación de la visualización de los datos en las computadoras del departamento de producción de Inolasa

Para realizar la visualización de las mediciones almacenadas en la base de datos, se crea un archivo en Microsoft Excel; en él se realiza una conexión de datos externa de SQL server, para así poder exportar los datos desde Microsoft SQL Server Management Studio.

Una vez realizada la conexión con Microsoft SQL Server Management Studio, se realiza una opción para filtrar y poder exportar los datos por la fecha y hora de la medición, para así obtener únicamente los datos deseados. En la figura 4.7 se da un ejemplo donde se filtran dando únicamente dos registros de datos con un minuto de diferencia. También se observa que las mediciones finales son mayores que la primera.

Fecha inicial	10/21/2016 8:46	
Fecha Final	10/21/2016 8:47	
Flujometro	Medicion	Fecha
1	45856932	10/21/2016 8:46
2	25698965	10/21/2016 8:46
3	34589644	10/21/2016 8:46
4	71581458	10/21/2016 8:46
5	41459174	10/21/2016 8:46
6	36958695	10/21/2016 8:46
7	75874569	10/21/2016 8:46
8	19825687	10/21/2016 8:46
9	47796658	10/21/2016 8:46
10	56986313	10/21/2016 8:46
1	45857115	10/21/2016 8:47
2	25699124	10/21/2016 8:47
3	34589826	10/21/2016 8:47
4	71581598	10/21/2016 8:47
5	41459364	10/21/2016 8:47
6	36958867	10/21/2016 8:47
7	75874592	10/21/2016 8:47
8	19825843	10/21/2016 8:47
9	47796839	10/21/2016 8:47
10	56986494	10/21/2016 8:47

Figura 4.7 Programa en Microsoft Excel para la visualización de los datos almacenados en Microsoft SQL Server Management Studio.

Después de exportar los datos en Microsoft Excel de la base de datos, se realiza una tabla dinámica, conformada por los diez Micro Motion Transmitter, los respectivos valores medidos y la fecha de la medición. En esta tabla se puede filtrar los diferentes Micro Motion Transmitter y las fechas de la base de datos previamente exportada. En la figura 4.8 se observa la tabla dinámica creada para visualizar los datos tomados.

Suma de Medicion	Etiquetas de columna										
Etiquetas de fila	1	10	2	3	4	5	6	7	8	9	Total general
10/21/2016 8:46	45856932	56986313	25698965	34589644	71581458	41459174	36958695	75874569	19825687	47796658	456628095
10/21/2016 8:47	45857115	56986494	25699124	34589826	71581598	41459364	36958867	75874592	19825843	47796839	456629662

Figura 4.8 Tabla dinámica de los datos tomados de los medidores de flujo almacenados en la base de datos.

Finalmente, se realiza un gráfico dinámico con los datos obtenidos de la base de datos. En la figura 4.9 se muestra el gráfico, y se puede observar en el eje “y” el dato de la

masa medida por el Micro Motion Transmitter y en el eje “x” la fecha de la medición del dato.

