

**DISEÑO DEL PROTOTIPO DE UN SISTEMA DE ADQUISICION DE DATOS
PARA UNA ESTACIÓN DE MUD LOGGING**

LUIS ALVARO AYALA ARTEAGA

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2018**

**DISEÑO DEL PROTOTIPO DE UN SISTEMA DE ADQUISICION DE DATOS
PARA UNA ESTACIÓN DE MUD LOGGING**

LUIS ALVARO AYALA ARTEAGA

**Trabajo de grado presentado como requisito para optar al título de Ingeniero
Electrónico**

**Director
Esp. Humberto Rolando Barahona Cabrera
Ingeniero Electrónico**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
SAN JUAN DE PASTO
2018**

“Las ideas y conclusiones aportadas en el siguiente trabajo son responsabilidad exclusiva del autor”.

Artículo 1ro del Acuerdo No. 324 de octubre 11 de 1966 emanado del Honorable Consejo Directivo de la Universidad de Nariño.

Nota de Aceptación

Presidente

Jurado

Jurado

San Juan de Pasto, 25 de Mayo de 2018

RESUMEN

El Mud Logging es el procedimiento por el cual se realiza el registro y monitoreo en tiempo real de algunas variables físicas, fundamentales durante el proceso de perforación de un pozo de hidrocarburos, una estación de Mud Logging monitorea durante este proceso, entre otros, una serie de parámetros de perforación tales como: nivel y flujo de lodo, presión de las bombas, peso del gancho y torque del taladro de perforación.

La información obtenida durante la perforación es de suma importancia para hacer seguimiento y tomar decisiones sobre el proceso tales como: crear puntos de revestimiento, continuar o detener la perforación, cambiar una broca por avería, etc.

En este trabajo se presenta el diseño de un prototipo de instrumento para la adquisición de datos en una estación de Mud Logging, el prototipo se ha diseñado para ser implementado a un bajo costo, teniendo la característica de monitorear las variables de torque del taladro, flujo de lodo y nivel de tanque de lodo, siendo capaz de monitorear estas variables en tiempo real y también de guardar los registros de la perforación para imprimir reportes, lo anterior haciendo uso de un software de aplicación desarrollado para controlar el instrumento e instalado en un computador portátil.

ABSTRACT

Mud Logging is the process in which real time physics parameters register and monitoring procedure are made on a drilling well, during this process important drilling parameters are monitored, such as: mud tank level, mud flow, pump pressure, hook load and rotary torque.

The information obtained during the mud logging process is quite important to make process tracing and -on process decisions- such as: make coating tube points, start or stop drilling, change damaged drill, etc.

In this work, the design of a data acquisition system prototype instrument for a mud logging station is presented, this prototype was designed to be a low cost implemented instrument and to be able to real time monitor the rotary torque, mud flow and mud tank level parameters and also keep logs for further report prints, all this features by means of an application software installed on a laptop and developed for the instrument control.

CONTENIDO

1. DESCRIPCION DEL PROBLEMA	16
1.1 PLANTEAMIENTO DEL PROBLEMA	16
1.2 FORMULACIÓN DEL PROBLEMA	16
2. JUSTIFICACION	17
3. OBJETIVOS	18
3.1 OBJETIVO GENERAL	18
3.2 OBJETIVOS ESPECÍFICOS	18
4. MARCO CONCEPTUAL	19
4.1 EXPLORACION Y EXPLOTACION DE HIDROCARBUROS	19
4.1.1 Proceso de Exploración.	19
4.1.2 Proceso de Perforación.	20
4.1.3 Proceso de Extracción.	20
4.2 MATERIALES Y HERRAMIENTAS DE PERFORACIÓN	21
4.2.1 Plataforma o Torre de Perforación.	21
4.2.2 Broca o Barrena de perforación.	23
4.2.3 Fluido o Lodo de Perforación (Mud)	24
4.2.4 Definición de Mud Logging.	26
4.3 VARIABLES QUE INTERVIENEN EN EL PROCESO DE MUD LOGGING	29
4.3.1 Profundidad y Rata de Penetración (ROP).	29
4.3.2. Carga del Gancho y Peso Sobre la Broca (Hookload y WOB).	31
4.3.3 Velocidad de Rotación y Torque de la Rotaria.	32
4.3.4 Rata de Flujo y Nivel de los Tanques.	34
4.4 MONITOREO DE VARIABLES Y SISTEMA DE ADQUISICION DE DATOS	36
5. METODOLOGIA	37
5.1 VARIABLES FISICAS DEL PROCESO Y SENSORES INDUSTRIALES	37
5.1.1 Variables a Monitorear Durante el Proceso de Perforación.	37
5.1.2 Sensor de flujo de lodo MD Totco Mud Flow Sensor 270700-Series:	38
5.1.3 Sensor de Nivel del tanque <i>VEGAPULS 65 radar level sensor</i> .	42
5.1.4 Sensor de torque MD Totco Rotary Torque Sensor 40223503 Series.	44

5.2 ADQUISICIÓN Y ACONDICIONAMIENTO DE SEÑALES	45
5.2.1 Lazo de Corriente 4-20 mA (Current Loop).	45
5.2.2 Transmisor de Corriente de 4-20 mA.	47
5.2.3 Receptor de Corriente de 4-20 mA.	49
5.2.5 Etapa de Adecuación de Señal.	50
5.2.6 Tratamiento Del Ruido.	53
5.3 PROCESAMIENTO Y ENVIO DE LA INFORMACIÓN:	54
5.3.1 Arduino Leonardo Board.	54
5.4 RECEPCIÓN Y MONITOREO DE LA INFORMACION EN EL PC.	56
5.4.1 Transmisión serial en C Sharp (C#).	56
5.4.2 Diseño del Software de Aplicación LoggingDAQ	58
6. RESULTADOS	60
6.1 TARJETA DE ADQUISICIÓN DE DATOS DAQ.	60
6.1.1 Diagrama Esquemático del Circuito Electrónico.	60
6.1.2 Diagrama de Circuito Impreso PCB.	61
6.1.3 Montaje del Circuito.	62
6.2 SOFTWARE DE APLICACIÓN 'LoggingDAQ'.	65
6.2.1 Módulo de Inicio.	65
6.2.2 Módulo de Configuración.	66
6.2.3 Módulo de Monitoreo.	67
6.2.4 Módulo de Generación de Reportes.	71
CONCLUSIONES	72
RECOMENDACIONES	73
BIBLIOGRAFÍA	74
ANEXOS	75

LISTA DE TABLAS

Tabla 1. Elementos de una torre de perforación.

Tabla 2. Índice de partes del sensor de flujo.

Tabla 3. Puntos de verificación del sensor de flujo.

Tabla 4. Tipos de señales ANSI/ISA50.1- 1982(R1992).

Tabla 5. Especificaciones Técnicas del Arduino Leonardo Board.

LISTA DE FIGURAS

Figura 1, Torre de perforación realizando un pozo para extracción de crudo.

Figura 2, Diagrama simplificado de una torre de perforación.

Figura 3, Diseños de barrenas de perforación.

Figura 4, Tipos de fluidos de perforación.

Figura 5, Esquema funcional de una estación de Mud Logging

Figura 6, Rata de penetración vs Velocidad de Rotaria

Figura 7, cambio en ROP causado por un Drilling Breake.

Figura 8, Relación de torque (ft-lbs x amperio).

Figura 9, Sensores de nivel de tanque: Ultrasónico y de Varilla

Figura 10, Sensor de Flujo eléctrico MD Totco Mud Flow Sensor 270700-Series.

Figura 11, Ajuste del sensor eléctrico de flujo. 41

Figura 12, Diagrama esquemático del componente eléctrico. 42

Figura 13, MD Totco Mud Flow Sensor 270700-Series. 42

Figura 14, VEGAPULS 65 radar sensor. 43

Figura 15, Rango de medición de la sonda.

Figura 16, Diagrama del sensor de torque.

Figura 17, MD Totco Rotary Torque Sensor 40223503 Series.

Figura 18, Topología común de un Lazo de Corriente 4-20mA.

Figura 19, Circuito Transmisor de 4-20 mA a partir de 0-5 vdc.

Figura 20, Diagrama Esquemático Transmisor 4-20mA.

Figura 21, Diagrama PCB Transmisor 4-20 mA.

Figura 22, Transmisor 4-20mA ensamblado.

Figura 23, Diagrama esquemático del conversor 4-20mA.

Figura 24, Diagrama Esquemático de la Etapa de Adecuación de Señal.

Figura 25, Barrera Aislada KFD2-CD-Ex1.32.

- Figura 26**, Arduino Leonardo Board.
- Figura 27**, Instancia Principal de Visual C# Express 2008.
- Figura 28**, Componente SerialPort en Visual C#.
- Figura 29**, Pantalla Principal del Software LoggingDAQ.
- Figura 30**, Diagrama Esquemático de la Tarjeta de Adquisición de datos DAQ.
- Figura 31**, Diagrama del circuito PCB.
- Figura 32**, Placa de Circuito Impreso Revelada.
- Figura 33**, Placa de Circuito Impreso Perforada.
- Figura 34**, Tarjeta ensamblada y montada sobre la carcasa.
- Figura 35**, Ensamble final del DAQ.
- Figura 36**, Módulo de inicio del software LoggingDAQ.
- Figura 37**, Módulo de configuración del software LoggingDAQ.
- Figura 38**, Pantalla principal del software LoggingDAQ.
- Figura 39**, Pantalla principal en estado: 'Conectado'.
- Figura 40**, Menú Vista.
- Figura 41**, Modo de visualización numérico.
- Figura 42**, Modo de visualización gráfico en tiempo real.
- Figura 43**, Notificación de fin de sesión de monitoreo actual.
- Figura 44**, Menú generar log para impresión.
- Figura 45**, Reporte generado por el aplicativo en MS Excel.

LISTA DE ANEXOS

Anexo 1. Descripción de pines y diagrama esquemático de la placa Arduino Leonardo.

Anexo 2. Diagrama esquemático de la placa Arduino Leonardo.

Anexo 3. Código del programa desarrollado en C++ para el Arduino Leonardo.

Anexo 4. Código del software de aplicación desarrollado en lenguaje C#.

GLOSARIO

API: la interfaz de programación de aplicaciones, abreviada como API del inglés: Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

AWG: Calibre de alambre estadounidense (en inglés american wire gauge o AWG) es una referencia de clasificación de diámetros. En muchos sitios de Internet y también en libros y manuales, especialmente de origen norteamericano, es común encontrar la medida de conductores eléctricos (cables) indicados con la referencia AWG. Cuanto más alto es este número, más delgado es el alambre. El alambre de mayor grosor (AWG más bajo) posee menos resistencia interna y, por lo tanto, soporta mayores corrientes a distancias más grandes.

BARRENA DE PERFORACION: Es la herramienta utilizada para triturar o cortar la roca. Todo lo que se encuentra en un equipo de perforación asiste directa o indirectamente a la barrena para la trituración o el corte de la roca. La barrena se encuentra en la parte inferior de la sarta de perforación y debe cambiarse cuando se desgasta excesivamente y deja de avanzar.

BENTONITA: La bentonita es una arcilla de grano muy fino (coloidal) del tipo de montmorillonita que contiene bases y hierro. Tiene aplicaciones en cerámica, entre otros usos. El nombre deriva de un yacimiento que se encuentra en Fort Benton, Estados Unidos.

BOOTLOADER: Un gestor de arranque o arrancador (en inglés «bootloader») es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita para iniciar el sistema operativo. Normalmente se utilizan los cargadores de arranque multietapas, en los que varios programas pequeños se suman los unos a los otros, hasta que el último de ellos carga el sistema operativo.

CRUDO: El petróleo crudo es una mezcla de compuestos orgánicos, principalmente hidrocarburos insolubles en agua, es conocido como petróleo crudo o simplemente crudo. También se le conoce como oro negro.

ESTÁNDAR ECMA: Es una especificación de lenguaje de programación publicada por ECMA International. El desarrollo empezó en 1996 y estuvo basado en el popular lenguaje JavaScript propuesto como estándar por Netscape Communications Corporation. Actualmente está aceptado como el estándar ISO 16262.

HMI: La interfaz de usuario-máquina, es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo.

IDE: Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

LITOLOGIA: Es la parte de la geología que estudia las características de las rocas que aparecen constituyendo una determinada formación geológica.

OLEODUCTO: Se denomina oleoducto a la tubería e instalaciones conexas utilizadas para el transporte de petróleo, sus derivados y biobutanol, a grandes distancias.

SARTA DE PERFORACIÓN: Es la combinación de la columna de perforación, el arreglo de fondo de pozo y cualquier otra herramienta utilizada para que la barrena gire en el fondo del pozo.

SURGENCIA: Las surgencias son un fenómeno oceanográfico que consiste en el movimiento vertical de las masas de agua, de niveles profundos hacia la superficie.

INTRODUCCIÓN

En la actualidad es evidente como en la mayoría de procesos industriales, la labor manual ha sido en gran parte reemplazada por sistemas de control automático, lo cual con la ayuda de los sistemas de adquisición de datos por computador, ha permitido obtener mejores resultados en cuanto a la calidad de información obtenida y la optimización de los procedimientos realizados.

El control geológico de un pozo durante la perforación y explotación de hidrocarburos, es un proceso industrial que se realiza en un ambiente rústico y en términos generales de alto riesgo para la integridad de los operarios, requiere la utilización de sistemas de instrumentación automatizados, los cuales reducen los costos, mejoran la eficiencia y alcanzan mayores grados de precisión, incluso cuando se perforan pozos con trayectorias complicadas en ambientes desafiantes.

El proceso de adquisición de datos realizado durante la perforación, juega un papel clave dentro del procedimiento de explotación de hidrocarburos, consiste en el monitoreo, almacenamiento y registro de los datos, tanto geológicos como ingenieriles que se obtienen durante el proceso, brindando la posibilidad de evidenciar cualquier cambio importante en el transcurso de la perforación, para que las empresas dedicadas al Mud Logging puedan realizar este proceso, necesitan una infraestructura base de alto costo que consta de sensores industriales, equipos y software diseñados para realizar el monitoreo de parámetros de perforación, análisis de suelo, registros y control geológico.

Es precisamente en éste campo de la instrumentación donde este proyecto hace énfasis ya que se propone desarrollar el prototipo de un sistema de adquisición y monitoreo de datos, a un bajo costo, este instrumento es conocido en el ámbito como DAQ, el prototipo de instrumento que se propone, pretende cumplir con los requerimientos básicos establecidos para implementar un sistema electrónico de instrumentación en una estación de Mud Logging, el cual debe permitir monitorear en tiempo real los datos de perforación establecidos, procesar la información y registrar los eventos determinados para monitorear el proceso.

1. DESCRIPCION DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Ante el alto costo que conlleva la necesidad de las estaciones de *Mud Logging* de disponer de una infraestructura stock propia, cuyas características ayuden a cumplir con los requerimientos del contratante final, se hace evidente la necesidad de diseñar y fabricar un equipo de adquisición y monitoreo de datos DAQ, con propósito y funciones similares a los ofrecidos comercialmente, a un menor costo de adquisición, pero a su vez diseñado con unas características de operación dedicadas, que permitan la inmediata adecuación del instrumento dentro del proceso en contexto, diferenciándolo así de los equipos de adquisición comerciales, los cuales no brindan una interfaz ni un diseño personalizado dentro de las necesidades inherentes a esta actividad, el planteamiento del problema se enfoca entonces en lograr realizar el diseño de un prototipo de instrumento, con las características mencionadas anteriormente pero con un costo mucho menor en su implementación.

Al reducir el costo del instrumento conocido como DAQ, se aporta a la necesidad de implementar una infraestructura stock que cumpla con los mismos requerimientos que entrega una infraestructura comercial actual, pero a un menor costo de implementación y con características de operación caracterizadas para el proceso de perforación, lo cual reditúa en beneficios económicos para la empresa y eventualmente genera mayores utilidades al realizar una menor inversión económica.

1.2 FORMULACIÓN DEL PROBLEMA

¿Es posible desarrollar un prototipo de instrumento de adquisición de datos (DAQ) y un software de control, con un bajo costo de implementación y con características dedicadas para el proceso de adquisición y monitoreo de los parámetros requeridos durante el proceso de perforación en una estación de Mud Logging?

2. JUSTIFICACION

La investigación y el trabajo de campo correspondientes al presente proyecto corresponden a la experiencia realizada en los terrenos petroleros aledaños al municipio El Coca, provincia de Orellana, en la república del Ecuador, durante el desarrollo de labores como Auxiliar Técnico de Mud-Logging (logger), de ésta manera ha sido posible experimentar las necesidades que tienen éste tipo de empresas que se dedican a ésta labor que es fundamental en el proceso de exploración de hidrocarburos y se ha logrado hacer un análisis de las necesidades que se presentan en los sistemas de exploración a nivel técnico instrumental.

Para realizar los procesos de logging, se utilizan sensores industriales estándar de un costo muy elevado, los cuales son generalmente propiedad de las empresas contratantes dueñas de los pozos, ya que éstos sensores hacen parte de la estructura física base que se instala para realizar la perforación, en consecuencia, se hace evidente la necesidad de desarrollar un prototipo de sistema de adquisición de datos, cuyo alcance está enfocado principalmente en desarrollar un instrumento básico de monitoreo a un menor costo y caracterizado para el proceso de Mud Logging, a diferencia de los sistemas de adquisición comercial estándar.

El soporte al proceso se dificulta cuando se presentan inconvenientes con los instrumentos, ya que los equipos son generalmente importados y no tienen ningún tipo de soporte inmediato, entonces el encargado tiene que resolver por su propia cuenta cualquier inconveniente, teniendo en cuenta lo anterior se pretende desarrollar un instrumento que presente funciones caracterizadas con el contexto del proceso, lo cual permite brindar un soporte técnico adecuado, de esta forma se aporta una solución a la necesidad de implementar una infraestructura stock que cumpla con los mismos requerimientos que entrega una infraestructura comercial, pero a un menor costo de implementación y con características de operación dedicadas para el proceso de perforación. Al reducir el costo de la inversión en infraestructura se generan beneficios económicos para la empresa y eventualmente se generan mayores utilidades.

Siendo el anterior postulado la justificación principal de este proyecto, existe una justificación a un nivel más particular, también se pretende dar a conocer ésta área de competencia y fomentar el interés por parte de profesionales de la región para participar en este sector brindando este tipo de servicios, aunque en el departamento de Nariño no existe mucha demanda con respecto a este tipo de servicio, si existe demanda en el interior del país y departamentos vecinos como el Putumayo y en la zona fronteriza con el Ecuador donde existen gran cantidad de pozos petroleros, actualmente son empresas de otros países y muy pocas del interior de nuestro país las que abarcan todo este mercado tanto en los pozos de Colombia como en el Ecuador.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un prototipo de instrumento de adquisición de datos (DAQ) y un software de control, con un bajo costo de implementación y con características dedicadas al proceso de adquisición y monitoreo de los parámetros requeridos durante el proceso de perforación en una estación de Mud Logging.

3.2 OBJETIVOS ESPECÍFICOS

- Diseñar una etapa de acondicionamiento de señal, para adecuar las señales que brindan los tres sensores industriales predefinidos de Torque, Flujo y Nivel y entregar valores adecuados para su posterior procesamiento.
- Desarrollar un firmware para una tarjeta arduino basada en microcontrolador Atmel, para permitir mediante la utilización de este hardware, la reducción de costos de implementación planteada en el objetivo general. En el diseño del programa se deben contemplar las etapas de recepción, conversión análoga-digital A/D, procesamiento de datos y transmisión de la información hacia el PC y las funciones que se definan para caracterizar el proceso de Mud Logging.
- Desarrollar y programar un software de control y visualización de datos (Software de Aplicación), como interfaz de usuario para realizar el monitoreo, registro de información y la comunicación bidireccional con el instrumento desde un computador portátil.
- Diseñar y elaborar un prototipo final de tarjeta de adquisición de datos, basada en el ensamble de las etapas desarrolladas para cada proceso, verificando que cumpla con el propósito y las funciones especificadas en el diseño general.

4. MARCO CONCEPTUAL

4.1 EXPLORACION Y EXPLOTACION DE HIDROCARBUROS

El petróleo crudo es sin duda el recurso natural más importante de los países industrializados y es fácil entender porque muchos de los elementos y productos que utilizamos hoy en día, se derivan de este recurso no renovable también conocido como “oro negro”.

En primer lugar, el crudo es una fuente de energía, gracias a la cual podemos utilizar nuestros vehículos y otros medios de transporte, pues tanto la gasolina como los lubricantes que se utilizan en la industria automotriz, se derivan de él.

El petróleo es una sustancia orgánica compuesta básicamente de hidrocarburos extraídos desde el interior de la tierra, hidrocarburos que se obtienen a partir de la fosilización de restos orgánicos, como por ejemplo los de los dinosaurios.

Para realizar la extracción de los hidrocarburos del sub-suelo y convertirlos en los valiosos derivados, los cuales son la materia prima de la mayoría de elementos que utilizamos cotidianamente, se requieren 3 procesos: exploración, perforación y extracción o producción.

4.1.1 Proceso de Exploración.

La exploración es el estudio de suelos que se realiza para identificar posibles yacimientos, se toman muestras de roca y suelo en los que se describen las características de la superficie. Imágenes satelitales y fotografías aéreas, de ser ubicadas las zonas de un posible yacimiento, se da inicio a una segunda fase llamada sísmica (marina o terrestre), la cual consiste en un estudio litológico y geológico para determinar las zonas aptas para encontrar hidrocarburos dando paso a la perforación exploratoria.

Cuando los informes geológicos indican que existe la posibilidad de encontrar una cantidad significativa de hidrocarburo a una profundidad determinada es momento de proceder con el proceso de perforación, el primer pozo que se perfora en una área geológicamente inexplorada y que permite saber si realmente hay o no hidrocarburos, se denomina pozo exploratorio¹.

¹ WSC WELL CONTROL SCHOOL, Drilling Well Control Course, Harvey Louisiana: 70058 Aberdeen Drilling School, 2002, p.11-12.

4.1.2 Proceso de Perforación.

Diferentes tipos de roca exigen diferentes tipos de perforación, cuanto más rápido se perfora menos costará el procedimiento por lo tanto es importante tomar las decisiones adecuadas antes de iniciar la perforación, para realizar el proceso previamente se fundamentan sobre el informe geológico que se ha realizado al pozo que será explotado, con lo cual se hace una idea clara de la estructura de la roca que se va a perforar y así determinar el tipo de broca adecuada para el caso correspondiente, las brocas son generalmente fabricadas con carburo de tungsteno el cual es 4 veces más fuerte que el acero y se funde a una temperatura de 2870 grados centígrados, para complementar su resistencia, se le colocan unos dientes de diamante el cuál es el material más resistente que se conoce hasta el momento.