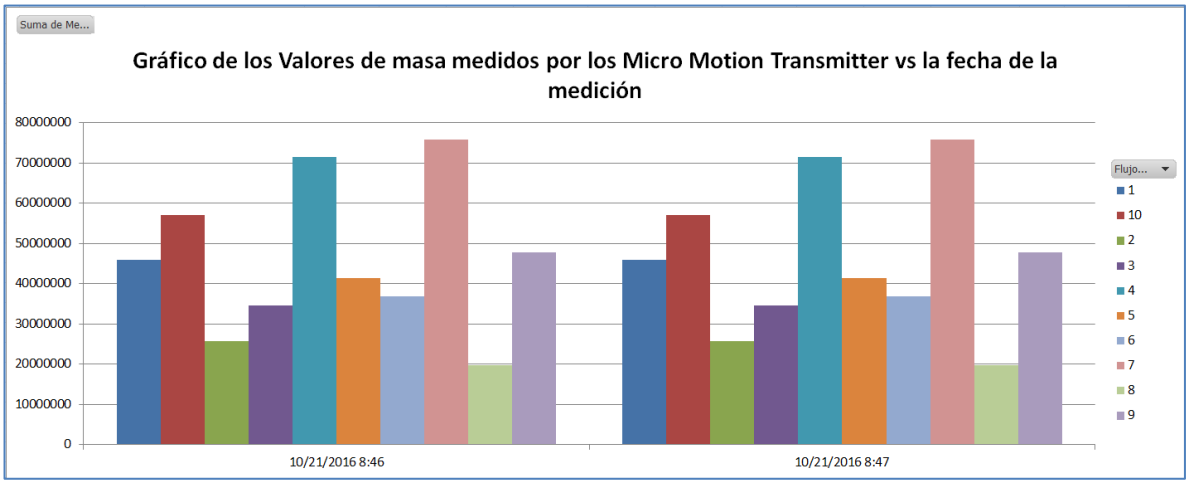


Figura 4.9 Grafico dinámico de los valores de masa medidos por los Micro Motion Transmitter vs la fecha de la medición.

De la misma manera que se pueden filtrar los datos de cada Micro Motion Transmitter y la fecha de medición en la tabla dinámica, se pueden filtrar también en el gráfico dinámico. En la figura 4.10 se muestra el gráfico dinámico mostrando únicamente el dato le Micro Motion Transmitter 6.

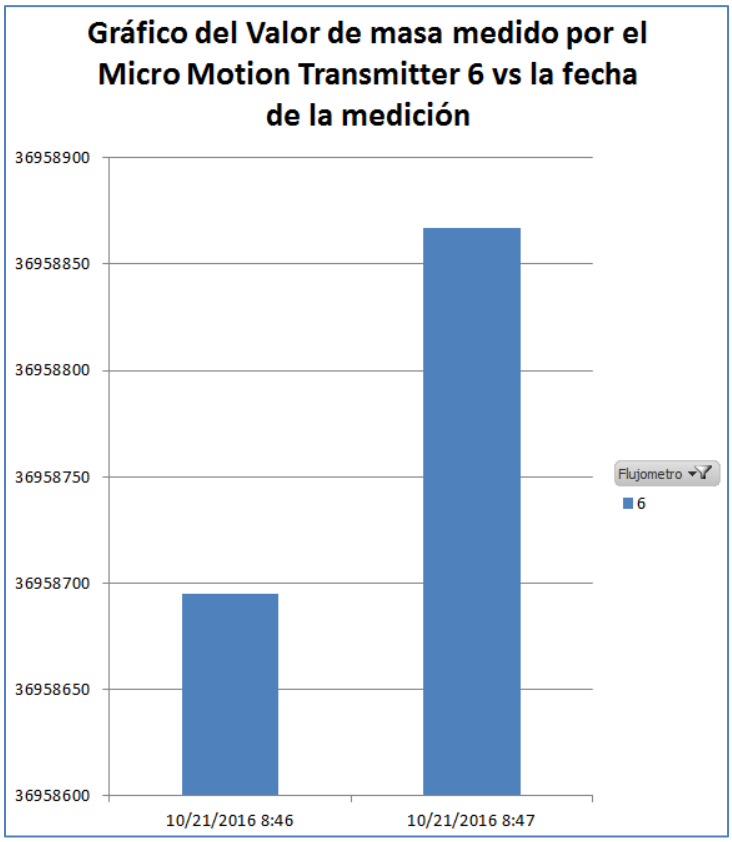


Figura 4.10 Gráfico dinámico del valor de masa medido por el Micro Motion Transmitter 6 vs la fecha de la medición.

Capítulo 5: Conclusiones

Se diseñó, implementó y validó el funcionamiento del sistema de comunicación y almacenaje de la información de las masas de aceite medidas por los Micro Motions Transmitter en la refinería de aceite de INOLASA. De esta manera, se elimina el problema de la empresa en cuanto a la adquisición de datos manualmente.

Se validó la comunicación Modbus RTU, donde el dato que se obtiene del Micro Motion Transmitter es decodificado en cuatro bytes en formato de "coma flotante", por lo que se debe realizar la conversión a un valor decimal.

Se validó la comunicación del flujo de señal de 4 a 20. Estos datos presentaron un error relativo menor al 2.5%, el cual se debe a una combinación del cálculo para obtener la masa, por medio del promedio del flujo y por la tolerancia de la carga en las salidas de 4 a 20 mA. Por otra parte, se ajusta la ecuación teórica para aumentar la precisión de medición.

Se diseñó, desarrolló y validó un programa para un microcontrolador Arduino, capaz de tomar los datos de los medidores de flujo de la planta de refinería de Inolasa, acondicionarlos para obtener el valor en kilogramos de la masa fluida por el sistema de la refinería y transferir estos a través de la red local de Inolasa.