Para alcanzar la profundidad a la cual se encuentra el hidrocarburo se debe proceder básicamente a instalar una torre de perforación compuesta por una serie de herramientas, tales como motores que imprimen la fuerza que requiere el proceso de perforación, un tipo especial de tubería de revestimiento la cual se cementa a las paredes del pozo en las diferentes etapas de perforación previniendo derrumbes y posibles contaminaciones de acuíferos y un sistema de prevención de la salida de fluidos provenientes del subsuelo.

Todo el proceso se basa en una torre de perforación que contiene todo el equipamiento necesario para bombear el fluido de perforación, bajar y elevar la línea, controlar las presiones bajo tierra, separar las rocas del fluido que retorna, y generar la energía necesaria eléctrica y mecánica para la operación, generalmente mediante grandes motores diésel².

4.1.3 Proceso de Extracción.

Al finalizar la perforación se realiza el completamiento y prueba de los pozos, si el volumen encontrado justifica su extracción, se inicia la producción de crudo y gas natural, mediante un instrumento llamado cañón, se perfora el último tramo de tubería abriendo pequeños orificios por dónde empezará a filtrarse el hidrocarburo saliendo a la superficie de forma controlada, de ser necesario se instalan sistemas para bombear el crudo por insuficiencia natural de presión.

En superficie se utilizan diferentes equipos para limpiar las impurezas, separando sedimentos agua y gas natural y finalmente se coloca el crudo en tanques de almacenamiento para iniciar su transporte hacia las refinerías generalmente

² INTERNATIONAL LOGGING, INC. Basic Mud Logging Manual, Versión 1.0. p.29-34.

haciendo uso de oleoductos, se denomina oleoducto a la tubería e instalaciones conexas utilizadas para el transporte de petróleo y sus derivados a grandes distancias.

Los oleoductos son la manera más rápida de transportar grandes cantidades de petróleo, se fabrican de tubos de acero o plástico, donde sea posible, se construyen sobre la superficie, sin embargo, en áreas que sean más desarrolladas, urbanas o con flora sensible, se entierran a una profundidad típica de 1 metro.

4.2 MATERIALES Y HERRAMIENTAS DE PERFORACIÓN

4.2.1 Plataforma o Torre de Perforación.

La torre de perforación está compuesta por un equipo que consta principalmente de una broca o barrena que se fija a una serie de tuberías de acero que se van uniendo a medida que se avanza en la profundidad conformando la sarta de perforación, a su vez las tuberías van enganchadas a una barra cuadrada llamada kelly, la cual atraviesa un disco de acero ubicado en el piso de la torre, un motor hace girar la mesa rotatoria a medida que la broca avanza perforando el suelo, la broca de perforación o barrena es empujada por el peso del kelly y la tubería sobre ella, las cuales proporcionan presión contra el suelo.

La línea de perforación se alarga gradualmente, incorporando aproximadamente cada 10 metros un nuevo tramo de tubo en la superficie entre este y el Kelly, mientras se perfora, se bombea dentro del caño un lodo especial (fluido de perforación), preparado con arcilla, agua y varios productos químicos, el lodo se inyecta a la tubería y al circular permanentemente retorna por el exterior, lubrica y refrigera la broca y sostiene las paredes del pozo, la finalidad del lodo es traer a la superficie el material sólido que se va desintegrando y los pedazos de roca cortados, el lodo en superficie es filtrado de impurezas y escombros para ser bombeado al pozo.



Fig. 1, Torre de perforación realizando un pozo para extracción de crudo.

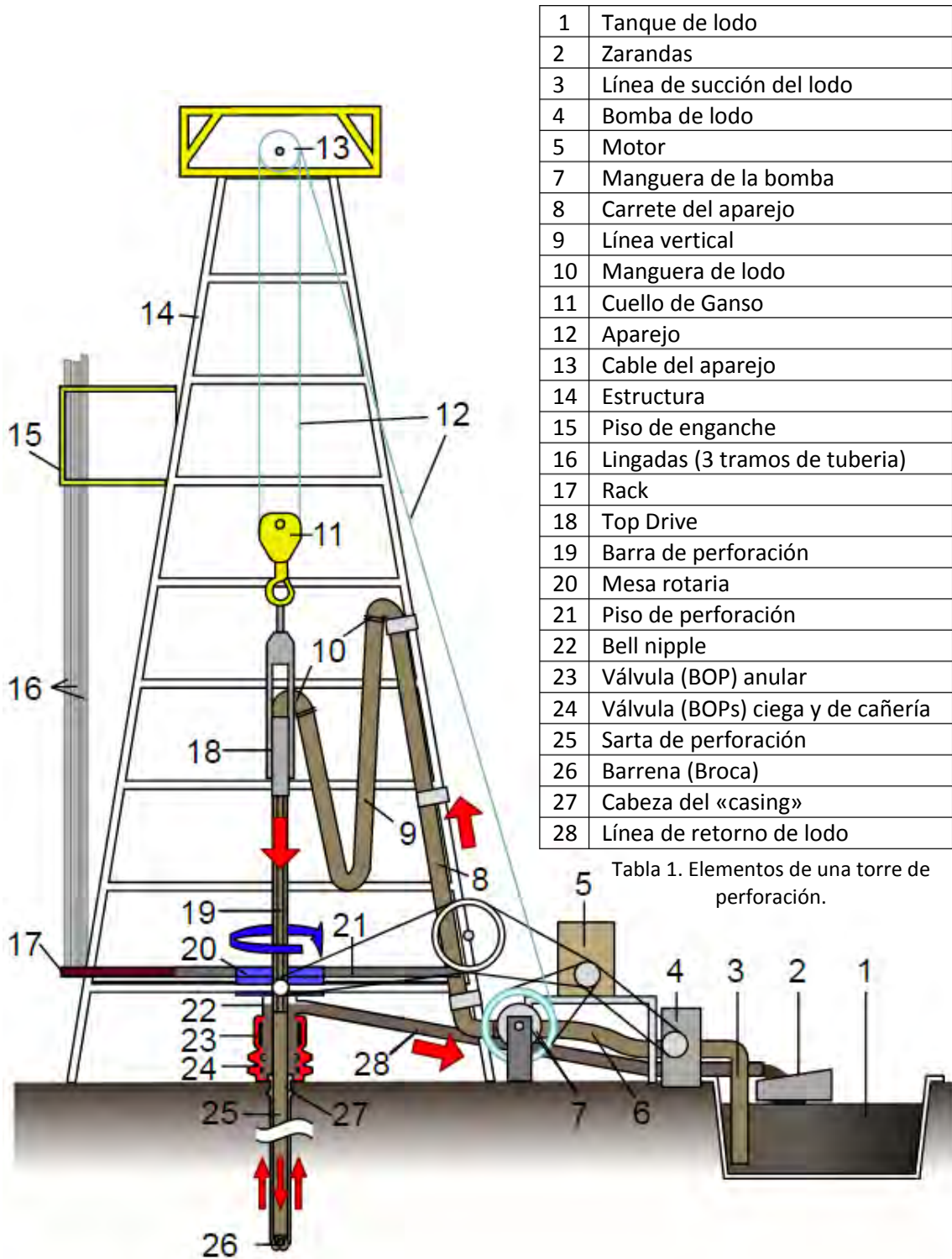


Fig. 2, Diagrama simplificado de una torre de perforación.

4.2.2 Broca o Barrena de perforación.

El proceso de perforación de un pozo se lleva a cabo mediante una herramienta llamada broca o barrena, la cual está localizada en el extremo inferior de la sarta de perforación que se utiliza para cortar o triturar la formación rocosa penetrando el subsuelo terrestre, la acción de corte de sus dientes consiste en el movimiento rotatorio y la fuerza para penetrar en el suelo procede de la carga ejercida por las tuberías que soportan.

Si se cuenta con datos precisos sobre las formaciones que deberán perforarse en el intervalo objetivo, se podrá seleccionar con más facilidad la estructura óptima de corte y la densidad que requiere la aplicación.

Generalmente la información litológica es fundamental para determinar la mejor selección de la barrena, definidos los tipos de rocas, se pueden asociar con la mecánica de corte de las barrenas de diamante, el tipo de roca ayuda a determinar el tipo de corte necesario para vencer su resistencia y ayuda a determinar el mejor tipo de corte: surcado o molido.

Desde hace tiempo la selección de la barrena se considera clave para el éxito de las operaciones de perforación, la barrena correcta desempeña un rol esencial en la optimización de la velocidad de penetración (ROP), lo que ayuda a minimizar Los costos del equipo de perforación y acorta el tiempo existente entre la puesta en marcha de un proyecto y la primera producción.

En los programas de desarrollo de campo petrolero, la predicción de la ROP es crítica para la asignación eficiente de equipos de perforación, personal y material, los operadores están perforando pozos de alcance extendido cada vez más complejos en los que una barrena que no se adecúa correctamente a la formación puede introducir aspectos dinámicos indeseados o generar fuerzas que produzcan la desviación del pozo, por el contrario una barrena diseñada correctamente genera un pozo más calibrado y un trayecto menos tortuoso.

Los factores que afectan el desgaste de las barrenas se puede dividir en geológicos, operativos, de manejo y de transporte, los dos últimos parámetros pueden obviarse; pero el primero debe ser bien estudiado antes de definir el tipo de barrena que se va a utilizar. Esto permitirá minimizar el desgaste y determinar su rendimiento de operación sobre las formaciones que se van a perforar.

La IADC (International Association of Drilling Contractors), ha desarrollado códigos para la denominación de los diferentes tipos de barrena de acuerdo con sus aplicaciones, el desgaste de las barrenas se debe evaluar de acuerdo con el código IADC para calificación de barrenas³.

³ <http://www.iadc.org/category/drillbits/>

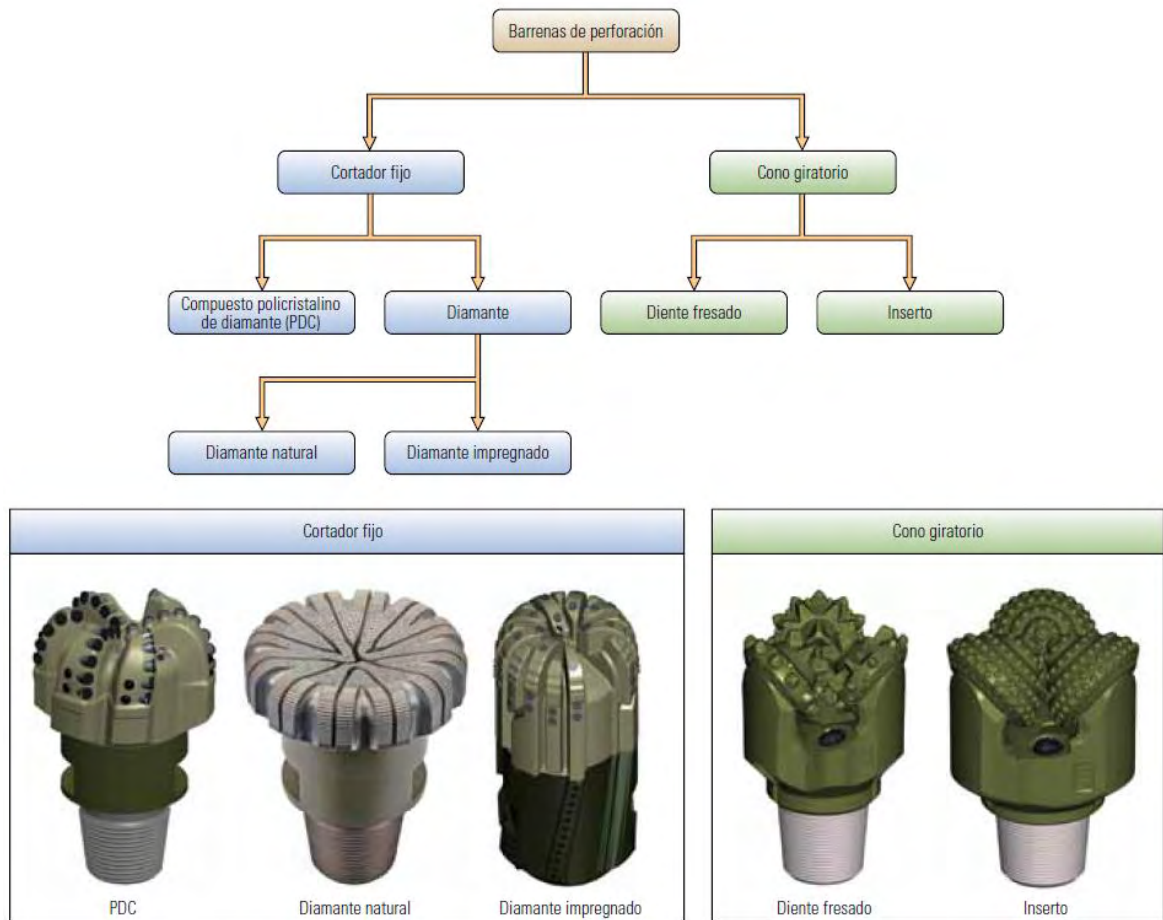


Fig. 3, Diseños de barrenas de perforación.

4.2.3 Fluido o Lodo de Perforación (Mud)

En el uso general, cualquiera de una serie de fluidos líquidos y gaseosos utilizados en operaciones de perforación de pozos de sondeo en la tierra, es sinónimo de "fluido de perforación", aunque algunos prefieren reservar este término a los lodos más sofisticados y bien definidos.

Los fluidos de perforación constituyen una parte vital e importante para ser empleado durante la construcción de un pozo, sus funciones son tan básicas como fundamentales para llevar a buen término la perforación que permitirá la explotación de un pozo de petróleo y/o gas.

Estos fluidos deben ser formulados de manera tal que puedan conservar su calidad y utilidad de la manera más óptima posible, es por ello que su preparación se realiza en el campo y para un tiempo próximo a la fase en la cual ha de ser

empleado, de manera tal que conserve sus propiedades lo más óptimo posible, como condiciones generales, estos fluidos no tienen que ser tóxicos ni corrosivos, pero si mantenerse inerte ante la presencia de otras sustancias, como fluidos de la formación por ejemplo, y así como permanecer estable ante altas temperaturas.

Se ha intentado clasificar los fluidos de perforación de muchas maneras, un esquema de clasificación general, se basa únicamente en la composición del lodo, distinguiendo el componente que define con claridad la función y el rendimiento del fluido: (1) a base de agua, (2) no a base de agua y (3) gaseoso (neumático).



Fig. 4, Tipos de fluidos de perforación.

El fluido de perforación o lodo, también llamado bentonita, circula dentro del hoyo para cumplir una serie de funciones vitales, como revocar las paredes de la perforación evitando o previniendo derrumbes, lubricar y refrigerar la broca y llevar los residuos a la superficie.

Revisión de tanques y capacidades.

La revisión constante de los volúmenes en cada uno de los tanques es responsabilidad vital de los Ingenieros de lodos, ya que con ello monitorean la funcionalidad del circuito cerrado, identificando cualquier situación que conduzca a problemas con el fluido (pérdida de circulación, por ejemplo) y responder a tal situación de la manera correcta, el ingeniero de fluidos por lo general emplea una serie de fórmulas para estimar el comportamiento teórico de cada desplazamiento, entre las fórmulas más empleadas está la de cálculo de volumen en tanques rectangulares:

$$V = L \text{ (pies)} \times W \text{ (pies)} \times h \text{ (pies)} / (5,61 \text{ bls/ft}^3)$$

Donde L es la longitud, w el ancho y h la altura del tanque, todas estas medidas están en pies, y para transformar los pies cúbicos en barriles se divide entre la constante 5,61.

Funciones del lodo de perforación.

Estas funciones describen las tareas que los lodos de perforación desempeñan, aunque no todas se aplicaran en cada pozo, puesto que hay otros factores que influyen en la utilización de dichos lodos, como por ejemplo las características del terreno y las necesidades específicas de perforación, la remoción de los recortes del pozo y el control de las presiones de la formación son funciones sumamente importantes.

Aunque el orden de importancia sea determinado por las condiciones del pozo y las operaciones en curso, las funciones más comunes de los lodos de perforación son las siguientes:

- Mejorar la limpieza de los recortes durante la perforación.
- Controlar las presiones de la formación.
- Aumentar la velocidad de penetración.
- Facilitar la cementación.
- Aumentar la estabilidad del pozo.
- Proporcionar intensa refrigeración en el área de fricción.
- Proporcionar excelente lubricación al sistema.
- Transmitir la energía hidráulica a las herramientas y a la barrena.
- Reducir las pérdidas por filtraciones en formaciones permeables.
- Prevenir la oxidación del acero componente del instrumental.
- Evitar la expansión de las arcillas con alto poder de encapsulación.
- Llevar los residuos a la superficie.

4.2.4 Definición de Mud Logging.

La palabra Mud Logging significa "registro del lodo", es un servicio que es contratado por las compañías petroleras durante las distintas operaciones que se llevan a cabo durante la explotación de un depósito de hidrocarburos y el proceso consiste en la recolección, almacenamiento e interpretación de datos, tanto geológicos como ingenieriles que se obtienen durante la perforación de un pozo petrolero, juega un papel importante al momento de informar sobre cambios significativos dentro de los parámetros preestablecidos en el transcurso de la perforación, siendo esta su principal razón⁴.

⁴ INTERNATIONAL LOGGING, INC. Basic Mud Logging Manual, Versión 1.0. p.59-67

El Mud Logging es un conjunto integral de procesos, que tienen lugar durante la fase de perforación y a menudo dura hasta la fase temprana de producción, su ejecución entrega como resultado información importante del yacimiento, esta información tiene su origen principalmente en las muestras tomadas del fluido de perforación, en el gas liberado de las formaciones geológicas atravesadas durante la perforación y en varios parámetros físicos medidos durante estas operaciones.

Durante el proceso de Mud Logging se hace toda la recolección de datos, de la que se obtienen varios parámetros de perforación gracias a los sensores instalados por toda la plataforma de perforación, sumado a estos parámetros tenemos el análisis de las muestras tomadas del fluido de perforación, para su posterior procesamiento, análisis geológico e interpretación litológica, toda la información se almacena en una base de datos y finalmente se refleja en diferentes reportes, informes y gráficos, según los requerimientos de cada cliente.

El contenido que suministra la estación de Mud Logging, constituye una valiosa herramienta para determinar diferentes variables, tales como el tipo de roca que se está perforando y elementos químicos existentes en el suelo, los cuales van determinando la posible existencia de hidrocarburos, también posteriormente ayudan a tratar aquellos puntos críticos en la estabilidad de un pozo o su comportamiento desde el punto de vista de la reacción de la formación litológica con respecto a las propiedades de los fluidos de perforación.

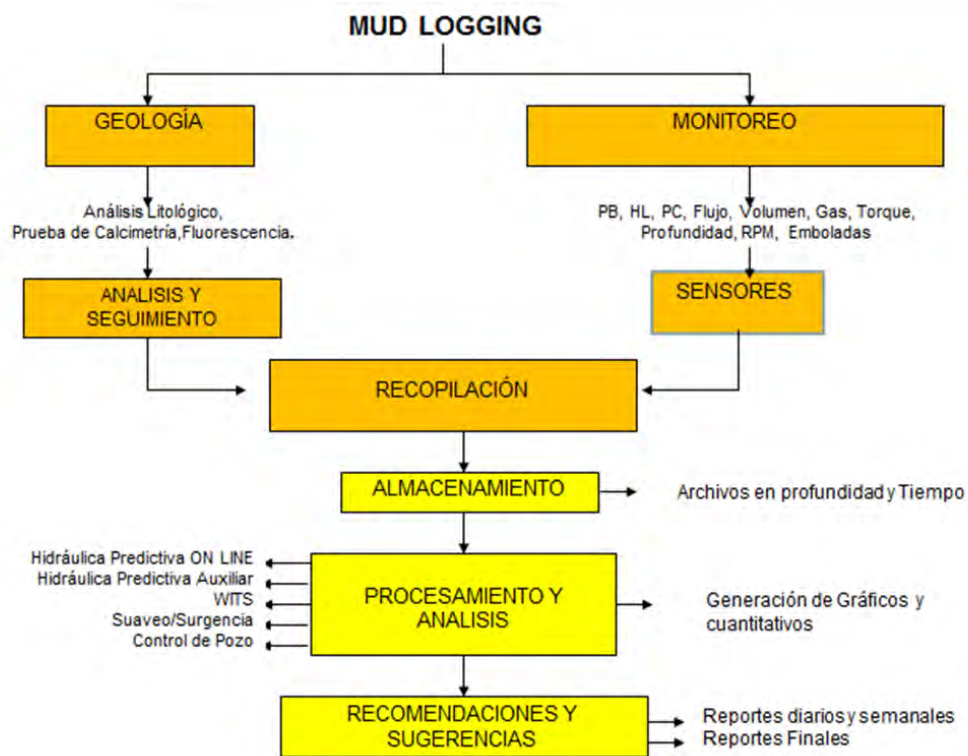


Fig. 5, Esquema funcional de una estación de Mud Logging

También es muy importante el rol que ocupa este servicio dentro del control en la seguridad en el pozo, basándose en los parámetros analizados y en los análisis e interpretación de datos (gas, presión, hidráulica....) Todo ello con el objetivo primario de asistir con eficacia y eficiencia en la ejecución de la perforación.

Otro de los aspectos fundamentales de la unidad de Mud Logging es el hardware y software base, el monitoreo y registro de las variables físicas que intervienen en el proceso se realiza a través de un sistema de adquisición de datos conocido como DAQ y de un conjunto de sensores industriales, cuyo aporte es determinante no solo desde el punto de vista tecnológico, sino como una herramienta de trabajo, porque su importancia radica en detectar cualquier comportamiento anormal de dichos parámetros y determinar la naturaleza e importancia de un determinado evento (presiones anormales, pérdidas o ganancias en el sistema activo de circulación...) y que en aquellos momentos críticos de la operación, el registro (log) permita dar la voz de alarma al equipo y a los supervisores para tomar decisiones acertadas que contribuyan a superar en forma rápida, acertada y oportuna los inconvenientes que se presenten y con base en la información recolectada por los instrumentos, permita determinar las soluciones necesarias para el buen éxito de la operación.

En este proceso converge la investigación y el trabajo realizado en éste proyecto, ya que uno de los objetivos principales del Mud Logging es registrar diferentes parámetros que intervienen en el proceso de perforación, tales como flujo de lodo, nivel del tanque, presión de las bombas y profundidad del pozo, de ésta forma se puede monitorear y controlar el proceso y determinar la causa ante una eventual falla.

Teniendo como base la experiencia desarrollada en este campo, es precisamente en esta etapa donde se determinó, que se hace evidente la necesidad de diseñar e implementar un sistema que permita monitorear y registrar las variables físicas que intervienen en el proceso de perforación, ya que se encontró que generalmente, una estación de Mud Logging, no cuenta con éste instrumento conocido como DAQ, el cual junto con los sensores industriales, son generalmente propiedad de las empresas perforadoras dueñas de las plataformas y cuyo costo comercial es considerablemente alto.

4.3 VARIABLES QUE INTERVIENEN EN EL PROCESO DE MUD LOGGING

A continuación se describen las variables más relevantes del proceso de perforación que intervienen en el proceso de Mud Logging, las cuales son monitoreadas y registradas para su posterior procesamiento, análisis y evaluación, independientemente de la interpretación litológica de las muestras.

4.3.1 Profundidad y Rata de Penetración (ROP).

Saber la profundidad actual de la broca en todo momento durante la perforación y otras maniobras es de la mayor importancia, esta información es contra la cual todos los otros valores y datos son referenciados, durante la perforación permite determinar los cambios en el pozo y en la formación y permite también llevar a cabo los cálculos de presión, el cambio en la rata de perforación (ROP) permite identificar cambios litológicos y en las condiciones de perforación, durante las maniobras de viaje, conocer la profundidad y la velocidad de movimiento permite determinar y monitorear los volúmenes de desplazamiento y las presiones inducidas, esto permite también sentar revestimientos en puntos especificados previamente y localizar precisamente las zonas productivas.

La rata a la cual el pozo está siendo perforado proporciona uno de los principales parámetros registrados durante la operación de perforación. Las unidades de medida pueden estar en términos de la profundidad ganada sobre un intervalo de tiempo dado (eg m/hr, ft/hr) ó en términos de la longitud de tiempo tomada para perforar un intervalo de profundidad dado, por ejemplo:

$$60 \text{ m/hr} = 1.0 \text{ min/m}, 30 \text{ m/hr} = 2.0 \text{ min/m}$$

Hay varios factores que pueden afectar la ROP, que incluyen:

- Selección de la Broca
- Peso sobre la Broca (WOB)
- Velocidad de Rotación (RPM)
- Presión Diferencial
- Hidráulica y limpieza del hueco
- Desgaste de la barrena

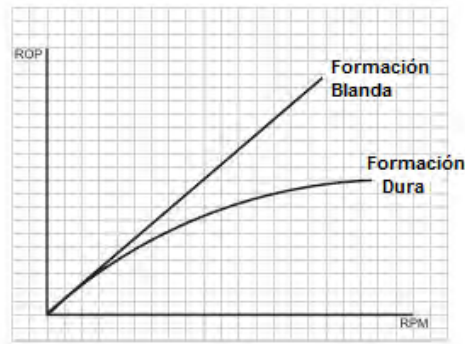


Fig. 6, Rata de penetración vs Velocidad de Rotaria

Aumentos de penetración (Drilling Breaks). Un marcado aumento en la rata de penetración es conocido como un –drilling break- (igualmente una reducción es conocida como un drilling break negativo) y es responsabilidad de ambos del personal de registro y del perforador identificar lo más rápido posible que está causando este rompimiento rápido, cuando el personal de registro identifica un drilling break, el o ellos deben notificar al perforador inmediatamente y también deben dejar registro de que esto se ha hecho.

La importancia de la forma del drilling break, es que si el cambio de la ROP no fuera debido a un cambio en los parámetros de perforación tales como WOB o RPM, entonces es debido a un cambio en la formación. Simplemente puede ser un cambio litológico (más suave, menos consolidado, más débil o nada de cemento, una porosidad mayor, etc.), en tal caso el personal de registro debe “lagear el break” y recoger cortes adicionales, cuando llegue a superficie para identificar el cambio.

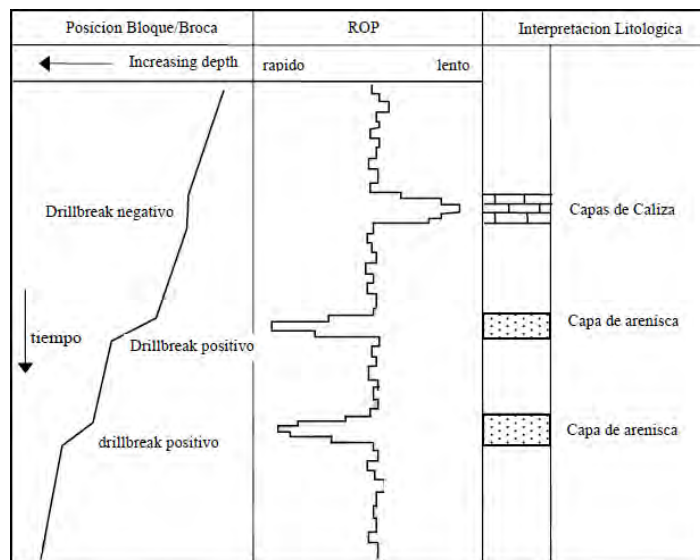


Fig. 7, cambio en ROP causado por un *Drilling Breake*.

4.3.2. Carga del Gancho y Peso Sobre la Broca (Hookload y WOB).

Conocer el peso de la sarta de perforación que es soportada por el gancho permite determinar información importante cuando se perfora o se corre tubería.

El peso total de la sarta de perforación (BHA y tubería) es conocido como peso del gancho, se determina multiplicando la densidad de la tubería componente Kg/m o lbs/pie, por la longitud de cada sección, un sensor colocado abrazando el cable de perforación que soporta el bloque viajero y el gancho mide este peso, esto es conocido como carga del gancho (hookload).

Cuando la sarta esta fuera de fondo y sin movimiento, la carga del gancho es igual al peso efectivo de la sarta cuando el bloque soporta todo el peso de la misma, el peso efectivo de la sarta será ligeramente diferente al calculado o teórico pues se debe a un factor de flotación en el lodo, cuando la broca toca el fondo del hueco, la carga del gancho disminuirá, parte del peso de la sarta se apoyara en el fondo del hueco, esto es conocido como el peso sobre la broca.

Peso de la sarta = carga del gancho + peso sobre la broca

El peso que se transfiere desde el gancho soportado, al fondo del pozo, es controlado por el perforador, este opera un freno con el cual puede controlar la descarga de la línea del taladro desde el malacate, mientras levanta o baja el bloque y el gancho, por el descenso del bloque cuando la broca está en fondo, mayor peso de la sarta será transferida al fondo del hueco, por consiguiente nosotros veremos un aumento en el peso sobre la broca, como se explicó anteriormente, mientras se aumenta el WOB se producirá una rata de penetración ROP mayor.

El objetivo del operador en este caso será medir los cambios en la tensión de la línea de perforación que es debido a las diferentes cargas.

Peso sobre la broca (WOB). Conocer el peso del gancho, y por consiguiente el WOB, permite al perforador controlar la cantidad de peso o fuerza que es aplicada a la broca, manteniéndola constante o haciendo cambios, desde el punto de vista de registros, esto le permite al personal de registro determinar si los cambios en la rata de penetración son debidos a cambios en el WOB, los cambios en el WOB afectaran la rata de penetración, el desgaste de la broca y el control direccional.

Hay dos principios que controlan el máximo peso que puede ser aplicado a la broca:

- Las especificaciones técnicas del fabricante deber ser reconocidas para prevenir el desgaste o fallas en las balineras y no exceder las limitaciones de la broca.

- El peso global de los “drill collars” provee la cantidad y también las limitaciones de peso, el peso del “drill collar” siempre debe exceder el WOB, esto asegura que los “drill collar” siempre estén en compresión considerando que la sarta de perforación siempre está en tensión.

4.3.3 Velocidad de Rotación y Torque de la Rotaria.

Velocidad de Rotación (RPM).

La rotación puede ser aplicada en la broca desde la superficie o desde los motores de turbina en el fondo del hueco, la rotación en superficie puede proporcionarse a través de la mesa rotaria y de la “Kelly”, o a través de piezas de poder o “top drives”. La RPM es medida por un sensor de proximidad que detecta un objetivo metálico el cual está atado bien sea a la mesa rotaria, al árbol de leva rotatorio, o en la parte superior del top drive, un pulso o señal, es producido cuando cada objetivo pasa por el sensor.

La rotación aplicada por los motores de fondo o turbinas, depende de la cantidad de flujo que pase a través del motor, una rata de flujo rápida, producirá más rotación, por ejemplo, Si una turbina genera una rotación por cada 10 litros de fluido que pasa a través de la misma:

$$(0.1 \text{ revs/litro}): \text{Rata de flujo de circulación} = 1.5 \text{ m}^3/\text{min. (1500 litros/min)}$$

$$\text{Entonces, las RPM del motor} = 1500 \times 0.1 = 150 \text{ RPM}$$

Por consiguiente las RPM del motor son determinadas numéricamente, multiplicando un factor (revoluciones por unidad de volumen del fluido – este número es proveído por el operador del motor) por la rata de flujo medida en superficie (la velocidad de la bomba x la capacidad de la bomba.

Dependiendo del tipo, tamaño y capacidad y tipo de ensamblaje, las brocas tienen rangos específicos de rotación para prolongar la vida de la misma y lograr óptimas ratas de penetración, un cambio en las RPM tiene un efecto directo en la rata de penetración.

El objetivo del operador en este caso será determinar desde su propio punto de vista si los cambios de rata de penetración son debidos a cambios en la RPM.

Torque de la Rotaria (Rotary Torque).

Torque es la medida de la fuerza necesaria para producir una rotación en la broca y sarta de perforación, hay una relación directamente proporcional entre el torque y la velocidad de rotación, de igual forma una velocidad lenta causara un torque bajo.

La medida del torque varia, dependiendo del tipo de fuente que maneje el sistema rotatorio, la mesa rotaria puede ser manejada eléctricamente o mecánicamente. Igualmente los “Top Drive” pueden ser mecánicos o eléctricos.

Con los sistemas eléctricos, el torque representa la cantidad de corriente que es requerida para manjar la mesa rotaria, esto es determinado por una pinza de torque que mide el campo magnético inducido alrededor del cable de poder, con las mesas rotarias manejadas mecánicamente, la tensión producida en las cadenas de la rotaria, son los cambios en el torque, esta tensión es medida con un transductor de presión (el principio es el mismo que con el de la carga del gancho que mide los cambios en la tensión de la línea de perforación), un traductor de presión normal es todo lo que se requiere para medir el torque en los sistemas que son manejados hidráulicamente, la unidad de medida del torque de la rotaria es la fuerza que es aplicada contra una distancia movida, por ejemplo: Newton Metro (Nm) o libras por pie (Lbs/ft).

Si una medida eléctrica es la obtenida, el torque puede ser expresado en términos de corriente (Amperios) o puede ser convertida a una unidad de fuerza por distancia, sin embargo, esta conversión no es lineal, y variara en cada equipo dependiendo del poder y del equipo de rotación.

En condiciones similares a las demás variables expuestas, el torque de la rotaria se incrementa con la profundidad desde el aumento de la sarta de perforación y por consiguiente el aumento de la fricción con las paredes del pozo, la fricción fluctúa contra la rotación, de tal forma que con más tubería en contacto con el hueco, más fuerza es requerida para producir la misma rotación, un cambio en el torque es una medida de los cambios de fuerza friccionales que actúan contra la rotación y puede ser el resultado de cambios mecánicos, fallas mecánicas o cambios en el fondo del hueco.

Diferentes tipos de brocas y cortes en superficie también resultan en diferentes medidas del torque, pero no obstante el torque proporciona una muy útil información en lo que se refiere a la evaluación de la formación y la condición del agujero, es por consiguiente muy importante que los cambios en el torque sean evaluados y que sean determinadas las causas que lo generaron.

Incrementos en el torque pueden ser el resultado de:

- Aumento en el WOB
- Desgaste de la broca
- Aumento de las RPM
- Desgaste o falla en las balineras
- Perdida de los conos
- Incremento en la presión de formación
- Limpieza pobre del hueco lo que produce brocas redondeadas

- Desviación del hueco
- Fracturas, llevando típicamente a torque alto y errático
- Huecos apretados
- Pegamiento de la tubería. En esta situación las pegas de tubería pueden llevar a torques altos, Incluso llevar a parar la rotaria, y el riesgo de una falla en la tubería, o “twist-off” desenroscamiento, es muy significativo.

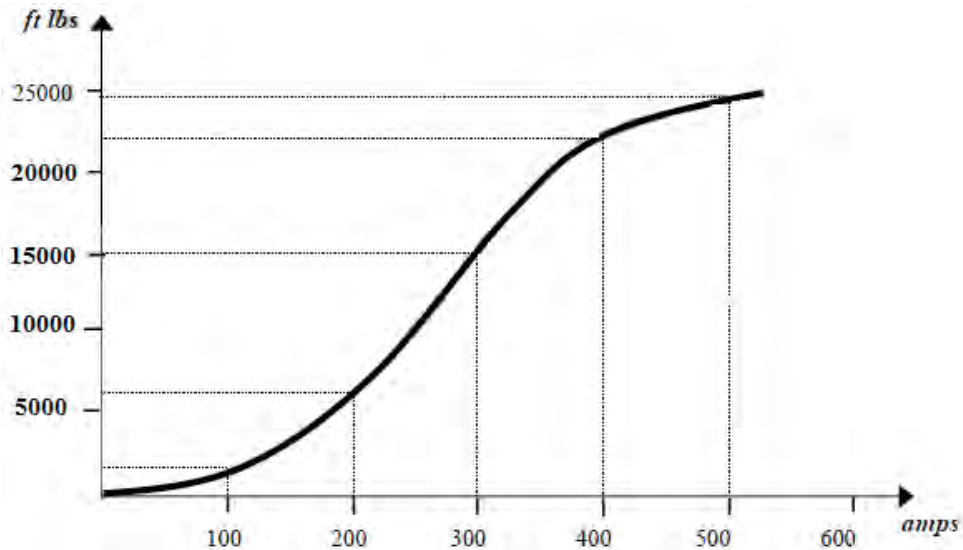


Fig. 8, Relación de torque (ft-lbs x amperio).

4.3.4 Rata de Flujo y Nivel de los Tanques.

Como se ha descrito previamente, el sistema de circulación de lodo puede ser considerado como un sistema cerrado y el flujo que sale del anular es el mismo es el mismo flujo de lodo que está entrando por la tubería, la rata de flujo entra al hueco, vía bombas, líneas de presión y tubería, es determinado por la velocidad de la bomba y la capacidad de la misma:

$$\text{Rata de flujo } Q = \text{SPM} \times \text{Capacidad de la bomba} \times \text{Eficiencia}$$

La rata de flujo de salida del hueco es típicamente determinada en la línea de flujo, la cual está conectada de la cabeza del pozo a la caja de las zarandas, normalmente un sensor de deflexión de paleta o de velocidad de una turbina son usados para determinar la rata de flujo de salida del lodo.

Para realizar una determinación de la rata de flujo real, la medición se hace cualitativamente, por ejemplo, cuando usamos la paleta de flujo, una deflexión completa representaría el 100% de flujo y cero deflexión, con la paleta en reposo, representa cero flujo, para una rata de bomba constante, la rata de flujo saliendo

del hueco (MFO) también debe ser constante, este parámetro debe ser el indicador primario de cualquier cambio en el fluido en el fondo del hueco.

Por ejemplo:

Una reducción en el MFO – pérdida de circulación, pérdida de fluidos en fracturas

Un incremento en el MFO – Influjos de fluidos de formación, arremetidas

El nivel de los tanques (tanques de lodo) es supervisado por una razón similar, primordialmente, sin cambios en la tasa de bombeo, el volumen de lodo en los tanques solamente caerá un poco de acuerdo al nuevo hueco perforado (alineando al hueco, parte del activo o la succión al sistema de tanques), cualquier variación en el volumen del tanque puede indicar un cambio en las condiciones del fondo del hueco, una pérdida de volumen del lodo indica una pérdida de lodo en formación o un incremento del volumen de los tanques indica un influjo de fluidos.

El volumen de los tanques es monitoreado por dos tipos de sensores, ultrasónicos o flotadores de varilla (boya), los ultrasónicos son colocados en la parte superior del tanque mandando continuamente pulsos de sonido, los cuales son reflejados desde la superficie del lodo al sensor, las dos señales son procesadas y calibradas en términos de distancia del sensor y a su vez son convertidos en volumen de lodo equivalente, el operador tiene que tener mucho cuidado en la colocación del sensor, evitando señales erráticas al colocarlo en sitios de turbulencia del lodo, como los agitadores o entradas de lodo al tanque.

Las varillas consisten en flotadores, colocados en un cuerpo de acero, los cuales suben y bajan de acuerdo a los cambios en la altura del nivel del lodo, cuando esto sucede, el flotador pasa por sensores magnéticos colocados dentro de la varilla, estos determinan la posición del flotador, la altura a la cual es calibrado equivalente a un volumen de lodo.

Así como es un indicador primario de pérdidas de lodo o influjo durante la perforación, el nivel de los tanques es monitoreado en otras operaciones, como por ejemplo:

- Desplazamientos de lodo durante los viajes sacando tubería.
- Desplazamiento del lodo en corridas de revestimientos, asegurándose que el lodo no sea perdido debido a presiones de surgencia rompiendo la formación.
- Desplazamientos del lodo durante bombeo de cemento, nuevamente asegurándose que no esté causando un rompimiento de la formación, resultando en una pérdida de lodo.

Existen otras causas de cambios en el nivel del lodo, las cuales se deben tener en cuenta para evitar alarmas falsas e identificar errores en movimientos de lodo los

cuales podrían resultar en pérdidas en superficie y contaminación al medio ambiente:

- Adiciones de lodo nuevo (base agua, diesel, aceite, etc.) o químicos.
- Transferencia de tanques o desde una fuente externa.
- Válvulas que son abiertas por error.
- Volúmenes en las líneas de superficie (retornos dentro de los tanques cuando las bombas son apagadas o llenando las líneas cuando son enciendas).
- Cambios similares cuando los equipos de superficie, centrifugas, desarenadores, son encendidos o apagados.
- Cambios aparentes cuando los agitadores son apagados o encendidos.
- Movimiento de las olas en equipos flotantes.



Fig. 9, Sensores de nivel de tanque: Ultrasónico y de Varilla

4.4 MONITOREO DE VARIABLES Y SISTEMA DE ADQUISICION DE DATOS

Las variables descritas anteriormente, son las más relevantes durante el proceso de perforación del pozo y son las que generalmente se van a tener en cuenta para realizar el registro (Log) del proceso de Mud Logging, independientemente del análisis e interpretación litológica y geológica de las muestras, a cargo de los geólogos.

Estas variables son monitoreadas y registradas haciendo uso de un conjunto de sensores industriales implantados en la plataforma y un sistema de adquisición de datos (DAQ) instalado en la estación de Mud Logging.

5. METODOLOGIA

La base metodológica descrita a continuación, propone la descripción de los procedimientos realizados y los elementos que se utilizaron para cumplir con cada uno de los objetivos planteados en este proyecto.

5.1 VARIABLES FISICAS DEL PROCESO Y SENSORES INDUSTRIALES

5.1.1 Variables a Monitorear Durante el Proceso de Perforación.

Las variables que se determinó serán monitoreadas por el sistema desarrollado durante el proceso de perforación del pozo, son:

Flujo de lodo (Mud Flow). Para realizar una determinación de la rata de flujo real, la medición se hace cualitativamente, por ejemplo, cuando usamos un sensor de paleta de flujo, una deflexión completa representaría el 100% de flujo y cero deflexión con la paleta en reposo, representa cero flujo.

Nivel del tanque de lodo (Mud Level). El volumen de los tanques es monitoreado por un sensor ultrasónico, los ultrasónicos son colocados en la parte superior del tanque mandando continuamente pulsos de sonido, los cuales son reflectados desde la superficie del lodo al sensor, las dos señales son procesadas y calibradas en términos de distancia del sensor.

Torque del motor de taladro (Rotary Torque). El torque representa la cantidad de corriente que es requerida para manjar la mesa rotaria, esto es determinado por una pinza de torque que mide el campo magnético inducido alrededor del cable de poder.

Sensores Utilizados para el Monitoreo.

La escogencia del modelo o referencia del sensor que se utiliza para cada variable es transparente para el desarrollo del instrumento, ya que su diseño comienza por la adecuación de unas variables ya establecidas que entregan los tres sensores industriales que se han predefinido como caso de estudio.

Estos sensores industriales son de alto costo y generalmente hacen parte de la infraestructura base (Stock) propiedad de las empresas de perforación, por lo cual no se dispone de ellos para realizar pruebas de campo, sin embargo las características generales de funcionamiento se han obtenido desde las especificaciones técnicas y el manual de servicio de cada sensor, como igualmente se hizo en la práctica, por lo tanto dichas señales serán emuladas reemplazando los sensores descritos, por transductores configurados para entregar las mismas señales eléctricas y así permitir su posterior adecuación y procesamiento.

5.1.2 Sensor de flujo de lodo MD Totco Mud Flow Sensor 270700-Series:

Propósito:

El sensor fue diseñado por *M/D Totco*, para su instalación en una plataforma de perforación, indica el porcentaje de flujo de lodo en el sistema existente, la supervisión y monitoreo continuo de las condiciones de flujo de lodo están disponibles mediante un mecanismo neumático o eléctrico, una infiltración de gas, invasión de los fluidos, formación o pérdidas de circulación se detectan inmediatamente con la información entregada por este sensor.

Descripción:

El sensor de flujo de lodo se compone de una base que está soldada al tubo de flujo de retorno, una paleta (paddle) conectada a través de un pivote a un transmisor neumático o eléctrico de posición y dos contrapesos ajustables (figura 10). Los contrapesos son accesibles fuera de la unidad, proporcionan una fuerza en la paleta contra el flujo de lodo y aseguran que el compresor vuelve a la posición cero cuando no hay caudal en la tubería, la sonda se retira fácilmente y la placa de recubrimiento cubre la base después de que el sensor se quita de una instalación.

Operación:

El lodo que fluye a través del conducto de retorno desvía la paleta del sensor de flujo, el grado de desviación es determinado por la velocidad del flujo de lodo y la cantidad de flujo de lodo, así como el ángulo del tubo y el diámetro. La posición de los contrapesos también afecta la cantidad de deflexión, por lo tanto, el ángulo de la paleta del sensor de flujo cambiará en proporción a estas variables. El movimiento de la paleta se enlaza directamente con el eje transmisor, la rotación del eje transmisor determina la cantidad de señal de salida del transmisor, el sensor está diseñado para soportar una presión máxima de 150 psig (10.5 kg/cm²) dentro de la tubería.

Funcionamiento del Transmisor Eléctrico:

El transmisor eléctrico funciona a partir de 5,0 Vdc de entrada y proporciona una salida de 0,47 a 3,0 Vdc. Cuando el compresor no tiene caudal de presión en contra, la señal de salida es de 0,47 voltios, cuando la paleta está en plena deflexión (90°), la señal de salida es de 3,0 voltios, la calibración requiere un ajuste de la vinculación entre el transmisor y el eje de eje de pivote, que operan en una proporción de 1:1, los contrapesos de la paleta también requieren ajuste para el correcto funcionamiento del sensor.