Se diseñó, desarrolló y validó en un único programa, una aplicación en C Sharp (C#), con la cual se obtienen los datos enviados por el microcontrolador mediante la red de Inolasa y son almacenados en una base de datos creada en el servidor de la empresa.

Se realizó una plantilla en Microsoft Excel, con el cual se puede acceder a la base de datos del servidor, donde se almacenan los datos tomados de los medidores de flujo, desde cualquier ordenador de la empresa con acceso al servidor y así poder extraer los datos necesarios para sus respectivos estudios.

Capítulo 6: Recomendaciones

Se recomienda implementar un método de reinicio de los valores de las variables de los medidores de flujo 8, 9 y 10, que funcione para el microcontrolador utilizado en este proyecto y para los demás controladores programables utilizados en la planta. Esto debido a que cada señal para el reinicio de los valores de las variables, para cada controlador en la planta, es independiente, por lo que genera un error en la sincronización del dato.

Se recomienda implementar en un futuro la comunicación Modbus RTU con los Micro Motion Transmitter 8, 9 y 10 en este proyecto, si llegan a eliminar el uso de esa salida. Lo anterior, debido a que existe un bajo error relativo en obtener los datos de la masa de aceite por medio del flujo de aceite medido en las salidas de 4 a 20 mA del Micro Motion Transmitter.

Se recomienda, cada año, crear una nueva tabla dentro de la base de datos ya hecha, y asignar la dirección de la nueva tabla en el programa de C#, debido a que en 365 días la tabla va a estar conformada por 5256000 filas por registro de datos, por lo que esta se verá saturada de datos.

Bibliografía

[1] Axon Group (s, f). *Teoría en protocolo modbus* Recuperado de http://www.axongroup.com.co/protocolo_modbus.php

[2] Barbancho concejero, J., Benjumea mondéjar, J., Rivera romero, O., Romero ternero, M., Ropero rodríguez, J., Sánchez antón, G. et al (2014). *Redes locales* (2 ed.) Madrid: Paraninfo.

[3] Date, C. (1986). *Introducción a los SISTEMAS DE BASE DE DATOS* (7 ed.) México : Pearson Educación.

[4] EMERSON (2009). *Transmisores modelo 1700 de Micro Motion® con salidas analógicas. Suplemento al manual de configuración y uso* Manuscrito no publicado.

[5] EMERSON (2012). *Transmisor modelo 2700 de Micro Motion® con FOUNDATION™ fieldbus. Manual de configuración y uso* Manuscrito no publicado.

[6] EMERSON Process Management (2002). Introducción a Hart Recuperado de http://www2.emersonprocess.com/siteadmincenter/PM%20Central%20Web%20Documents/EngSch-Buses_201_es.pdf

[7] Gosselin, D. (2009, de Marzo). *ASP.NET Programming with C# and SQL SERVER* (1 ed.) Boston: Course Technology Cengage Learning.

[8] INOLASA (s. f.). Recuperado de http://www.inolasa.com/

[9] *Laboratorio de Automatización II. Modbus* (s, f) Recuperado de http://iaci.unq.edu.ar/materias/laboratorio2/transparencias%5Cmodbus.pdf

[10] Makofske, D., Donahoo, M. & Calvert, K. (s. f.). *TCP/IP SOCKETS IN C# Practical Guides for Programmers* (1 ed.) San Francisco: ELSEVIER

[11] Navarrete, A. (2013, 05 de Noviembre). Automatización de procesos en la empresa GestioPolis Recuperado de http://www.gestiopolis.com/automatizacion-de-procesos-en-la-empresa/

[12] S. Tamboli, M. Rawale, R. Thoraiet and S. Agashe, "Implementation of Modbus RTU and Modbus TCP communication using Siemens S7-1200 PLC for batch process," *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on*, Chennai, 2015, pp. 258-263.

[13] Tecnología Digital del Bajío. (2012, 11 de Setiembre) Modbus Parte II: Comunicación a través de una red 485 Recuperado de http://tecdigitaldelbajio.com/blog/25-modbus-parte-ii-comunicacion-a-traves-de-una-red-rs-485.html.

[14] Tecnología Digital del Bajío. (2012, 14 de agosto) *Modbus Parte III: ¿Qué es el Modbus?* Recuperado de http://www.tecdigitaldelbajio.com/blog/27-modbus-parte-iii-que-es-el-modbus.html

[15] Torrente artero, O. (2013). *ARDUINO Curso práctico de Formación* (1 ed.) Madrid: RC Libros.

[16] S. Wolf, R. Smith (1992). *Guía para mediciones electrónicas y prácticas de laboratorio*

Apéndices

A.1 Creación de la base de datos

Para crear la base de datos, se utiliza el programa de Microsoft SQL Server Management Studio, el cual es un software que se encuentra instalado en el servidor de Inolasa.

Como primer paso se abre al programa, se busca la carpeta “Databases” ubicada en la ventana de la izquierda llamada “Explorador de objetos”; en esa carpeta se presiona click izquierdo y se escoge la opción de “nueva base de datos”, como se observa en la A.1.1.

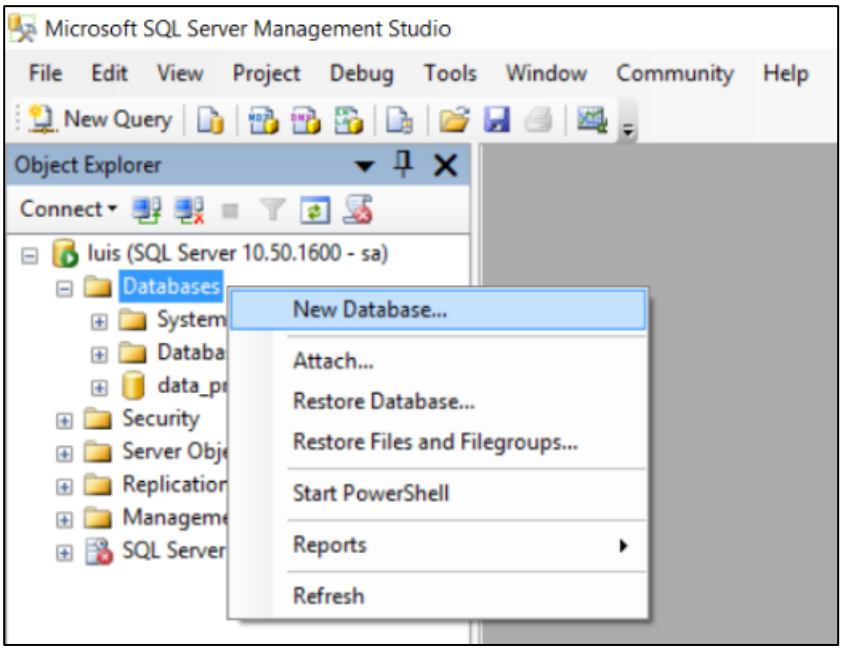


Figura A.1.1 Creación de Base de Datos en Microsoft SQL Server Management Studio.

Luego de escoger la opción de crear una nueva base de datos, aparece una pestaña, como se muestra en la figura A.1.2; en dicha pestaña se le asigna el nombre de la base de datos y se asigna el tamaño inicial de esta.