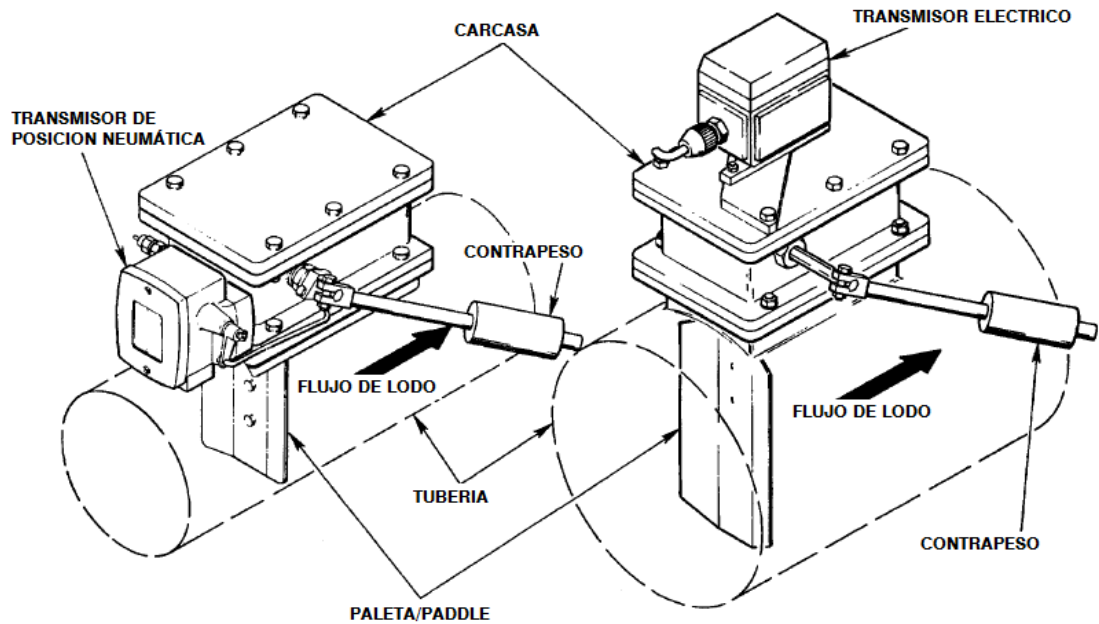


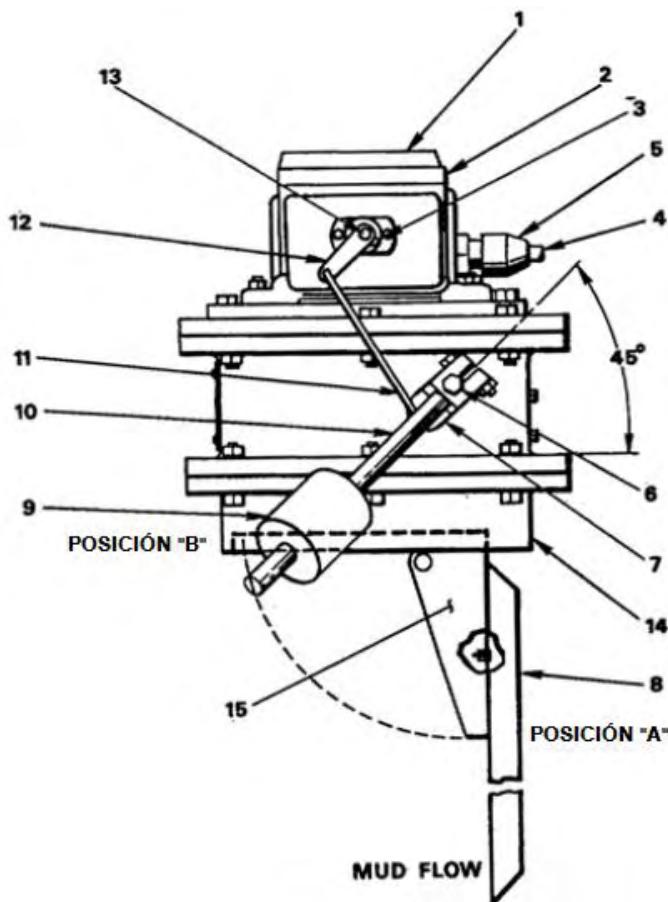
Fig. 10, Sensor de Flujo neumático y eléctrico *MD Totco Mud Flow Sensor 270700-Series*.

Instalación y Calibración:

Para realizar la instalación y calibración del sensor se deben realizar los siguientes pasos:

1. Coloque la paleta (8, Figura 11) en la posición "A".
2. Compruebe que la abrazadera del controlador (7, Figura 11) y el soporte del contrapeso (10) están a 45° desde la base (14) con la paleta en la posición "A" (véase la figura 11), si no es así, corregir el desalineamiento aflojando las tuercas de la abrazadera del controlador (7) y del soporte del contrapeso (10), cuando la abrazadera del controlador y el soporte estén en posición, apretar las tuercas.

3. Sin alimentación eléctrica, retire la tapa (1, Figura 11) del componente eléctrico (2) y el multímetro digital entre los terminales 1 y 2, ponga el multímetro en la escala de ohmios.
4. Con la paleta (8) en la posición "A" como se muestra en la Figura 11, afloje el tornillo de cabeza hueca (3) y gire el eje (13) para obtener 1450 ± 50 ohmios en el multímetro (el eje tiene 180° de rotación sin tope mecánico), aprete el tornillo de cabeza hueca para bloquear la palanca en esa posición (ver la tabla X para la verificación), si no hay 4.000 ± 30 ohmios en la posición "B", el eje está 180° fuera de fase. Al aplicar voltaje, las lecturas deberían aproximarse a las dadas en la tabla 3.
5. Colocar los contrapesos (9) a lo largo de los soportes (10) de modo que el compresor (8) vuelva a la posición "A" con ningún flujo de lodo.
6. Gire el soporte del contrapeso (10) varias veces para asegurarse de que el mecanismo se mueve libremente y vuelve a la posición "A".
7. Vuelva a poner la tapa (1).



1	Tapa
2	Componente Eléctrico
3	Tornillo de cabeza hueca
4	Conector del cable
5	Soporte del conector
6	Eje
7	Abrazadera del controlador
8	Paleta / Paddle
9	Contrapeso
10	Soporte del contrapeso
11	Varillaje
12	Palanca del transmisor
13	Eje del transmisor
14	Base
15	Soporte de la paleta

Tabla 2. Índice de partes del sensor de flujo.

Fig. 11, Ajuste del sensor eléctrico de flujo.

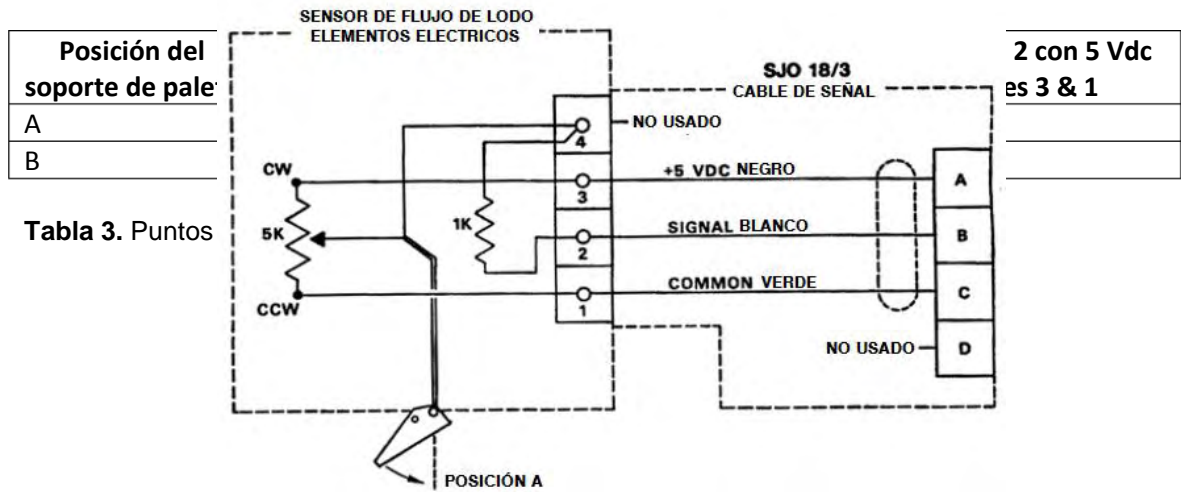


Fig. 12, Diagrama esquemático del componente eléctrico.

La sensibilidad del sensor de flujo de lodo se ve afectada por la posición de los contrapesos (9, Figura 11), existe mayor sensibilidad cuando los contrapesos están más cerca del eje (6) que cuando están más lejos, porque tiene menos flujo contra la paleta para mostrar un cambio en el flujo de lodo. Los contrapesos deben colocarse de manera que el compresor siempre vuelve a la posición "A" (Figura 11) cuando no hay flujo de lodo en el tubo, ambos contrapesos deben estar situados aproximadamente en la misma posición relativa, los tornillos en el contrapeso aseguran la posición.

Los siguientes pasos deben ser completados para completar ajustes del sensor de flujo:



Fig. 13, MD Totco Mud Flow Sensor 270700-Series.

1. Iniciar y las bombas y circular el flujo normal de lodo a través del sistema.
2. Ajuste la posición de los contrapesos (21, Figura 11), de manera que el transmisor indique el porcentaje de flujo deseado.
3. Apagar las bombas de lodo y verificar que la paleta vuelva automáticamente a 0% (posición "A"), si la paleta no regresa a 0%, reajustar la posición de los contrapesos en el soporte hasta obtener la condición deseada.

5.1.3 Sensor de Nivel del tanque *VEGAPULS 65 radar level sensor.*

Propósito:

El volumen de lodo en los tanques es monitoreado por sensores ultrasónicos los cuales son colocados en la parte superior del tanque para medir el nivel de altura del lodo, cualquier variación importante en el volumen del tanque puede indicar un cambio en las condiciones de operación del pozo.

Descripción:

Una sonda ultrasónica emite una serie de pulsos ultrasónicos desde el transductor, cada pulso se refleja como un eco en el lodo y es detectado por el cabezal de la sonda, el tiempo que emplea el pulso para viajar al líquido y volver, se mide y se convierte en distancia, la señal es entonces convertida a una magnitud eléctrica (miliamperios mA) y enviada al dispositivo DAQ para su procesamiento, las sondas ultrasónicas se alimentan con 24 Vdc.

Funcionamiento:

El VEGAPULS 65 es un sensor de radar con tecnología de banda C, con una frecuencia de emisión aproximada de unos 6 GHz, para medición continua de nivel, es especialmente adecuado para tanques que contienen líquidos corrosivos y en condiciones difíciles de trabajo.

La antena del sensor emite pulsos cortos de radar con una duración aproximada de 1 ns, estos pulsos se reflejan en el fluido y son recibidos por la antena como ecos, el tiempo empleado por los pulsos desde la emisión hasta la recepción es proporcional al distancia y por lo tanto al nivel, el nivel determinado es convertido en una señal de salida correspondiente y transmitido como valor medido.



Fig. 14, VEGAPULS 65 radar sensor.

La alimentación eléctrica y la transmisión la de valores medidos se realiza con un solo cable de dos líneas mediante el estándar 4-20mA, el rango de la tensión de alimentación puede variar, dependiendo de la versión de instrumento, generalmente se utilizan los 24 Vdc que entrega el DAQ.

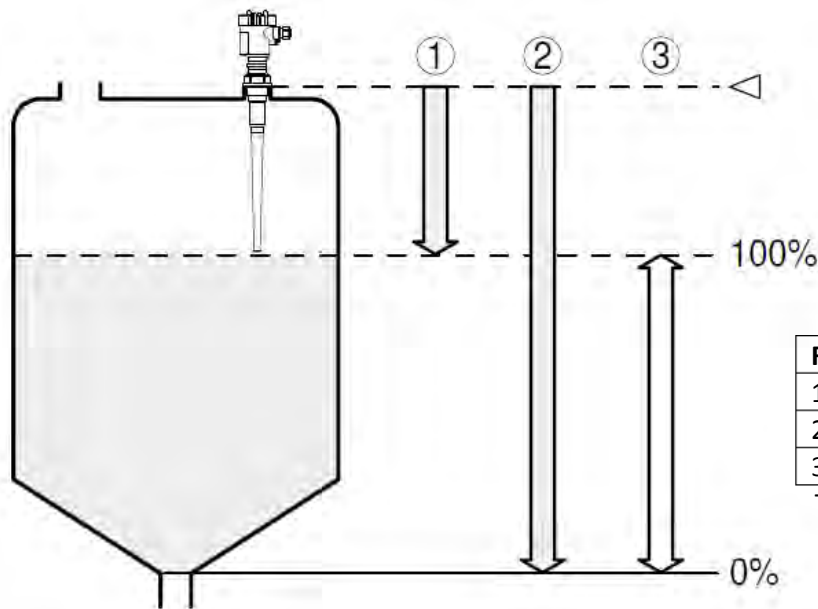
El sensor debe colocarse completamente perpendicular a la rejilla del depósito sin inclinarse. Idealmente, la sonda debe ser montada en un soporte por encima del tanque para evitar picos de señal, el sensor no debe estar en contacto con un material metálico. El intervalo mínimo es de 10" (25 cm), por lo que el sensor debe estar al menos a esta distancia desde el punto más alto del lodo en el tanque.

Instalación:

Seleccione una posición de instalación la cual permita lograr fácilmente el montaje y conexión, así como una posible adaptación y ajuste del módulo. La caja se puede rotar 330° sin el uso de herramientas, también puede instalar el módulo en cuatro posiciones diferentes (cada una desplazada por 90°).

Mediciones necesarias antes de calibrar el sensor:

1. La distancia desde el cabezal de la sonda en el fondo del tanque
2. La distancia desde el cabezal de la sonda con el lodo al máximo volumen



Rango de Operación	
1	Lleno
2	Vacío
3	Rango de Medición

Tabla 6. Rango de Operación de la sonda.

Fig. 15, Rango de medición de la sonda.

Calibración:

Para realizar la calibración se utiliza un sistema calibrador de procesos de 4-20mA para ajustar la precisión.

Calibración de nivel bajo: Si la sonda se ha configurado correctamente, al estar el tanque vacío el valor mínimo presente en el calibrador debe ser 4mA.

Calibración de nivel alto: Si la sonda se ha configurado correctamente, al estar el tanque en su nivel máximo el valor presente en el calibrador será de 20 mA.

5.1.4 Sensor de torque MD Totco Rotary Torque Sensor 40223503 Series.

Propósito:

Al avanzar en el proceso de perforación, el torque de la rotaria se incrementa con la profundidad, de tal forma que con más tubería en contacto con el hueco, más fuerza es requerida para producir la misma rotación, un cambio en el torque puede ser el resultado de cambios en el suelo, fallas mecánicas o cambios en el fondo del hueco, el torque proporciona una información valiosa en lo que se refiere a la evaluación de la formación y la condición del agujero, es por consiguiente muy importante que los cambios en el torque sean evaluados y que sean determinadas las causas que lo generan.

Descripción:

El sensor de torque es un transductor que se ancla alrededor de uno de los cables de alimentación del motor del taladro, el transductor recibe una señal de alimentación de 24 Vdc desde el DAQ y produce una señal de 4 a 20 miliamperios la cual representa proporcionalmente a un flujo entre 0-1000 Amperios de corriente a través del cable de poder.

Funcionamiento:

Cuando el transductor se ancla a una terminal de alimentación del motor, mide la cantidad de corriente usada por el motor la cual varía en proporción a la cantidad de torque desarrollado por el motor en un momento dado, el DAQ calcula entonces la cantidad de torque, este cálculo está basado en una tabla de

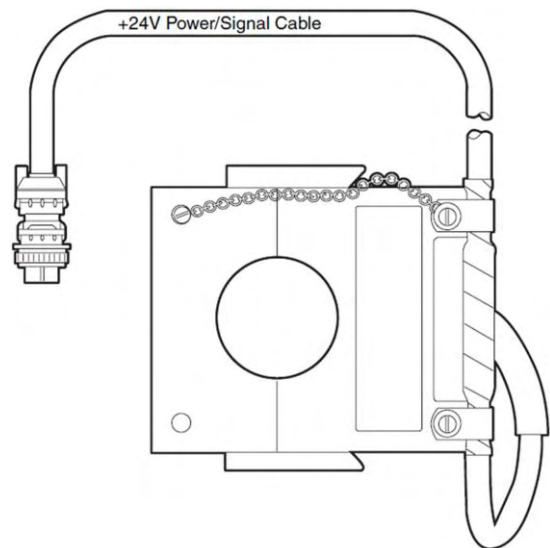


Fig. 16, Diagrama del sensor de torque.

correlación pre-determinada en la programación del DAQ, la cual corrige la curva no lineal entre corriente/torque y la transmisión entre el motor y la rotaria.

Comúnmente hay variaciones en la forma en que los clientes prefieren ver el torque relativo, algunos clientes prefieren ver un gráfico plano con solo pequeños picos durante la perforación, mientras que otros prefieren una lectura precisa, esto puede requerir un ajuste directo de la tabla de correlación y la frecuencia de actualización del canal del DAQ.

Calibración:

Para realizar la calibración se utiliza un sistema calibrador de procesos de 4-20mA para ajustar la precisión.

Calibración de nivel bajo: Cuando la rotaria no esté girando, se aceptaran valores mínimos presentes en el calibrador de entre 4-6 mA.

Calibración de nivel alto: Una calibración adecuada se debe realizar cuando el taladro esté perforando, para de esta forma establecer un nivel promedio de torque durante el proceso normal de perforación, de esta forma esta calibración permitirá identificar un aumento significativo en el torque.



Fig. 17, MD Totco Rotary Torque Sensor 40223503 Series.

5.2 ADQUISICIÓN Y ACONDICIONAMIENTO DE SEÑALES

A continuación se describen los procedimientos y elementos empleados para realizar la adquisición y el acondicionamiento de las señales adquiridas por los sensores industriales, descritos en la sección anterior.

5.2.1 Lazo de Corriente 4-20 mA (Current Loop).

Las señales analógicas en entornos industriales vienen reguladas por la norma ANSI/ISA50.1-1982 (R1992) (Compatibility of Analog Signals for Electronic Industrial Process Instruments), que establece los siguientes tipos de señales:

Salida de tensión:	Salida de corriente:
Rango 0/5 V	Rango 0/20mA
Rango -5/+5 V	Rango 4/20mA
Rango 0/10 V	
Rango -10/+10 V	

Tabla 4. tipos de señales ANSI/ISAS50.1- 1982(R1992).

Para adecuar la señal de los sensores industriales analizados en la sección anterior, dos de los cuales entregan una señal de 4-20mA, en concreto se debe detallar los tipos y modos de funcionamiento de la salida de corriente, el lazo de corriente es ampliamente empleado en procesos industriales de control, la principal ventaja, es que la señal no se ve afectada por la caída de tensión que se produce en la línea, lo que permite controlar actuadores u obtener medidas de sensores colocados a grandes distancias (1000m e incluso más).

La salida del transmisor es de corriente y proporcional a la variable que se desea transmitir, normalmente se emplean 4 mA para representar el nivel bajo del rango y 20 mA el nivel alto, de este modo, permite al receptor detectar si se ha producido algún fallo en la línea de transmisión e incluso alimentarse del propio lazo de corriente.

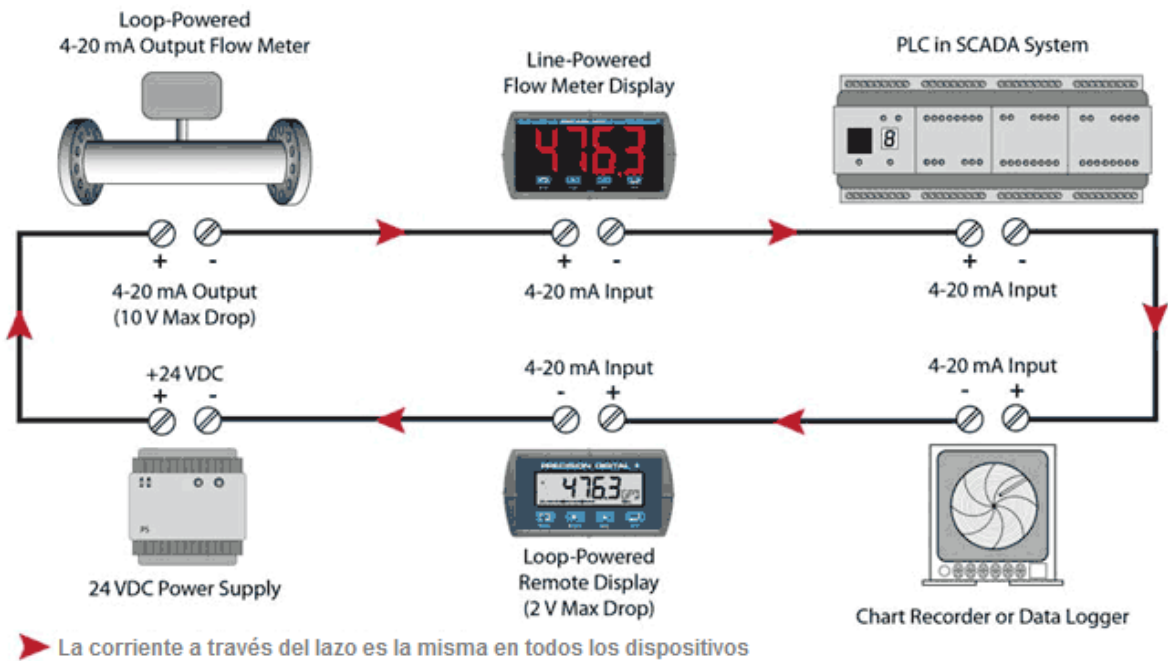


Fig. 18, Topología común de un Lazo de Corriente 4-20mA.

5.2.2 Transmisor de Corriente de 4-20 mA.

La forma más popular para transmitir señales en instrumentación industrial, es el estándar 4 a 20 miliamperios DC, esta es una señal industrial estándar, que significa que la señal de corriente es usada proporcionalmente, para representar magnitudes de variables medidas.

Típicamente, un valor de 4 miliamperios de corriente representa 0% de medida, y un valor de 20 miliamperios representa un 100% de la medida, y cualquier otro valor entre 4 y 20 miliamperios representa un porcentaje entre 0% y 100%.

Generalmente un instrumento calibrador, generador de una señal de 4-20mA es utilizado para calibrar el DAQ simulando una señal de corriente con las mismas características de un transductor que entrega una señal de 4-20mA.

Para lograr simular las señales que entregan los sensores industriales de torque y de nivel de tanque, se desarrolló un circuito (Fig. x), capaz de entregar una señal de 4-20mA a partir de una señal de voltaje de 0-5 Vdc, con éste dispositivo se logró cubrir la necesidad de tener presentes los sensores industriales, ya que el dispositivo reemplaza totalmente las condiciones y requerimientos descritos para la operación del DAQ, en condiciones iguales que utilizando la señal entregada por los sensores industriales.

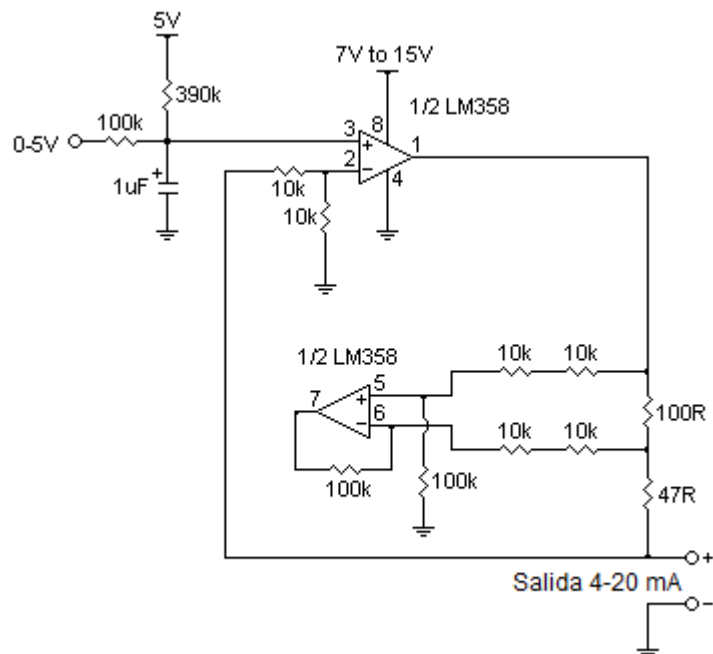


Fig. 19, Circuito Transmisor de 4-20 mA a partir de 0-5 vdc.

El anterior es un circuito de salida análoga con señales de alimentación y de salida adecuadas para el trabajo con microcontroladores, la característica principal es que cubre los rangos comunes de tensión y corriente utilizados en la instrumentación por el estándar 4-20mA.

La entrada análoga de 0-5 Vdc se puede generar, por ejemplo, desde un DAC, una señal PWM, o un potenciómetro, el resistor de 100k y el capacitor de 1uF filtran y suavizan la señal PWM que se convierte en Vdc, el primer amplificador operacional (Op Amp) es el controlador de salida y su ruta de realimentación es de voltaje, el primer Op Amp proporciona una ganancia de 2, el segundo Op Amp proporciona corriente de retroalimentación para el primer amplificador operacional, el circuito actúa como un amplificador diferencial, utilizando la resistencia 100R como elemento de detección de corriente, la resistencia de 47R no hace nada excepto arreglar un problema de linealidad en el circuito.

Tanto en el diseño, como en el montaje se utiliza como amplificador operacional el integrado LM358 Amplificador operacional doble de propósito general, no requiere fuente dual, alta ganancia, bajo consumo de potencia, ancho de banda 0.7 MHz. Encapsulado DIP (Dual in-line package), el circuito integrado LM358, tiene una amplia utilidad en diversos campos de la electrónica y es un circuito de fácil implementación.

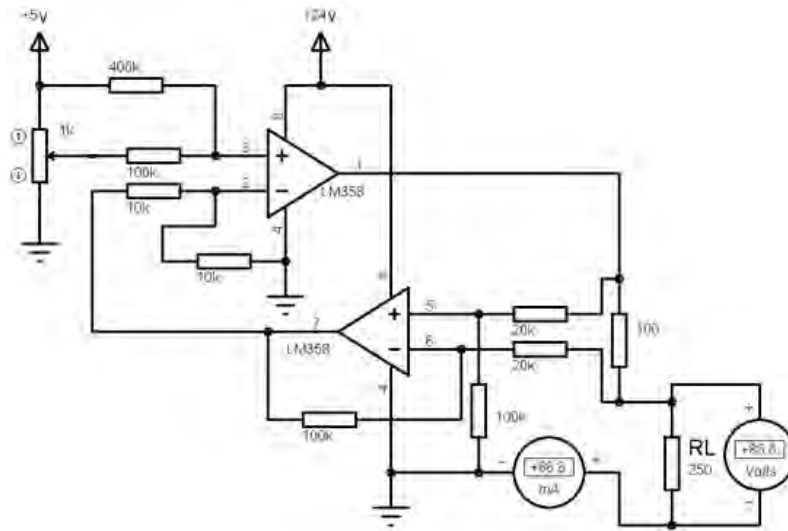


Fig. 20, Diagrama Esquemático Transmisor 4-20mA.

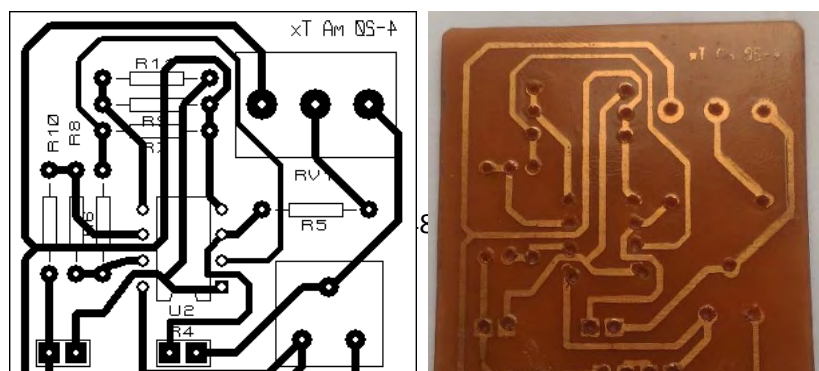


Fig. 21, Diagrama PCB Transmisor 4-20 mA.



Fig. 22, Transmisor 4-20mA ensamblado.

5.2.3 Receptor de Corriente de 4-20 mA.

Durante el proceso de adecuación de señal es fundamental la conversión de la señal de 4-20mA que entregan los sensores industriales o en su defecto el circuito simulador, en una señal de voltaje que pueda ser procesada por el DAQ.

El lazo de corriente de 4-20mA es una señal industrial, la cual necesita ser acondicionada adecuadamente, en este caso se ha montado un operacional con

un offset de 1V y realimentado en corriente que hace la conversión, si no se pone el offset para los 4mA se pierde un 20% de resolución del DAC. Cuando la señal va a campo y está sometida a interferencias, se recomienda una alimentación de 24Vdc, la cual es un estándar industrial, por eso generalmente los lazos de corriente de 4-20mA se alimentan a 24V.

Circuito conversor de 4-20mA a 5-10 vdc:

Se desarrolló dentro de la etapa de acondicionamiento de señal, un circuito que es capaz de convertir una señal de entrada de corriente de 4 a 20 mA en una señal de salida de voltaje de 0 a 5 Vdc, con una correspondencia lineal.

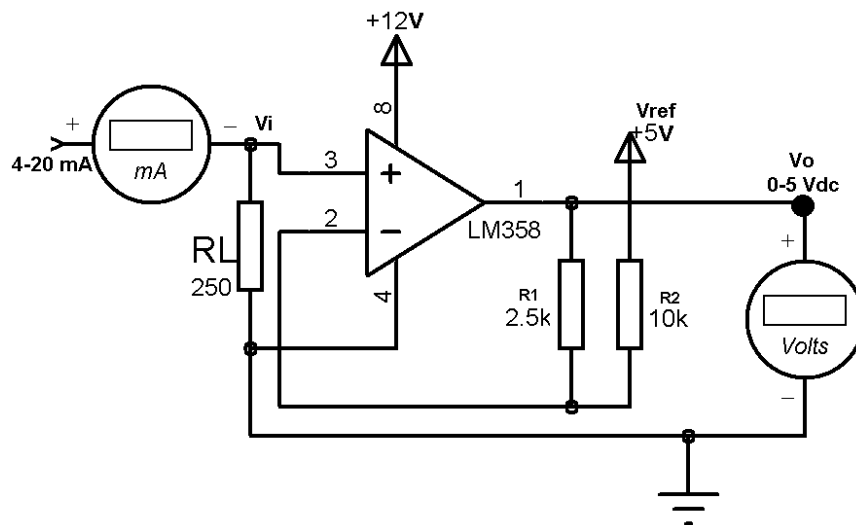


Fig. 23, Diagrama esquemático del convertidor 4-20mA.

La señal de corriente de entrada 4-20mA recorre la resistencia de carga ($R_L=250\Omega$), generando una señal de voltaje (V_i) a la entrada del amplificador operacional LM358, el cual está configurado como amplificador diferencial y mediante la realimentación que genera el voltaje de referencia (V_{ref}) y las resistencias R_1 y R_2 , genera un voltaje de salida (V_o) entre 0-5Vdc.

5.2.5 Etapa de Adecuación de Señal.

Una vez descrito el diseño y funcionamiento de cada uno de los módulos que comprenden la etapa de adquisición y acondicionamiento de señal:

- Transmisor 4-20mA.
- Receptor y conversor de 4-20mA a 0-5Vdc.
- Amplificador de Instrumentación.

A continuación se presenta el diseño electrónico esquemático, integrando los tres módulos en la tarjeta de adquisición de señal:

En la Figura 23 se puede observar el diagrama esquemático de la etapa completa de adecuación de señal, en donde para la señal que entregan los sensores industriales: S1 (Torque) y S2 (Nivel), se implementa un conversor de señal de 4-20mA a 0-5 vdc y para la señal del sensor industrial S3 (Flujo) se implementa un seguidor de voltaje debido a la característica de la señal que entrega (0.45v → 3v).

A la salida de cada conversor (V_{s1} , V_{s2} , V_{s3}), se ubica un arreglo que consiste en un diodo zener 1N4733A (5.1v) configurado como regulador de voltaje y una resistencia limitadora de corriente de $1K\Omega$, el arreglo está configurado para proteger la etapa siguiente (microcontrolador) de picos de voltaje que puedan sobrepasar los 5.1v y corrientes mayores a las nominales del microcontrolador que son de 200mA.

También se ubica un condensador de $10\mu F$ el cual ayuda a filtrar un posible ruido de baja frecuencia y suavizar la señal de entrada al microcontrolador.

De esta manera las tres señales (V_{s1} , V_{s2} , V_{s3}), ya filtradas y acondicionadas se interconectan al microcontrolador mediante 3 de sus entradas análogas, la etapa siguiente es la de procesamiento y envío de los datos la cual se describe en la siguiente sección.

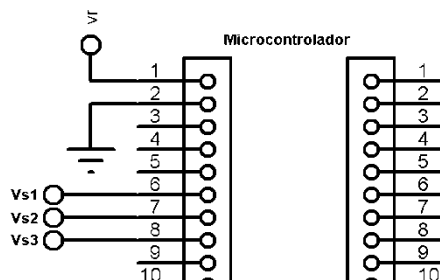
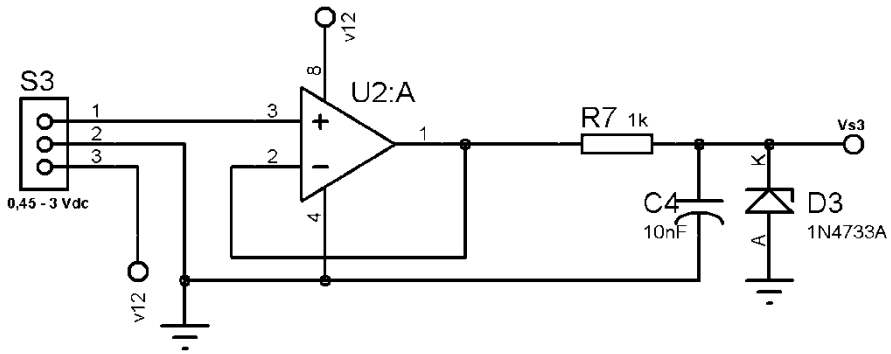
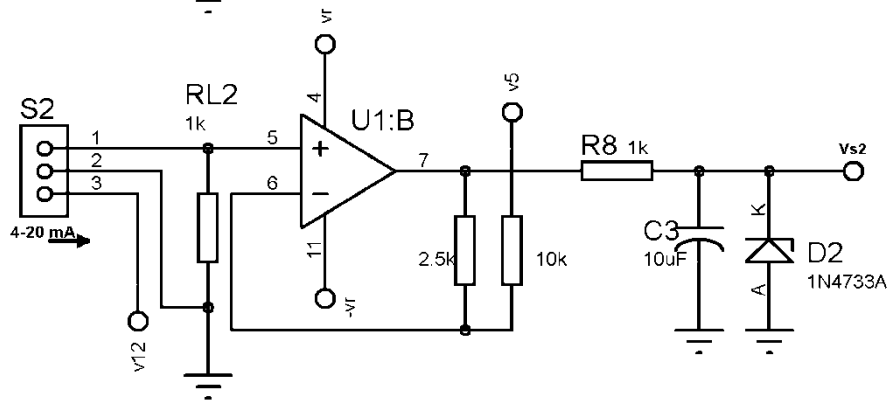
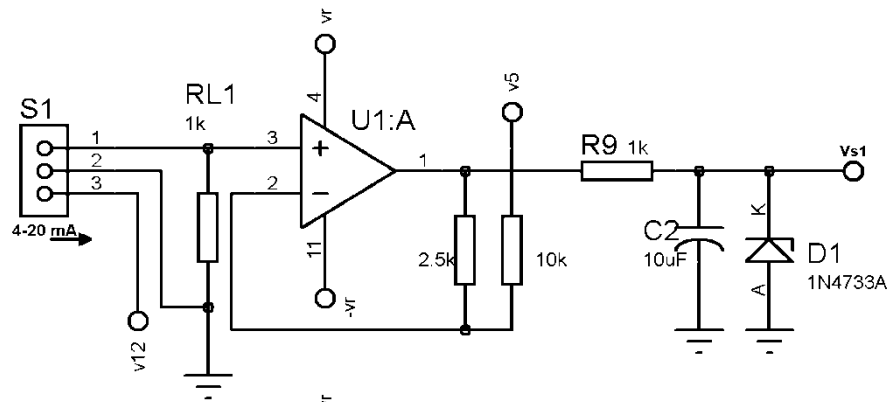


Fig. 24, Diagrama Esquemático de la Etapa de Adecuación de Señal.

5.2.6 Tratamiento Del Ruido.

Una práctica habitual en el acondicionamiento de la señal es el aislamiento eléctrico entre el transductor y el PC, para proteger al mismo de señales transitorias de alta tensión que puedan dañarlo, el motivo para usar el aislamiento es garantizar que las lecturas del transductor no son afectadas por diferencias en el potencial de masa o por tensiones en modo común lo que se conoce como ruido eléctrico, ya en la práctica real, durante el proceso de perforación, el sistema estará expuesto constantemente a condiciones de alto ruido y perturbaciones eléctricas en el ambiente de trabajo debido a la cercanía de cargas inductivas y maquinaria industrial trabajando sobre la misma red eléctrica.

Para realizar el debido tratamiento al ruido se utilizan barreras aisladas (Isolated Barriers) del tipo *KFD2-CD-Ex1.32* Marca *Pepperl+Fuchs GmbH*, los cuales son aisladores eléctricos protectores que limitan la cantidad de energía (corriente y voltaje) que entra desde un área peligrosa ante una eventual falla, por ejemplo una sobrecarga, corto circuito y ruido excesivo, en la práctica éstos equipos son requeridos únicamente para los sensores que entregan una señal de corriente (Lazo de corriente de 4-20mA), para los sensores que entregan señal de voltaje se ha evidenciado que no se presentan perturbaciones de ruido significativas ya que para las señales cuasi-continuas, (como la de nivel) se usa el estándar de 24v de alimentación y un filtro de ruido de baja frecuencia (unos 4 Hz) incluido en el sensor o en la etapa de acondicionamiento de señal, que elimina interferencias, incluidos los 50/60 Hz de la red eléctrica.

El dispositivo se utiliza en tecnología de control e instrumentación (tecnología C&I) para el aislamiento galvánico de señales como señales estándares de 20 mA y 10 V o, de forma alternativa, para adaptar o estandarizar señales, el dispositivo está diseñado para su instalación en un carril de montaje DIN de 35 mm conforme a EN 60715, es un aparato asociado, conforme a IEC/EN 60079-11 y es un aparato eléctrico para zonas peligrosas de tipo zona 2.



- Barrera aislada de 1 canal
- Alimentación de 24 V CC
- Entrada de corriente y tensión
- Salida de corriente o tensión
- Entrada/salida configurada en fábrica
- Precisión 0,1 %

Fig. 25, Barrera Aislada *KFD2-CD-Ex1.32*.

5.3 PROCESAMIENTO Y ENVIO DE LA INFORMACIÓN:

El microprocesador es el circuito integrado central de un sistema informático, es el encargado de ejecutar las instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas, lógicas y accesos a memoria, éstos han evolucionado tecnológicamente a una escala muy grande y actualmente existen procesadores muy potentes los cuales son utilizados en los computadores de última generación.

Para las tareas de procesamiento de datos que nos requiere el desarrollo de éste proyecto, tenemos la posibilidad de utilizar un microcontrolador, la diferencia principal entre un microcontrolador y un microprocesador son las capacidades de entradas y salidas, así como el rendimiento de la CPU.

Un microcontrolador (μ C, UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria, está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica, un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

5.3.1 Arduino Leonardo Board.

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo (IDE), diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Una tarjeta de desarrollo de Arduino se puede utilizar para desarrollar elementos electrónicos autónomos, conectarse a dispositivos e interactuar tanto con el hardware como con el software, sirve tanto para controlar un elemento o bien para leer la información de una fuente, como puede ser un sensor y convertir la información en una acción.

Arduino dispone de una amplia variedad de placas (boards) para usar dependiendo de las necesidades de hardware, la placa de desarrollo que más se ajusta a los requerimientos evidenciados para el desarrollo del sistema de adquisición de datos propuesto en este proyecto es el *Arduino Leonardo*.

El Arduino Leonardo está basado en un microcontrolador ATmega32u4, cuenta con 20 pines de entrada/salida digitales, de los cuales, 7 están configurados como salidas PWM (Pulse-Width Modulation) y 12 como entradas analógicas, además cuenta con un oscilador de cristal que funciona a 16 MHz, una conexión micro USB, un conector de alimentación 7-12V, una cabecera de ICSP (In-Circuit Serial Programming) y, un botón de reinicio.

Una de las novedades que incorpora el Leonardo, es que se diferencia de las placas anteriores en que el ATmega32u4 incorpora comunicación USB.



Fig. 26, Arduino Leonardo Board.

Microcontroller	ATmega32u4
Voltaje de Operación	5V
Voltaje de Entrada (recomendado)	7-12V
Voltaje de Entrada (límites)	6-20V
Pines Digitales I/O	20
Canales PWM	7
Canales Analógicos	12
Corriente DC I/O por pin	40 mA
Corriente DC para pin 3.3v	50 mA
Memoria Flash	32 KB (ATmega32u4) of which 4 KB used by bootloader
SRAM	2.5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad del Reloj	16 MHz

Tabla 5. Especificaciones Técnicas del Arduino Leonardo Board.

La placa Arduino Leonardo, puede ser alimentada a través de la conexión micro USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente, para usar alimentación externa (no mediante USB), ésta puede venir de un adaptador de CA o de una batería, el adaptador lo podemos conectar mediante un Jack de 2.1mm con centro positivo a la toma de alimentación de la placa, puede operar mediante una alimentación externa de 6 a 20v, el rango recomendado de alimentación es de 7 a 12 Voltios.

La descripción de pines y el diagrama esquemático de la placa Arduino Leonardo se especifica en los Anexos x y x

5.4 RECEPCIÓN Y MONITOREO DE LA INFORMACION EN EL PC.

Hasta esta etapa la señal de los tres sensores ya ha sido medida, acondicionada, digitalizada y enviada al PC por parte del módulo de transmisión de datos del DAQ, de ahora en adelante se describirá el proceso que realiza el software de aplicación, el cual se encarga de establecer la comunicación con el instrumento, procesar los datos recibidos, generar las gráficas de monitoreo en tiempo real y generar el log de perforación.

Para desarrollar el software de aplicación se utilizó el lenguaje de programación C Sharp (C#), descrito en la sección 4.5.6, a continuación se describen los diferentes módulos que hacen parte del software de aplicación y las funciones que realizan.

5.4.1 Transmisión serial en C Sharp (C#).

Como ya se especificó anteriormente, para el desarrollo del proyecto se escogió el software de Microsoft Visual C# Express 2008, el objetivo principal consiste en crear una aplicación de Windows para poder utilizar la información recibida y/o enviada desde el instrumento, en este caso para procesar los datos recibidos desde el DAQ mediante el puerto serial.

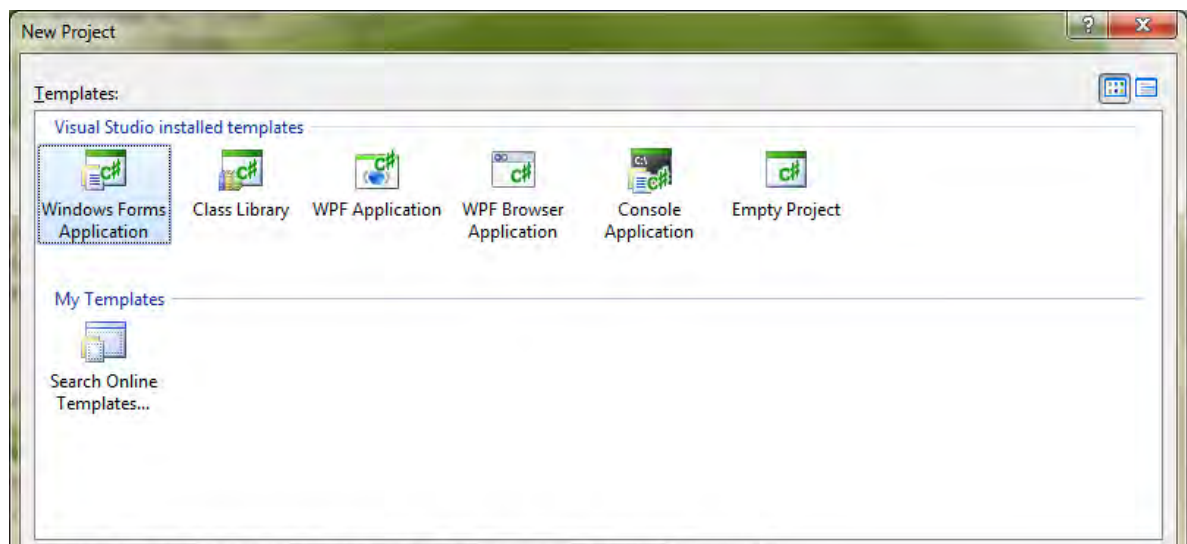


Fig. 27, Instancia Principal de Visual C# Express 2008.