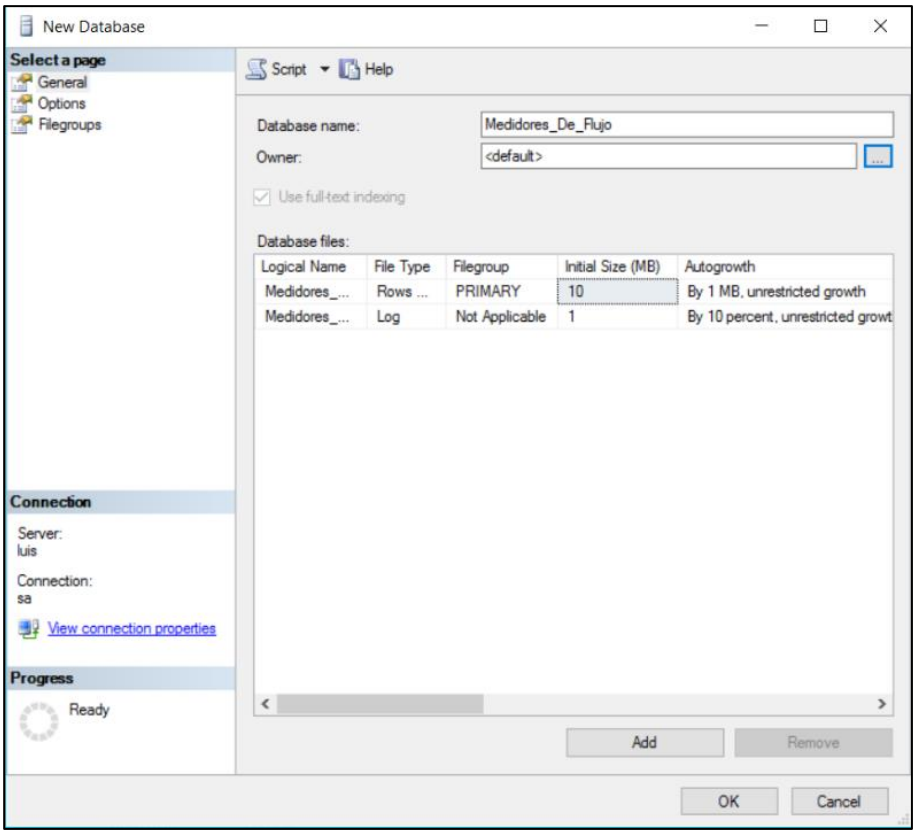


Figura A.1.2 Asignación del nombre de la Base de Datos creada en Microsoft SQL Server Management Studio y asignación de espacio inicial.

Con la base de datos creada, se regresa a la ventana “Explorador de objetos” y se encuentra la base de datos ya creada. Se busca la carpeta “Tablas”, ubicada en la base de datos y se presiona click izquierdo y se escoge la opción “Nueva Tabla”, como se observa en la figura A.1.3.

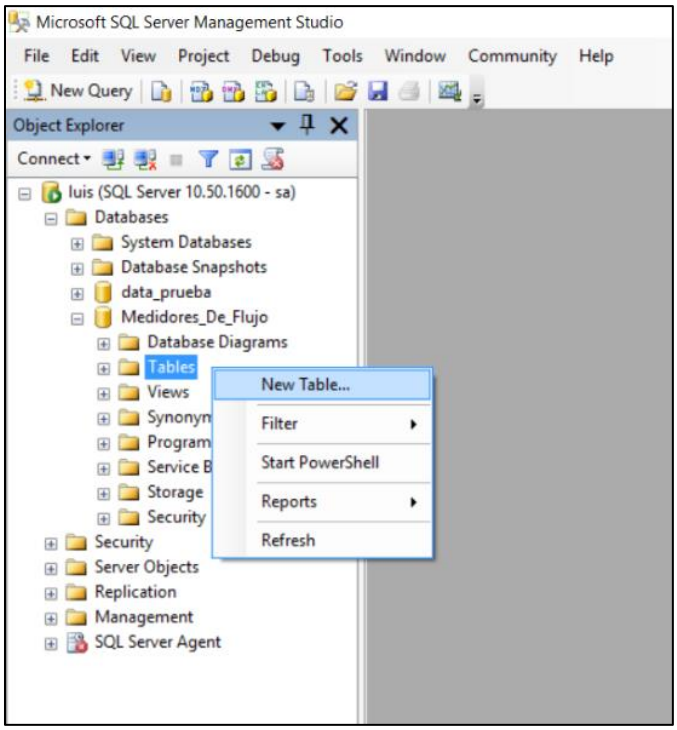


Figura A.1.3 Creación de la tabla en la Base de Datos hecha en Microsoft SQL Server Management Studio.

Después de crear una nueva tabla, se deben asignar las columnas, el tipo de dato y la opción de permitir datos nulos. En la figura A.1.4 se observa la asignación de una de las columnas, donde la primera lleva como nombre “Flujómetro” y el tipo de dato asignado es “varchar” de 50 caracteres. A la segunda columna se le asignó “Medición” y es de tipo entero. Por último, se encuentra la columna “Fecha” y es de tipo “datetime” para almacenar fecha y hora. En todas las columnas se acepta dato nulo.

Column Name	Data Type	Allow Nulls
Flujometro	varchar(50)	☑
Medicion	int	☑
Fecha	datetime	☑

Figura A.1.4 Asignación de las columnas, el tipo de dato y opción de permitir el dato nulo de la tabla.

Como se puede ver en la figura A.1.5, luego de crear la tabla, pueden visualizarse sus columnas y sus respectivos datos presionando la opción “Edit Top 200 Rows” en la tabla creada, ubicada en la ventana “Explorador de objetos”.