Inicialmente se abre un nuevo proyecto de Microsoft Visual C# Express 2008 y en la instancia principal de Visual C# se escoge la opción de *Windows Form Application* como se observa en la figura 28, el nuevo proyecto permite acceder a los objetos que brinda C# para el desarrollo de aplicaciones, uno de los más importantes es el componente *SerialPort* el cual permite configurar el puerto serial virtual por medio del cual se realiza la comunicación con el instrumento.

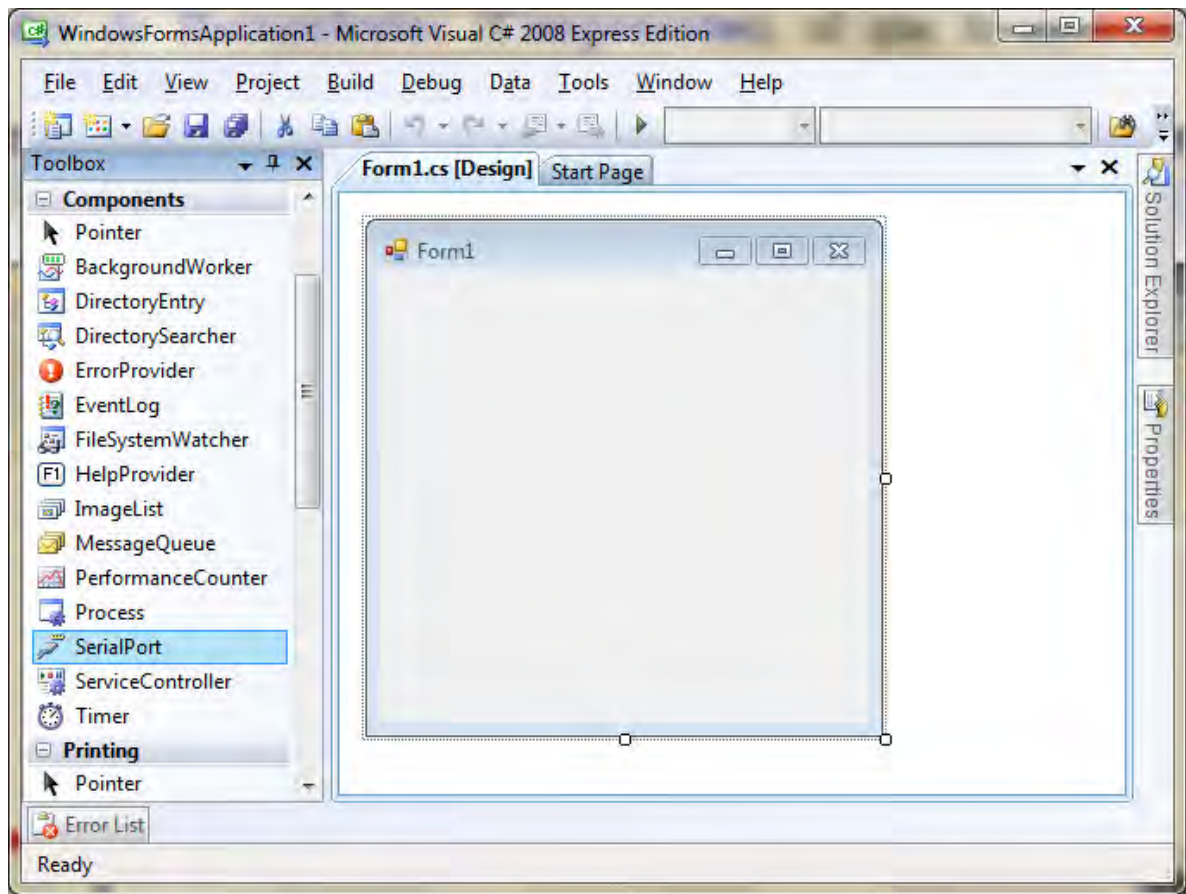


Fig. 28, Componente SerialPort en Visual C#.

El componente *SerialPort* requiere para su utilización, la configuración de sus propiedades principales, las cuales regulan el tipo de comunicación serial a realizar, entre ellas están:

Name: Nombre del componente.

BaudRate: Velocidad de transmisión de datos o bit rate en bps.

PortName: Nombre del puerto serial, aparece por defecto COM1

DataBits: Numero de bits en una trama de datos.

StopBits: Numero de bits de parada.

DtrEnable: Habilita la línea DTR (Data Terminal Ready) durante la transmisión.

La estructura y descripción del programa desarrollado en C# se analizan en la sección de resultados y están descritos en el Anexo x (Código del software de aplicación desarrollado en lenguaje C#).

5.4.2 Diseño del Software de Aplicación LoggingDAQ

El software de aplicación que se diseñó en lenguaje C#, se lo denominó como: 'LoggingDAQ' y tiene como funciones principales:

- Realizar la comunicación con el instrumento DAQ.
- Procesar la información para realizar el monitoreo de las variables en tiempo real.
- Guardar los datos para generar un reporte de la sesión de perforación.
- Generar reporte imprimible exportado a MS EXCEL.

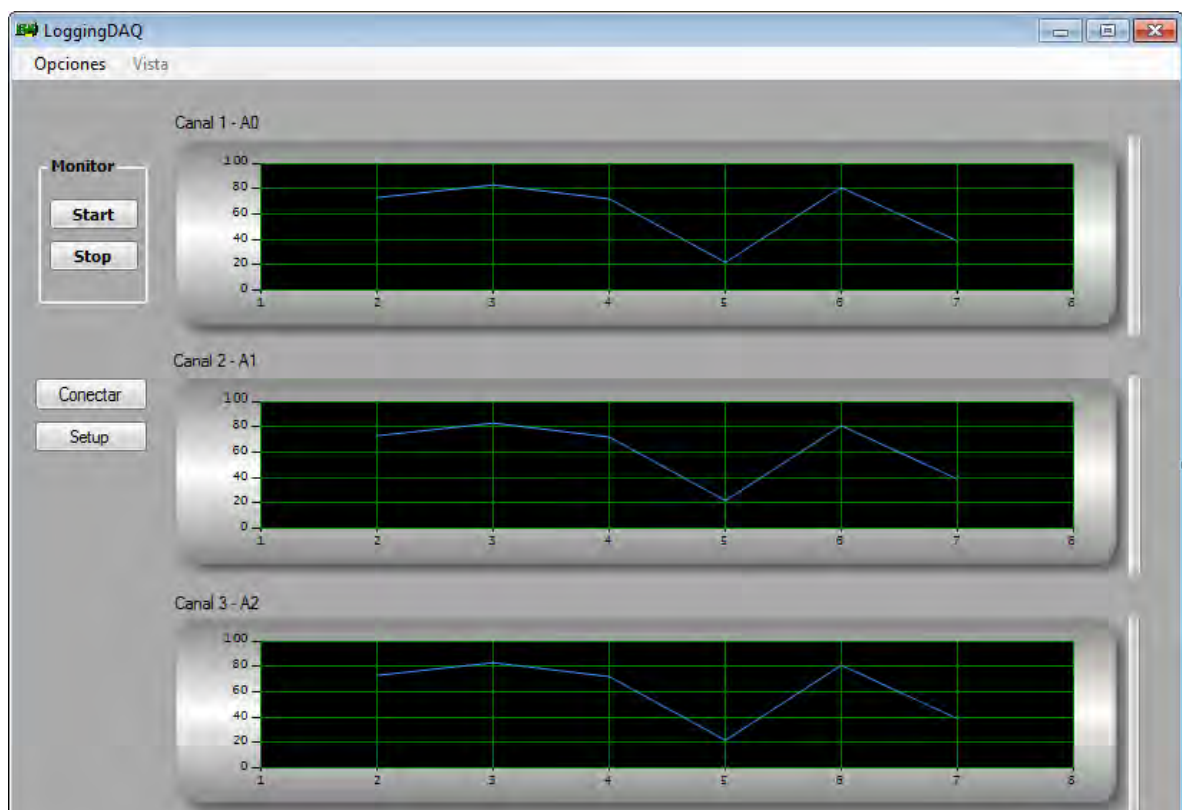


Fig. 29, Pantalla Principal del Software LoggingDAQ.

Como se observa en la figura 29 el software tiene 3 canales para el monitoreo de las variables en tiempo real, permite intercambiar entre dos tipos de visualización: Numérica o RTC (Realtime Chart) y tiene un módulo '*Setup*' para la configuración del puerto serial y la habilitación de los canales analógicos.

La estructura y descripción de los módulos del software de aplicación, se analizan en la sección de resultados y están descritos en el Anexo x (Código del software de aplicación desarrollado en lenguaje C#).

6. RESULTADOS

Para evaluar los resultados del proyecto, tenemos en cuenta la formulación del problema y los objetivos planteados inicialmente, a continuación se describen los resultados obtenidos con el desarrollo del proyecto.

6.1 TARJETA DE ADQUISICIÓN DE DATOS DAQ.

Se desarrolló una tarjeta de adquisición de datos con una fuente de poder integrada (dual $\pm 12V$), 3 canales de entrada analógicos y un puerto USB para comunicación con el computador.

6.1.1 Diagrama Esquemático del Circuito Electrónico.

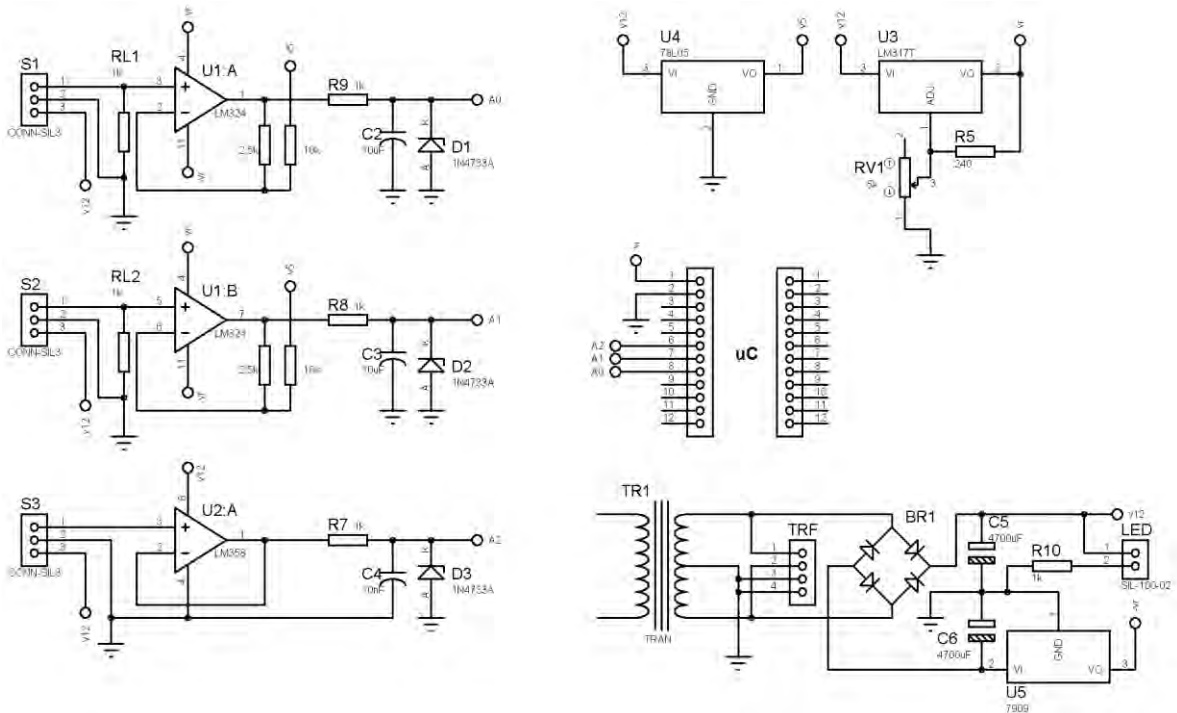


Fig. 30, Diagrama Esquemático de la Tarjeta de Adquisición de datos DAQ.

En el diagrama electrónico se puede observar las entradas S1, S2, S3 correspondientes a los 3 sensores industriales, también se observa la fuente de poder dual integrada de $\pm 12V$ la cual entrega los voltajes de referencia V_{12} , V_5 , V_r y $-V_r$ y el socket μC para el microcontrolador, desde donde se desprende el puerto USB para la transmisión de datos con el computador.

6.1.2 Diagrama de Circuito Impreso PCB.

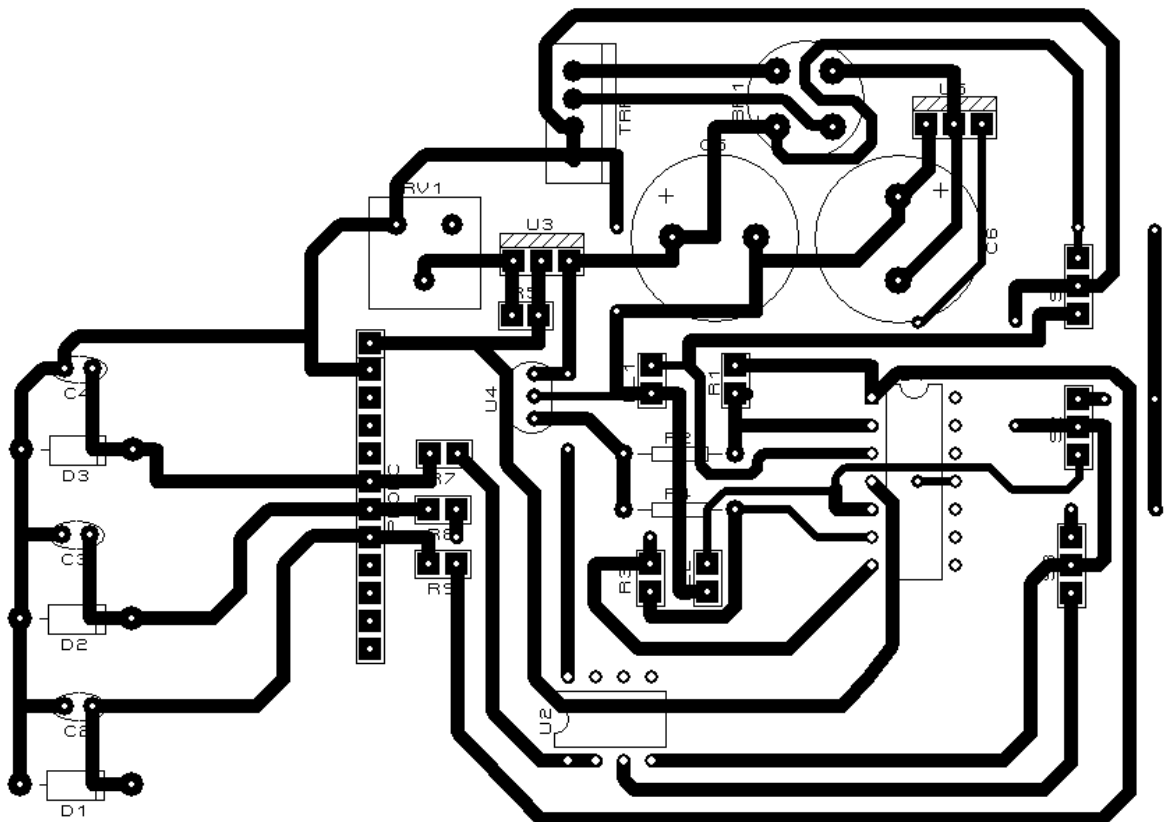


Fig. 31, Diagrama del circuito PCB.

En la figura 31 se puede observar el diseño del PCB (Printed Circuit Board), realizado en el software de simulación ARES de Labcenter Electronics, en donde la capa superior (Top Silk) describe los componentes electrónicos utilizados y la capa inferior (Bottom Copper) define los buses y rutas de interconexión entre los elementos.

El tamaño aproximado de la placa es de 10x12cms y el revelado se hizo utilizando la técnica de plotter de corte con base de vinilo adhesivo.

6.1.3 Montaje del Circuito.

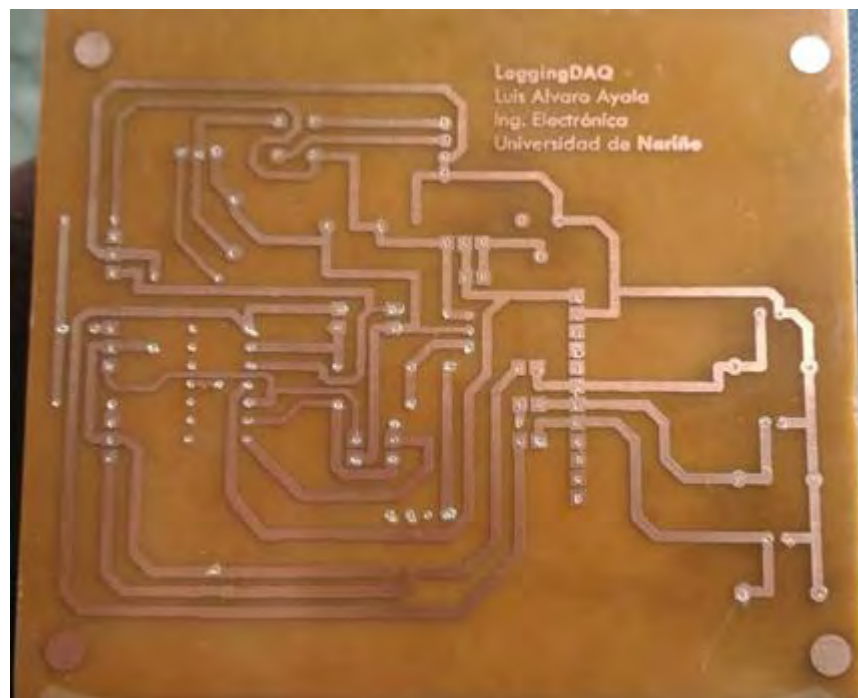


Fig. 32, Placa de Circuito Impreso Revelada.

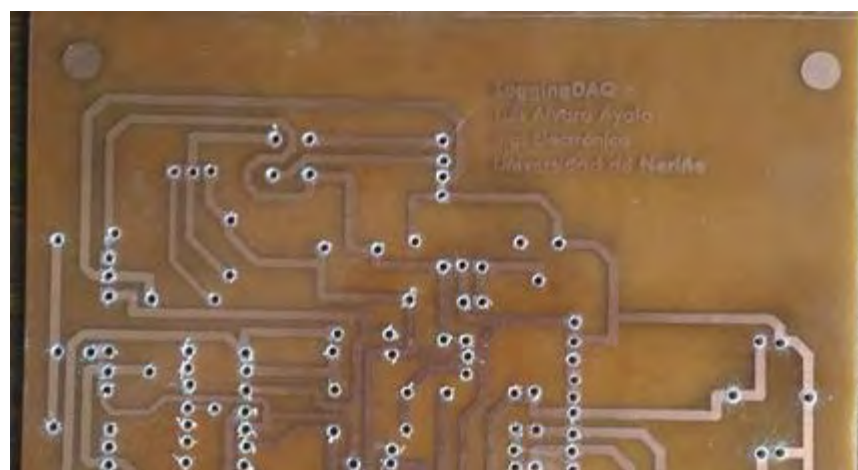


Fig. 33, Placa de Circuito Impreso Perforada.



Fig. 34, Tarjeta ensamblada y montada sobre la carcasa.



Fig. 35, Ensamble final del DAQ.

6.2 SOFTWARE DE APLICACIÓN 'LoggingDAQ'.

A continuación se describe el resultado del diseño del software de aplicación el cual se denominó como *LoggingDAQ* y las características de cada uno de sus módulos:

6.2.1 Módulo de Inicio.

Este es el módulo que se presenta al iniciar el programa, en donde se puede observar que solo hay dos opción habilitadas, la primera es el botón 'Setup' ya que antes de iniciar el proceso de monitoreo se debe configurar las opciones del puerto de comunicación, la cantidad de canales a utilizar y las opciones de alarma que tiene cada canal, la segunda opción es el botón 'Salir', el cual permite cerrar la aplicación.

También está habilitado el menú 'Opciones' donde se brinda información acerca del programa y calibración del instrumento

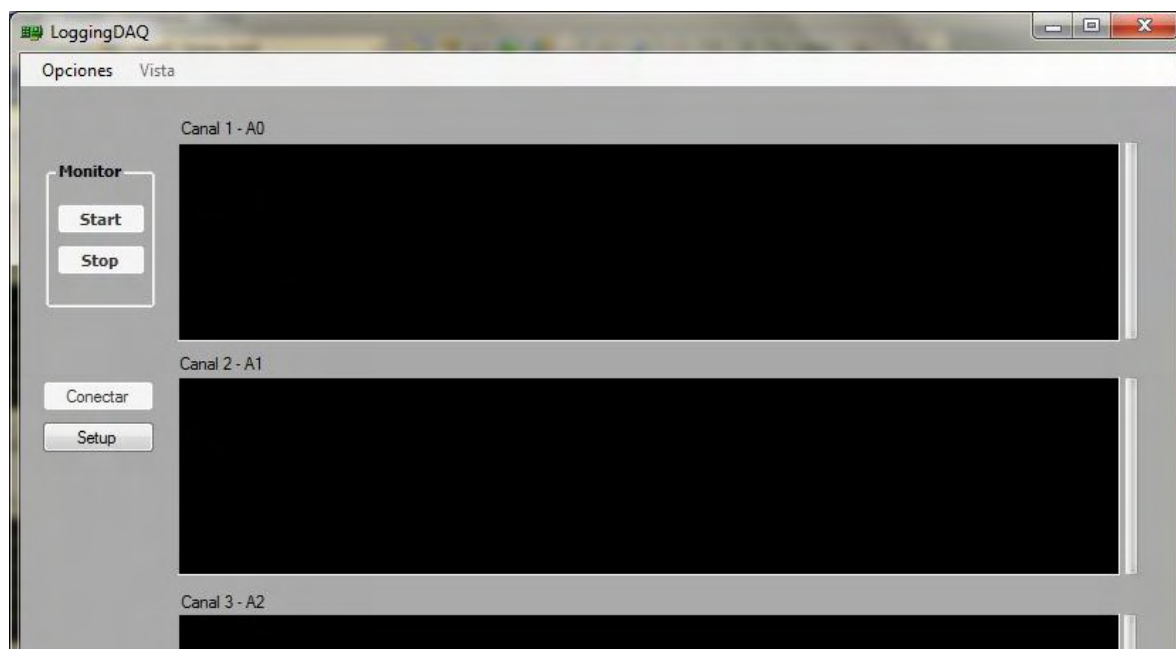


Fig. 36, Módulo de inicio del software LoggingDAQ.

6.2.2 Módulo de Configuración.

El módulo de configuración del software de aplicación 'LoggingDAQ' se compone de 3 bloques funcionales, los cuales se encargan de configurar las opciones de funcionamiento que presenta el instrumento.

Al módulo de configuración se accede presionando el botón 'Setup' desde la pantalla de inicio, permite configurar las opciones principales del puerto de comunicación, también permite habilitar/deshabilitar los canales analógicos que se van a usar durante la siguiente transmisión y permite habilitar y configurar los valores de las niveles de alarma alto y bajo correspondientes a cada canal analógico.