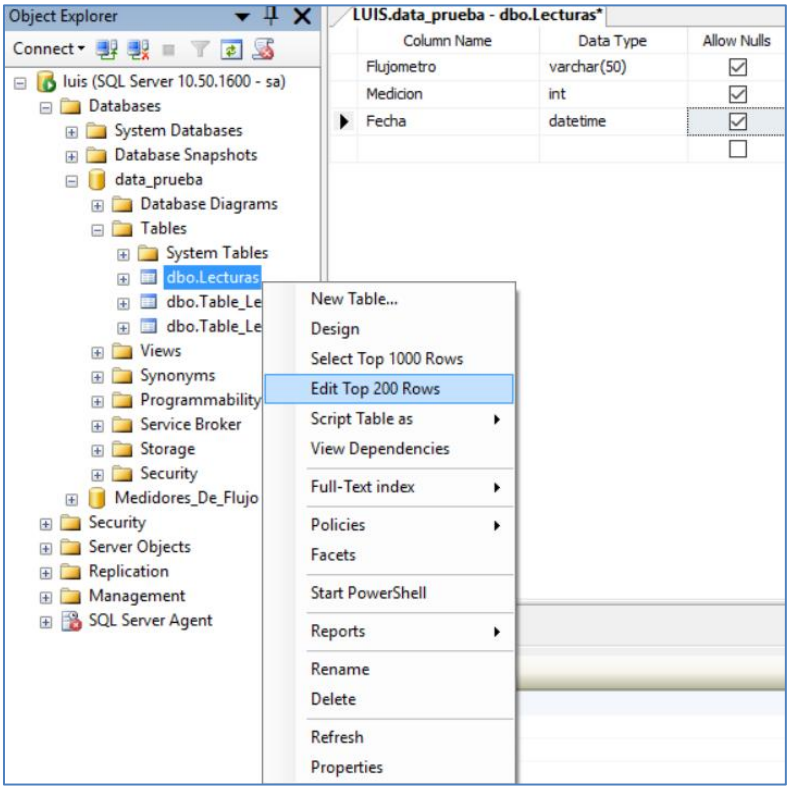


Figura A.1.5 Seleccionar la tabla en la Base de Datos hecha en Microsoft SQL Server Management Studio.

En la figura A.1.6 se visualiza la tabla creada con sus columnas respectivas y aún sin dato alguno.

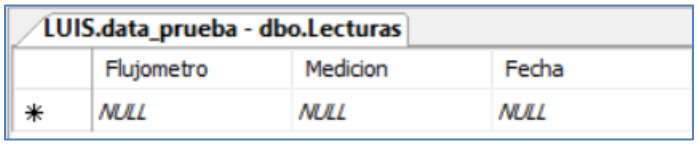


Figura A.1.6 Tabla realizada en la Base de Datos hecha en Microsoft SQL Server Management Studio.

A.2 Manual de Usuario

Introducción

Este manual le permitirá aprender a configurar la aplicación para el sistema de almacenamiento de la información de los medidores de flujo y de la misma manera y poder extraer los datos deseados.

Inicialización de la aplicación Micro Motion Transmitter

Se ejecuta la aplicación llamada Micro Motion Transmitter, la cual deberá abrir la ventana que se observa en la figura A.2.1. Debe seguir los siguientes pasos:

1. Colocar la dirección IP en el editor de texto 1, con la dirección 10.0.1.247, la cual es la dirección del microcontrolador instalado en la planta de refinería.
2. Asignar el puerto de la computadora donde se ejecute el programa, por el cual se conecta a la red. Si se encuentra conectado mediante cable Ethernet, debe utilizar el puerto 23. Si se encuentra conectada inalámbricamente, debe utilizar el puerto 80.
3. Presionar el botón “Conectar” de la aplicación.

De esta manera, se crea la conexión del microcontrolador hacia la base de datos y se encuentra almacenada la información en la base de datos.