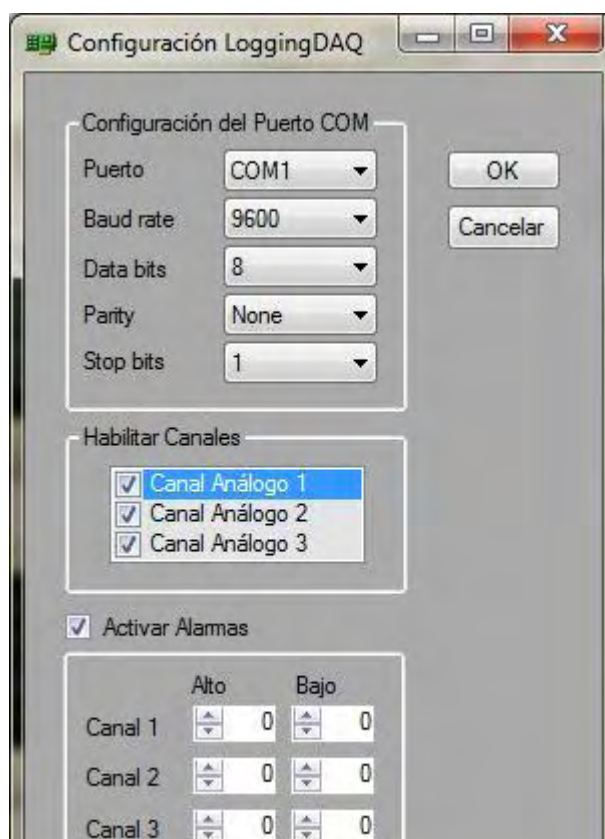


Fig. 37, Módulo de configuración del software LoggingDAQ.

6.2.3 Módulo de Monitoreo.

Una vez se han configurado las opciones de monitoreo del instrumento, la aplicación regresa a la pantalla principal (Fig.x), en donde ya se ha habilitado el botón 'Conectar' el cual establece la comunicación entre el PC y el instrumento.

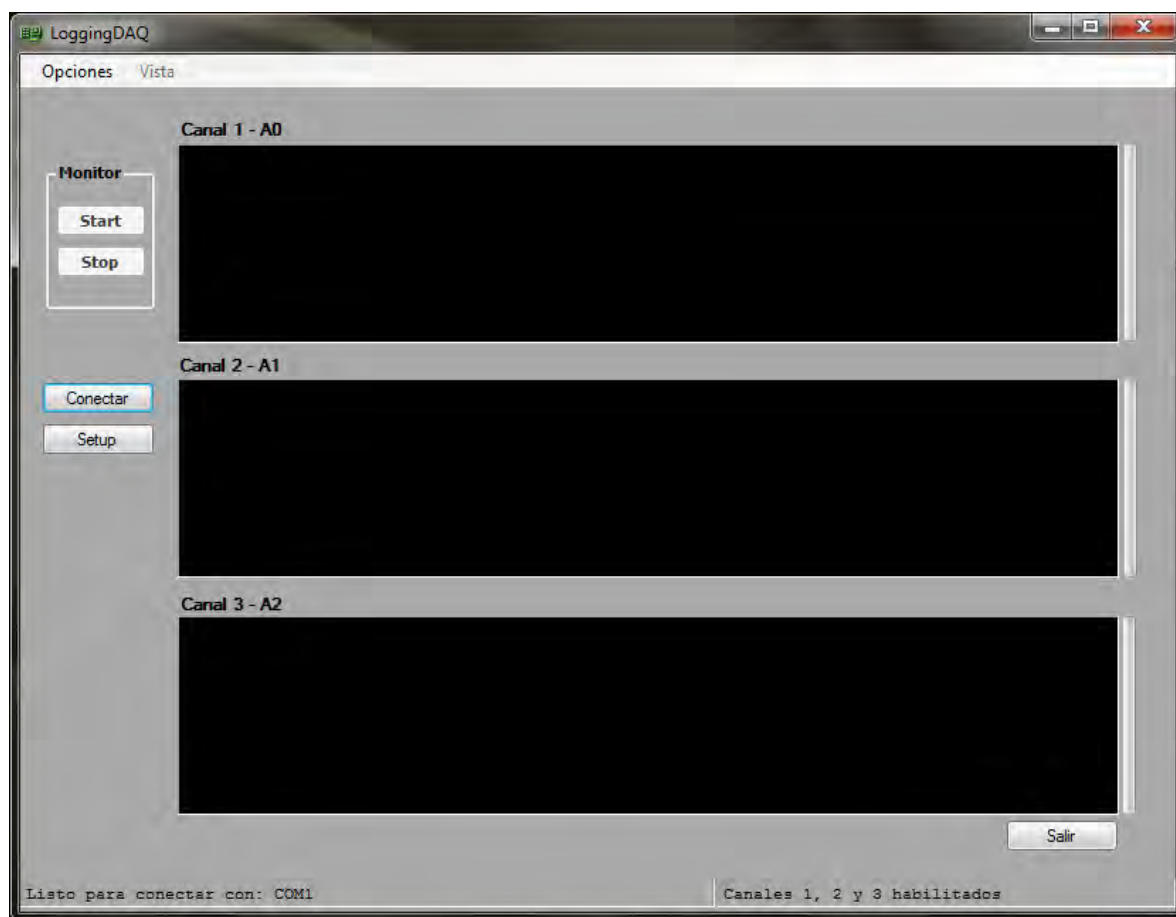


Fig. 38, Pantalla principal del software LoggingDAQ.

Se diseñó una terminal de información ubicada en la parte inferior de la pantalla de la aplicación en la cual se describe la información relevante correspondiente a cada estado del proceso, en este caso se indica el puerto por el cual se va a realizar la comunicación y los canales habilitados para la transmisión.

Al hacer click en el botón 'conectar', se establece una conexión con el instrumento, se habilita el botón 'Start' y se habilita el botón 'Desconectar', en la terminal se registra la información más relevante correspondiente a esta etapa del proceso: el estado de la conexión y los parámetros de configuración de la transmisión (Fig. x).

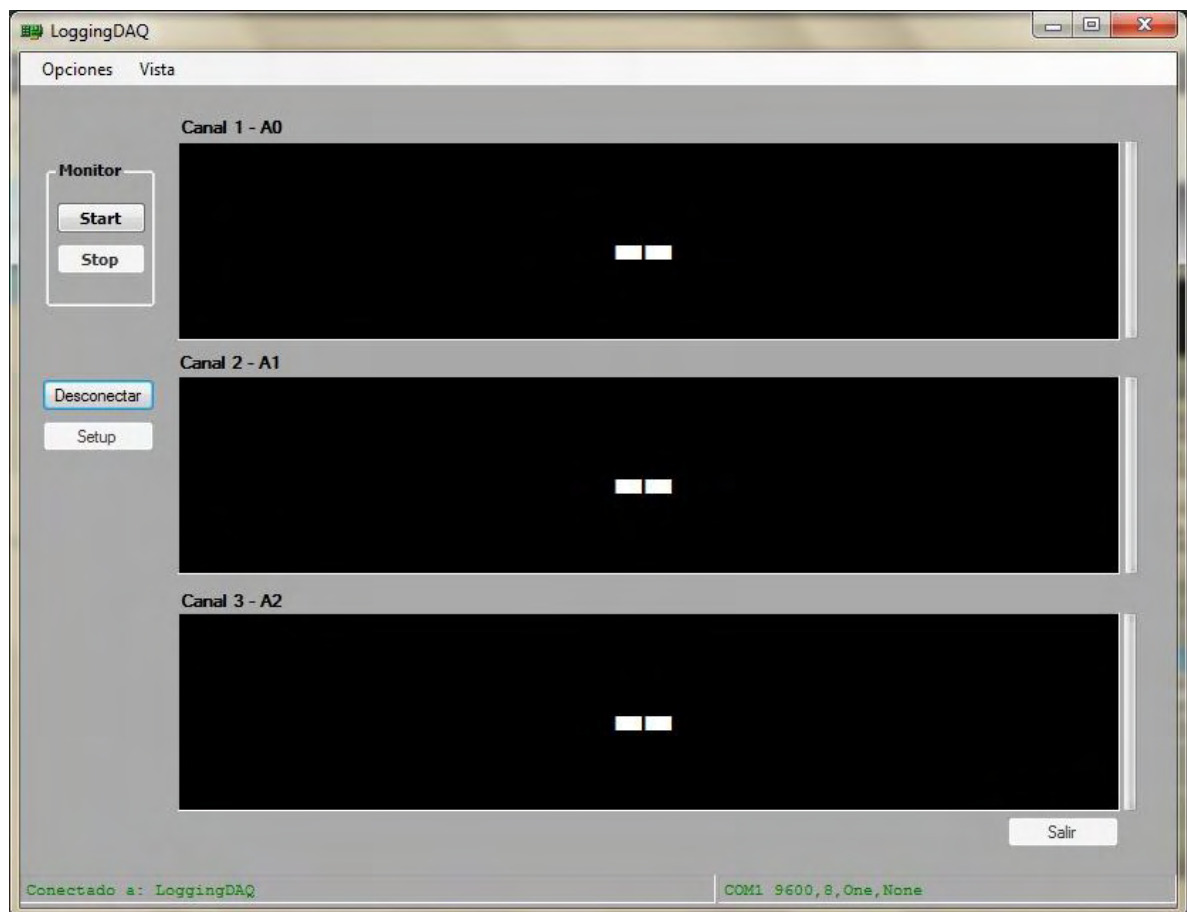


Fig. 39, Pantalla principal en estado: 'Conectado'.

Al hacer click en el botón 'Start' se inicia la sesión de monitoreo y el programa empieza a registrar los valores transmitidos por el instrumento, correspondientes a cada canal habilitado (Fig.x), también se habilita el botón de 'Stop' y en el menú 'Vista' se puede intercambiar entre los tipos de visualización: numérico o gráfico en tiempo real.

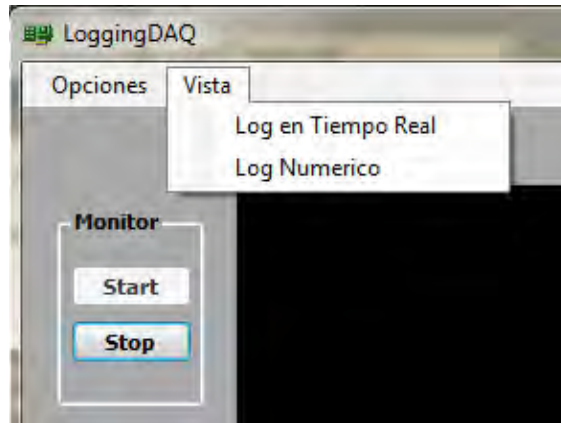


Fig. 40, Menú Vista.

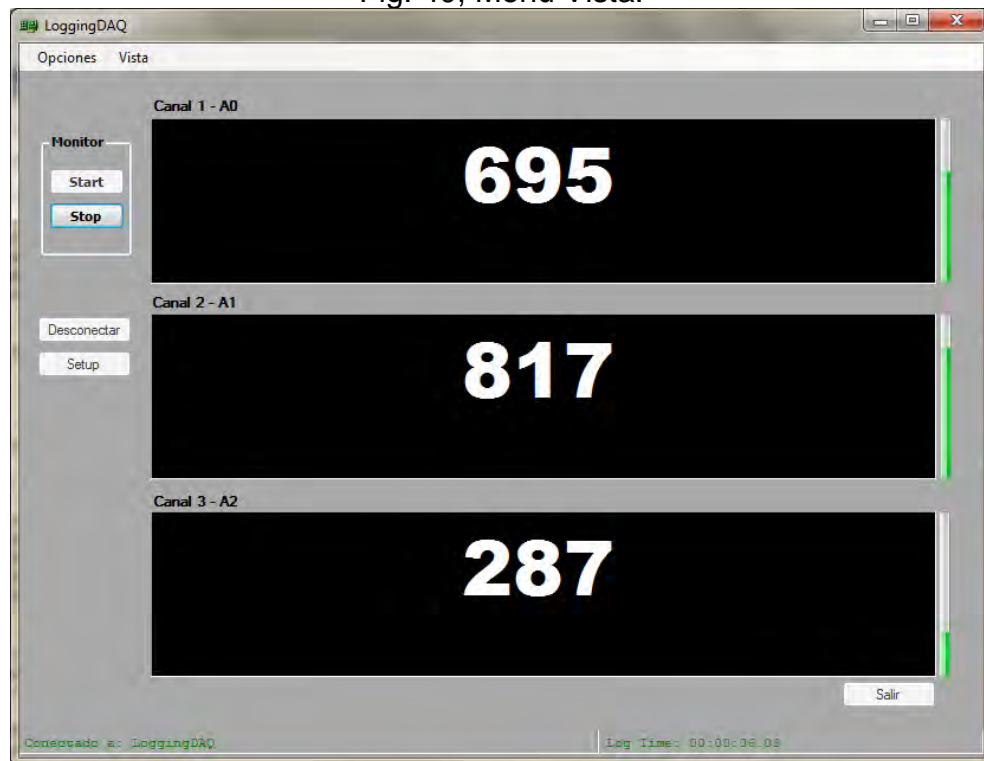


Fig. 41, Modo de visualización numérico.

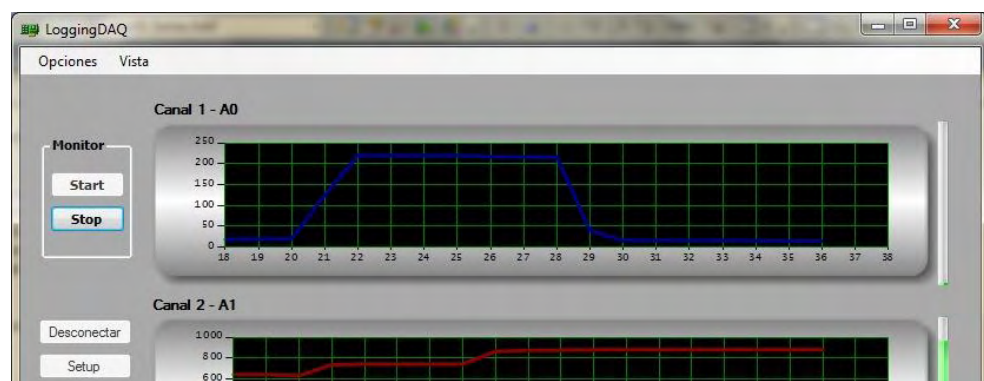


Fig. 42, Modo de visualización gráfico en tiempo real.
En las figuras x y x se observa cómo se registran los valores entregados por los sensores habilitados para el monitoreo, se puede observar también una barra de nivel ubicada al lado derecho de cada chart, la barra de nivel se programó para seguir el valor registrado en un rango establecido en el módulo de calibración.

Para terminar la sesión de monitoreo se presiona el botón 'Stop', enseguida el programa procede a guardar un log en una dirección predeterminada nombrado con la hora y fecha del evento y su respectiva duración.

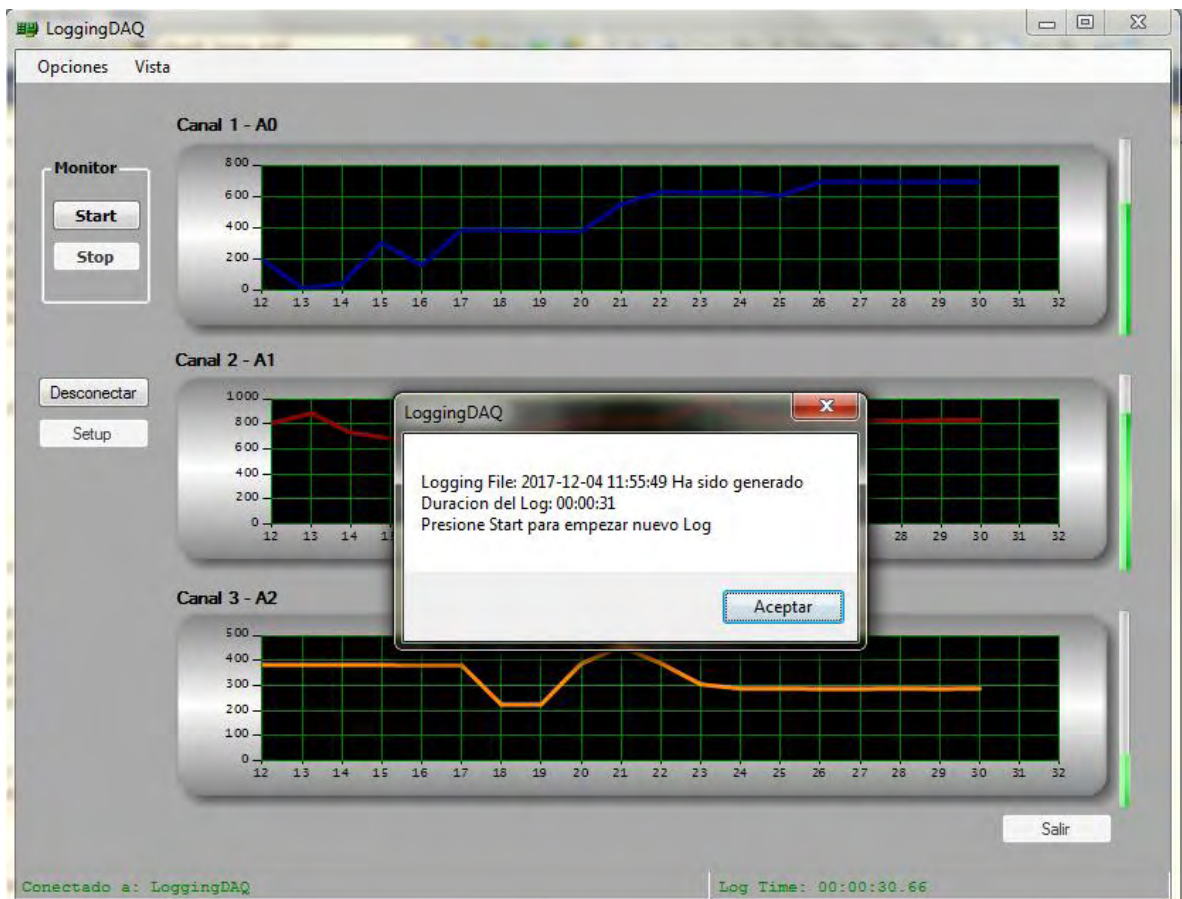


Fig. 43, Notificación de fin se sesión de monitoreo actual.

6.2.4 Módulo de Generación de Reportes.

Al finalizar una sesión de monitoreo el programa ofrece la posibilidad de generar un log para imprimir, se programó la opción de generar un log impreso haciendo uso de MS Excel y una macro estandariza el archivo y lo imprime en formato .pdf

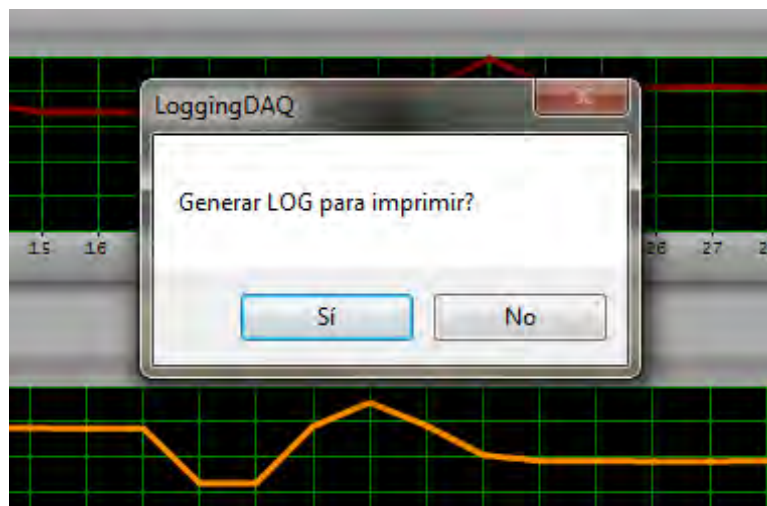


Fig. 44, Menú generar log para impresión.

Al hacer click en el botón 'Si' el programa genera el archivo gráfico tabulado en Excel con los valores correspondientes a la sesión inmediatamente anterior.

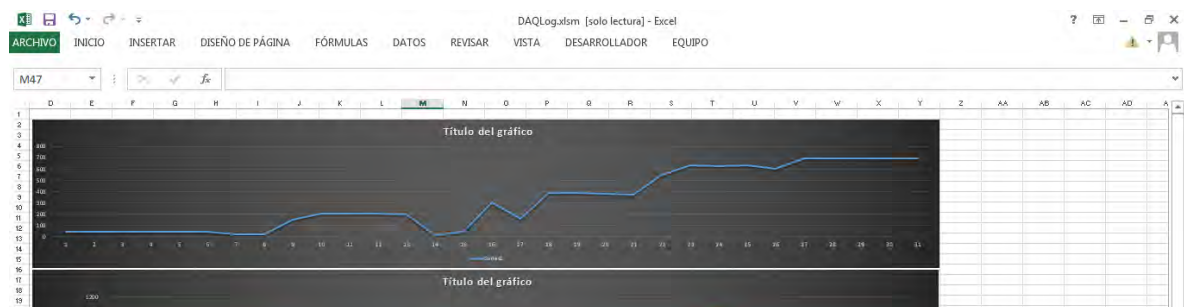


Fig. 45, Reporte generado por el aplicativo en MS Excel.

CONCLUSIONES

Se desarrolló el prototipo de un sistema de adquisición de datos para una estación de Mud Logging, compuesto por un DAQ de 3 canales analógicos cuyas características se establecieron inicialmente para cumplir con los requerimientos técnicos de 3 sensores industriales predefinidos durante el estudio del caso, a saber un sensor de nivel de tanque, un sensor de torque de taladro de perforación y un sensor de flujo de lodo.

Acorde a las características técnicas de los sensores escogidos, se desarrolló una tarjeta de adquisición de datos basada en el microcontrolador ATMEL ATmega32u4, montado sobre una placa de desarrollo Arduino Leonardo, se estableció un sistema de comunicación USB entre el instrumento y el PC.

Se implementó un software de aplicación desarrollado en un ambiente de programación orientado a objetos en lenguaje C# para realizar el control, monitoreo y registro de la información generada durante una sesión de monitoreo. El realizar el control del proceso haciendo uso de un software de aplicación diseñado y basado en los parámetros físicos establecidos inicialmente, nos permite alcanzar el objetivo del proyecto de diseñar un instrumento optimizado para el proceso de Mud Logging.

El resultado del desarrollo del proyecto es consecuente con la formulación del problema y los objetivos específicos planteados, ya que se desarrolló un prototipo de instrumento de adquisición de datos (DAQ) y un software de aplicación, con un bajo costo de implementación y con características dedicadas al proceso de adquisición y monitoreo de los parámetros requeridos durante el proceso de perforación en una estación de Mud Logging.