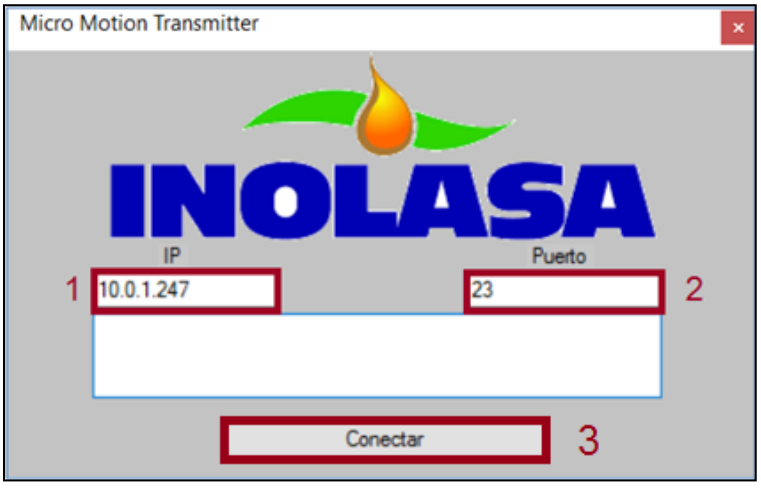


Figura A.2.1 Ventana de la aplicación “Micro Motion Transmitter”.

Visualización de los datos almacenados

Para visualizar la información almacenada en la base de datos, se abre la hoja de cálculo en Microsoft Excel, brindada con el sistema, la cual se observa en la figura A.2.2. Se deben seguir los siguientes pasos.

1. Colocar la fecha inicial registrada que se desea visualizar, de la misma manera como se observa en el cuadro rojo denotado con un “1” al lado derecho de la figura A.2.2.
2. Colocar la fecha final registrada que se desea visualizar, de la misma manera como se observa en el cuadro rojo denotado con un “2” al lado derecho de la figura A.2.2.
3. Presionar la opción “Actualizar Todo” en el programa de Microsoft Excel, para actualizar la tabla dinámica y la gráfica dinámica con los datos exportados.

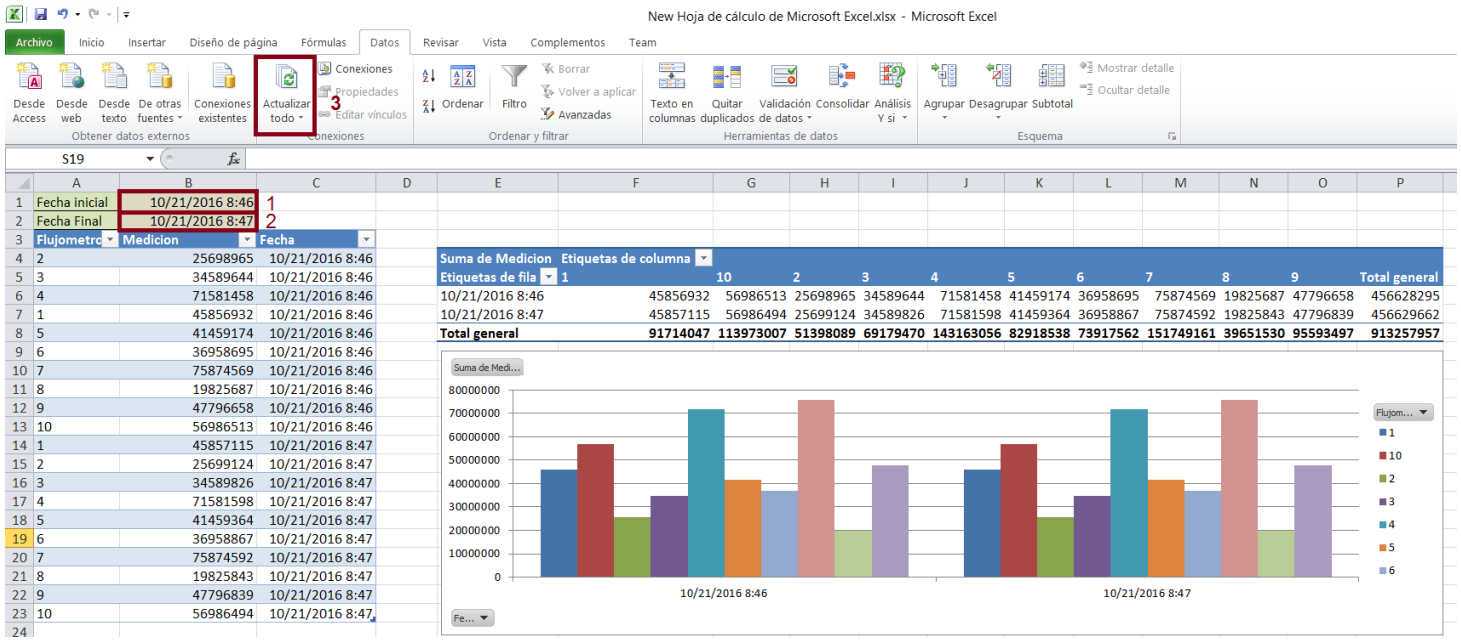


Figura A.2.2 Hoja de cálculo en Microsoft Excel para la visualización la información.

En caso que se desee cambiar de fechas, se deben repetir los pasos del 1 al 3 de este apartado.

A.3 Código para programa ejecutable realizado en C#

En el presente apéndice, se muestra el código principal, utilizado para realizar el programa, usando el lenguaje C#.

En la figura A.3.1 se observa el código de la ventana principal. En ella se visualizan tres funciones principales, las cuales son para recibir información mediante la red local, enviar información y la función de realizar la conexión TCP/IP cuando se presiona el botón de la aplicación.

```
public partial class Form1 : Form
{
    private TcpClient client;
    public StreamReader STR;
    public StreamWriter STW;
    public string MM1, MM2, MM3, MM4, MM5, MM6, MM7, MM8, MM9, MM10, MM11, sender_mm;
    public int sender_num =0;
    public String text_to_send;

    public Form1()
    {
        InitializeComponent();

        IPAddress[] localIP = Dns.GetHostAddresses(Dns.GetHostName());
        foreach (IPAddress address in localIP)
        {
        }
    }

    private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e) ...

    private void backgroundWorker2_DoWork(object sender, DoWorkEventArgs e) ...

    private void button3_Click(object sender, EventArgs e)...
}
```

Figura A.3.1 Formulario de la aplicación en Microsoft Visual Studio 2010.

En la figura A.3.1, se visualiza el código de la función accionada por el botón de la aplicación. En ella se observa el código para realizar la comunicación TCP/IP.

```
private void button3_Click(object sender, EventArgs e)
{
    client = new TcpClient();
    IPEndPoint IP_End = new IPEndPoint(IPAddress.Parse(textBox5.Text), int.Parse(textBox6.Text));
    try
    {
        client.Connect(IP_End);
        if (client.Connected)
        {
            STW = new StreamWriter(client.GetStream());
            STR = new StreamReader(client.GetStream());
            STW.AutoFlush = true;
```

Figura A.3.2 Código de la función del accionamiento del botón, obtenido y acondicionado de [10].

En la figura A.3.3 se observa el código utilizado para el envío de datos en la red local.

```
STW.WriteLine(text_to_send);
```

Figura A.3.3 Código para función de envío de datos en la red local.

En la figura A.3.4 se observa el código utilizado para el llamado del objeto Micromotion1", la asignación de valores de cada medidor de flujo y el llamado de la clase "MicromotionDAL".

```
Micromotion Micromotion1 = new Micromotion();
Micromotion1.mmotion = NM;
Micromotion1.flujo = numf;
Micromotion1.fecha = fecha;
int resultado = MicromotionDAL.Agregar(Micromotion1);
```

Figura A.3.4 Llamado del objeto "Micromotion1", asignación de valores de cada medidor de flujo y llamado de la clase "MicromotionDAL".

En la figura A.3.5 se observa el código de la clase MicromotionDAL, la cual es pública; esta se encarga de llenar los parámetros de forma correcta en la base de datos y llamar la clase BDComun.

```
public static int Agregar(Micromotion pMicromotion)
{
    int retorno = 0;
    using (SqlConnection Conn = BDComun.ObtenernerConexion())
    {
        SqlCommand Comando = new SqlCommand(string.Format("Insert Into Lecturas (Flujometro,Medicion,Fecha) values ('{0}' , '{1}' ,'{2}')",
            pMicromotion.mmotion, pMicromotion.flujo, pMicromotion.fecha), Conn);
        retorno = Comando.ExecuteNonQuery();
    }

    return retorno;
}
```

Figura A.3.5 Clase MicromotionDAL programada en Microsoft Visual Studio 2010.

En la figura A.3.6 se observa el código de la clase BDComun, la cual realiza la conexión con la base de datos donde se va a almacenar la información.

```
public class BDComun
{

    public static SqlConnection ObtenernerConexion()
    {
        SqlConnection Conn = new SqlConnection("Data source=luis; Initial Catalog=data_prueba; User Id=sa; Password=12345");
        Conn.Open();
        return Conn;
    }
}
```

Figura A.3.6 Clase BDComun programada en Microsoft Visual Studio 2010.