RECOMENDACIONES

Este trabajo de investigación establece una base para dar a conocer las especificaciones técnicas básicas que intervienen en el proceso industrial de explotación de hidrocarburos, se espera que el desarrollo de este proyecto permita a futuros estudiantes investigadores, profundizar en el desarrollo de instrumentos que ofrezcan ventajas económicas y técnicas sobre la instrumentación comercial.

Con respecto al desarrollo del proyecto, se puede establecer como una recomendación la posibilidad de optimizar el diseño del software, ya que el campo de la programación ofrece un espectro grande de opciones que se pueden ir evidenciando a través de la realización de nuevas investigaciones que requieran el desarrollo de herramientas optimizadas para un proceso específico.

Con respecto al software LoggingDAQ se podrían por ejemplo programar mejoras en la arquitectura del software, como por ejemplo establecer una pantalla inicial con user y password, ampliar el número de canales de monitoreo, programar sesiones de monitoreo automáticas (timer), etc. como también establecer diferentes tipos de transmisión de datos para determinar la mejor opción para transmitir datos en un ambiente de trabajo específico.

El instrumento al ser un prototipo está desarrollado con material didáctico, carcasa de plástico y cables de conexión comunes, ya en la práctica se exige que el diseño de este tipo de instrumento, se debe hacer cumpliendo unas características

especiales que poseen los instrumentos industriales, tales como carcasa anti-explosión, cables blindados y barreras contra el ruido.

BIBLIOGRAFÍA

ARENY, Ramón Pallas. Sensores y Acondicionadores de Señal. 4ta Edición. Barcelona: Marcombo, 2003. 494p.

ATMEL, ATmega16U4/ATmega32U4: 8-bit Microcontroller DataSheet [en linea]. <http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf> [Citado el 12 de Junio de 2016]

BOYLESTAD, Robert. Electrónica: Teoría de Circuitos y dispositivos electrónicos. 8 Ed. México: Pearson Education, 2003. 517p.

DATALOG. Procedimientos y Operaciones en el Pozo. Versión 3.0. Calgary: 2002. 278p.

EVANS, Bryan W. Arduino Notebook: A Beginner's Reference. 1ra edición. San Francisco: Creative Commons, 2007. 45p.

GOMEZ, Jordi Marco. PRAT, Ángela Martín. Programación en C++ para ingenieros. 1ra Edición. Madrid: Paraninfo S.A., 2006. 471p.

INTERNATIONAL LOGGING, INC. Basic Mud Logging Manual, Versión 1.0. Singapur: 2001. 257p.

MURACH, Joel. *Murach's C# 2012*. EEUU: Mike Murach & Associates, INC. 2013. 872p.

NATIONAL OILWELL VARCO. *M/D Totco Technical Manuals* [en línea]. <<https://www.nov.com/WorkArea/DownloadAsset.aspx?id=19316>> [Citado el 3 de Mayo de 2016]

OLIVA RAMOS, Rubén. *Monitoreo, control y adquisición de datos con arduino y visual basic.net*. 1ra Edición. Alfaomega, 2017. 358p.

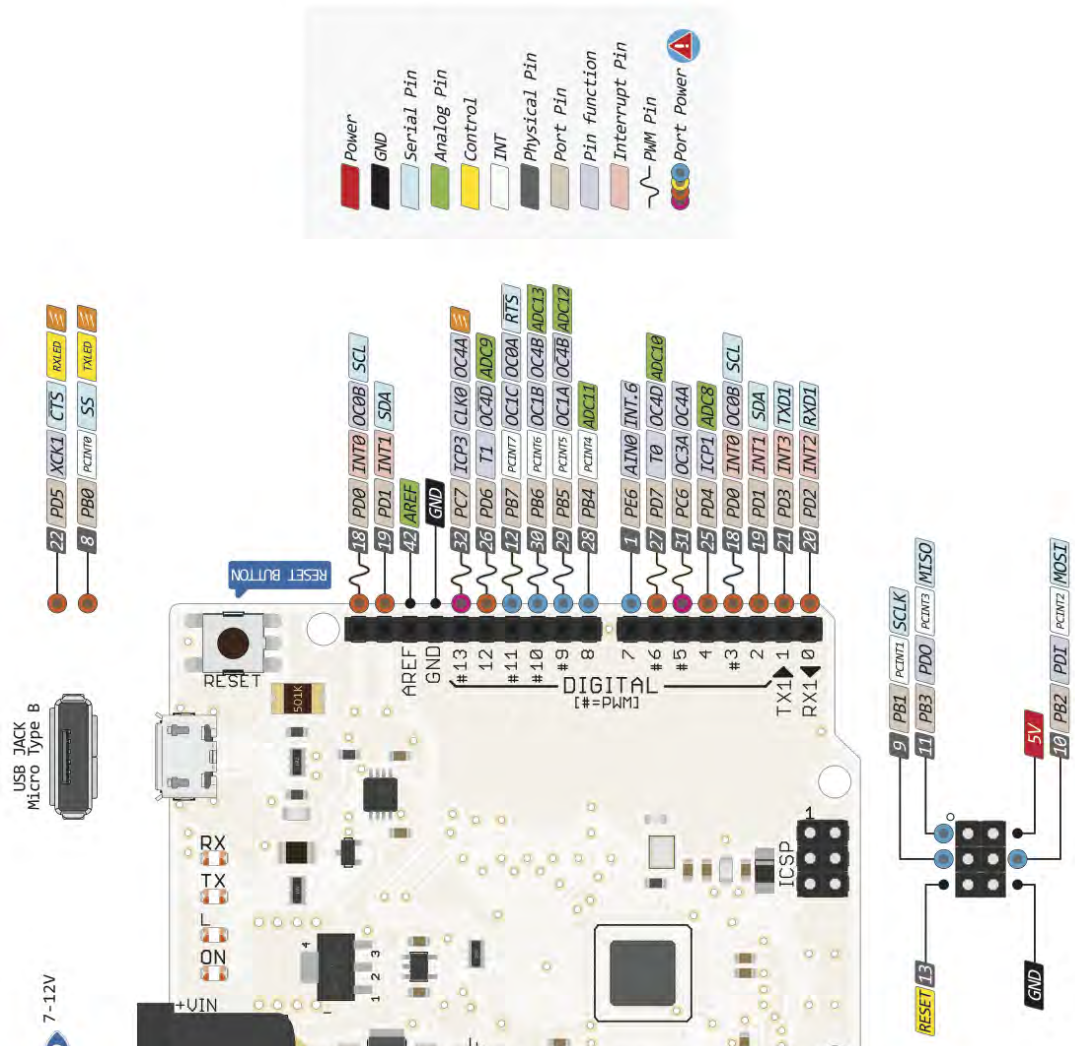
SEUBERT, Bernhard W. *The Wellsite Guide: An Introduction to Geological Wellsite Operations*. Jakarta: PetroPEP Nusantara, 2004. 138p.

USB IMPLEMENTERS' FORUM, *Universal Serial Bus Class Definitions for Communications Devices* [en línea]. <<http://nagasm.org/ASL/usb/usbcdc11.pdf>> [Citado el 15 de Junio de 2016]

WSC WELL CONTROL SCHOOL, *Drilling Well Control Course*, Harvey Louisiana: 70058 Aberdeen Drilling School, 2002. 577p.

ANEXOS

Anexo 1. Descripción de de la placa Arduino Leonardo.



Anexo 3. Código del programa desarrollado en C++ para el Arduino Leonardo.

```
int comando=0;
int sensor0=0;
unsigned long time = millis();

void setup() {
  Serial.begin(9600);
  SerialReceive();
}

void loop() {

  wait:
  if(millis()-time < 1000)
  {
    SerialReceive();
    goto wait;
  } time = millis();

  switch (comando) {
    case 10:
      sensor0 = analogRead(A0);
      Serial.println(sensor0, DEC);
      SerialReceive();
      break;

    case 20:
```

```

    sensor0 = analogRead(A1);
    Serial.println(sensor0, DEC);
    SerialReceive();
    break;

case 30:
    sensor0 = analogRead(A2);
    Serial.println(sensor0, DEC);
    SerialReceive();
    break;

case 40:
    sensor0 = analogRead(A0);
    Serial.println(sensor0, DEC);
    pasoA1:
    SerialReceive();
    if (comando=41)
    {
        delay(10);
        sensor0 = analogRead(A1);
        Serial.println(sensor0, DEC);
    }
    else
    {
        goto pasoA1;
    }
    SerialReceive();
    break;

case 50:
    sensor0 = analogRead(A0);
    Serial.println(sensor0, DEC);
    pasoA2:
    SerialReceive();
    if (comando=51)
    {
        delay(10);
        sensor0 = analogRead(A2);
        Serial.println(sensor0, DEC);
    }
    else
    {
        goto pasoA2;
    }
    SerialReceive();
    break;

case 60:
    sensor0 = analogRead(A1);
    Serial.println(sensor0, DEC);
    pasoA3:
    SerialReceive();
    if (comando=61)
    {
        delay(10);
        sensor0 = analogRead(A2);

```

```

        Serial.println(sensor0, DEC);
    }
    else
    {
        goto pasoA3;
    }
    SerialReceive();
    break;

case 70:
    sensor0 = analogRead(A0);
    Serial.println(sensor0, DEC);
    pasoA4:
    SerialReceive();
    if (comando=71)
    {
        delay(10);
        sensor0 = analogRead(A1);
        Serial.println(sensor0, DEC);
    }
    else
    {
        goto pasoA4;
    }
    pasoA5:
    SerialReceive();
    if (comando=72)
    {
        delay(10);
        sensor0 = analogRead(A2);
        Serial.println(sensor0, DEC);
    }
    else
    {
        goto pasoA5;
    }
    SerialReceive();
    break;

case 100: // STOP
    SerialReceive();
    break;

default:
    SerialReceive();
    break;
}
}

void SerialReceive()
{
    while(Serial.available())
    {
        comando = Serial.read();
    }
}
}

```


Anexo 4. Código del software de aplicación desarrollado en lenguaje C#.

```
using System;
using System.IO;
using System.Diagnostics;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.IO.Ports;
using System.Windows.Forms;
using Graph = System.Windows.Forms.DataVisualization.Charting;

namespace SimpleSerial
{
    public partial class Form1 : Form
    {
        public string com;
        public string channel;
        public string filename;
        public int comandoAmicro;
        private int pointIndex = 0;
        private int pointIndex2 = 0;
        private int pointIndex3 = 0;
        Stopwatch clock = new Stopwatch();
        private Random random = new Random();
        string MenuBuffer;
        string RxString;
        string fecha;
        string logtime;
    }
}
```

```

int s;
int xt = 0;
bool sw;
List<int> Sensor1 = new List<int>();
List<int> Sensor2 = new List<int>();
List<int> Sensor3 = new List<int>();

public Form1()
{
    InitializeComponent();
    label4.Text = "Click en Setup para iniciar.";
}

private void buttonStart_Click(object sender, EventArgs e)
{
    byte data = 0;
    byte[] buffer = new byte[] { data };

    switch (comandoAmicro)
    {
        case 1:
            data = 10; //Envia .10 S1
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            break;
        case 2:
            data = 20; //Envia .20 S2
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            break;
        case 3:
            data = 30; //Envia .30 S3
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            break;
        case 4:
            data = 40; //Envia .40 S1yS2
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            s = 0;
            break;
        case 5:
            data = 50; //Envia .50 S1yS3
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
            buttonStop.Enabled = true;
            s = 0;
            break;
        case 6:
            data = 60; //Envia .60 S2yS3
            buffer = new byte[] { data };
            serialPort1.write(buffer, 0, 1);
            buttonStart.Enabled = false;
    }
}

```

```

        buttonStop.Enabled = true;
        s = 0;
        break;
    case 7:
        data = 70; //Envia .70 S1,S2ys3
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
        buttonStart.Enabled = false;
        buttonStop.Enabled = true;
        s = 0;
        break;
    default:
        break;
}
clock.Start();

sw = true;
buttonConect.Enabled = false;
}

private void buttonStop_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        byte data = 100; //Envia .100 Stop
        byte[] buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
        buttonStart.Enabled = true;
        buttonStop.Enabled = false;
        buttonConect.Enabled = true;
    }
    clock.Stop();
    logtime = clock.Elapsed.ToString("hh\\:mm\\:ss");
    clock.Reset();

    if (sw == true)
    {
        StreamWriter writer = new StreamWriter(filename);
        int i = 0;
        while (i < xt)
        {
            writer.Write(Sensor1[i]);
            writer.Write(",");
            writer.Write(Sensor2[i]);
            writer.Write(",");
            writer.WriteLine(Sensor3[i]);
            i++;
        }
        writer.Close();
        MessageBox.Show("Logging File: " + fecha + " Ha sido
generado" + "\r\n" + "Duracion del Log: "
+ logtime + "\r\n" + "Presione Start para empezar nuevo
Log", "LoggingDAQ");

        DialogResult dialogResult = MessageBox.Show("Generar LOG
para imprimir?", "LoggingDAQ", MessageBoxButtons.YesNo);
        if (dialogResult == DialogResult.Yes)
        {
            StreamWriter writer2 = new
StreamWriter(@"c:\numerosfromsensor.txt");
            i = 0;

```

```

        while (i < xt)
        {
            writer2.Write(Sensor1[i]);
            writer2.Write(",");
            writer2.Write(Sensor2[i]);
            writer2.Write(",");
            writer2.WriteLine(Sensor3[i]);
            i++;
        }
        writer2.Close();
        System.Diagnostics.Process.Start(@"C:\DAQLog.xls");
    }
    else if (dialogResult == DialogResult.No) { /*Sigue*/}

    Sensor1.Clear();
    Sensor2.Clear();
    Sensor3.Clear();

    this.Invoke(new EventHandler(NewChart));

    xt = 0;
    sw = false;
}
}

private void Form1_FormClosing(object sender,
FormClosingEventArgs e)
{
    if (serialPort1.IsOpen) serialPort1.Close();
}

private void DisplayText(object sender, EventArgs e)
{
    byte data = 0;
    byte[] buffer = new byte[] { data };

    switch (comandoAmicro)
    {
        case 1:
            label1.Text = RxString;
            numVal = Int32.Parse(RxString);
            Sensor1.Add(numVal);
            Sensor2.Add(0);
            Sensor3.Add(0);
            xt++;
            this.Invoke(new EventHandler(CharTick));
            verticalProgressBar1.Value = numVal;
            break;

        case 2:
            label2.Text = RxString;
            numVal = Int32.Parse(RxString);
            Sensor1.Add(0);
            Sensor2.Add(numVal);
            Sensor3.Add(0);
            xt++;
            this.Invoke(new EventHandler(CharTick));
            verticalProgressBar2.Value = numVal;
            break;

        case 3:
            label3.Text = RxString;

```

```

numVal = Int32.Parse(RxString);
Sensor1.Add(0);
Sensor2.Add(0);
Sensor3.Add(numVal);
xt++;
this.Invoke(new EventHandler(CharTick));
verticalProgressBar3.Value = numVal;
break;

case 4:
if (s == 0)
{
    label1.Text = RxString;
    numVal = Int32.Parse(RxString);
    Sensor1.Add(numVal);
    verticalProgressBar1.Value = numVal;
    s = 1;
    data = 41;
    buffer = new byte[] { data };
    serialPort1.write(buffer, 0, 1);
}
else if (s == 1)
{
    label2.Text = RxString;
    numVal = Int32.Parse(RxString);
    Sensor2.Add(numVal);
    Sensor3.Add(0);
    xt++;
    this.Invoke(new EventHandler(CharTick));
    verticalProgressBar2.Value = numVal;
    s = 0;
    data = 40;
    buffer = new byte[] { data };
    serialPort1.write(buffer, 0, 1);
}
break;

case 5:
if (s == 0)
{
    label1.Text = RxString;
    numVal = Int32.Parse(RxString);
    Sensor1.Add(numVal);
    verticalProgressBar1.Value = numVal;
    s = 1;
    data = 51;
    buffer = new byte[] { data };
    serialPort1.write(buffer, 0, 1);
}
else if (s == 1)
{
    label3.Text = RxString;
    numVal = Int32.Parse(RxString);
    Sensor2.Add(0);
    Sensor3.Add(numVal);
    xt++;
    this.Invoke(new EventHandler(CharTick));
    verticalProgressBar3.Value = numVal;
    s = 0;
    data = 50;
    buffer = new byte[] { data };
    serialPort1.write(buffer, 0, 1);
}

```

```

    }
    break;

case 6:
    if (s == 0)
    {
        label2.Text = RxString;
        numVal = Int32.Parse(RxString);
        Sensor1.Add(0);
        Sensor2.Add(numVal);
        verticalProgressBar2.Value = numVal;
        s = 1;
        data = 61;
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
    }
    else if (s == 1)
    {
        label3.Text = RxString;
        numVal = Int32.Parse(RxString);
        Sensor3.Add(numVal);
        xt++;
        this.Invoke(new EventHandler(CharTick));
        verticalProgressBar3.Value = numVal;
        s = 0;
        data = 60;
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
    }
    break;

case 7:
    if (s == 0)
    {
        label1.Text = RxString;
        numVal = Int32.Parse(RxString);
        Sensor1.Add(numVal);
        verticalProgressBar1.Value = numVal;
        s = 1;
        data = 71;
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
    }
    else if (s == 1)
    {
        label2.Text = RxString;
        numVal = Int32.Parse(RxString);
        Sensor2.Add(numVal);
        verticalProgressBar2.Value = numVal;
        s = 2;
        data = 72;
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
    }
    else if (s == 2)
    {
        label3.Text = RxString;
        numVal = Int32.Parse(RxString);
        Sensor3.Add(numVal);
        xt++;
        this.Invoke(new EventHandler(CharTick));
        verticalProgressBar3.Value = numVal;
    }

```

```

        s = 0;
        data = 70;
        buffer = new byte[] { data };
        serialPort1.Write(buffer, 0, 1);
    }
    break;

    default:
    break;
}

label8.Text = "Log Time: " +
clock.Elapsed.ToString("hh\\:mm\\:ss\\.ff");
}

private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    RxString = serialPort1.ReadLine();
    this.Invoke(new EventHandler(DisplayText));
}

private void buttonConect_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        sw = false;
        this.Invoke(new EventHandler(buttonStop_Click));
        serialPort1.Close();
        buttonConect.Text = ("Conectar");
        label4.ForeColor = Color.Black;
        label8.ForeColor = Color.Black;
        label4.Text = "Listo para conectar con: " + com;
        label8.Text = channel;
        label5.Font = new Font(label5.Font, FontStyle.Regular);
        label6.Font = new Font(label6.Font, FontStyle.Regular);
        label7.Font = new Font(label7.Font, FontStyle.Regular);
        label11.Text = ("--"); label2.Text = ("--"); label3.Text =
("--");
        label11.Enabled = false; label2.Enabled = false;
label3.Enabled = false;
        verticalProgressBar1.Value = 0;
        verticalProgressBar2.Value = 0;
        verticalProgressBar3.Value = 0;
        buttonStart.Enabled = false;
        buttonConf.Enabled = true;
        salir.Enabled = true;
        vistaToolStripMenuItem.Enabled = false;
    }
    else
    {
        vistaToolStripMenuItem.Enabled = true;
        label11.Enabled = false; label2.Enabled = false;
label3.Enabled = false;
        switch (comandoAmicro)
        {
            case (1): label11.Enabled = true; break;
            case (2): label2.Enabled = true; break;
            case (3): label3.Enabled = true; break;
            case (4): label11.Enabled = true; label2.Enabled =
true; break;

```

```

        case (5): label1.Enabled = true; label3.Enabled =
true; break;
        case (6): label2.Enabled = true; label3.Enabled =
true; break;
        case (7): label1.Enabled = true; label2.Enabled =
true; label3.Enabled = true; break;
        default: break;
    }

    serialPort1.PortName = Convert.ToString(com);
    serialPort1.BaudRate = 9600;
    serialPort1.DataBits = 8;
    serialPort1.Parity = Parity.None;
    serialPort1.StopBits = StopBits.One;
    serialPort1.Handshake = Handshake.None;
    serialPort1.RtsEnable = true;
    serialPort1.DtrEnable = true;
    serialPort1.Open();
    label4.Text = "Conectado a: LoggingDAQ";
    label8.Text = serialPort1.PortName + " " +
serialPort1.BaudRate + "," +
serialPort1.DataBits + "," + serialPort1.StopBits + "," +
serialPort1.Parity;
    label4.ForeColor = Color.Green;
    label8.ForeColor = Color.Green;
    buttonStart.Enabled = true;
    buttonConf.Enabled = false;
    buttonConect.Text = ("Desconectar");
    salir.Enabled = false;
}

}

private void Form1_Load(object sender, EventArgs e)
{
    label1.Enabled = false; label2.Enabled = false;
label3.Enabled = false;
    chart1.Visible = false; chart2.Visible = false;
chart3.Visible = false;
    this.Invoke(new EventHandler(NewChart));
}

private void ChartTick(object sender, EventArgs e)
{
    int numberOfPointsInChart = 20;
    int numberOfPointsAfterRemoval = numberOfPointsInChart - 1;

    chart1.Series["A1"].Points.AddXY(pointIndex,
Sensor1[pointIndex]);
    ++pointIndex;

    if (pointIndex < numberOfPointsInChart)
    {
        chart1.ChartAreas[0].AxisX.Maximum =
numberOfPointsInChart;
    }
    else chart1.ChartAreas[0].AxisX.Maximum = pointIndex;
    while (chart1.Series["A1"].Points.Count >
numberOfPointsInChart)
    {
        while (chart1.Series["A1"].Points.Count >
numberOfPointsAfterRemoval)
    {

```



```

        chart1.Series["A1"].Points.RemoveAt(0);
    }
    chart1.ChartAreas[0].AxisX.Minimum = pointIndex -
numberOfPointsAfterRemoval;
    chart1.ChartAreas[0].AxisX.Maximum =
chart1.ChartAreas[0].AxisX.Minimum + numberOfPointsInChart;
    }
    chart1.Invalidate();
    chart2.Series["A2"].Points.AddXY(pointIndex2,
Sensor2[pointIndex2]);
    ++pointIndex2;

    if (pointIndex2 < numberOfPointsInChart)
    {
        chart2.ChartAreas[0].AxisX.Maximum =
numberOfPointsInChart;
    }
    else chart2.ChartAreas[0].AxisX.Maximum = pointIndex2;
    while (chart2.Series["A2"].Points.Count >
numberOfPointsInChart)
    {
        while (chart2.Series["A2"].Points.Count >
numberOfPointsAfterRemoval)
        {
            chart2.Series["A2"].Points.RemoveAt(0);
        }
        chart2.ChartAreas[0].AxisX.Minimum = pointIndex2 -
numberOfPointsAfterRemoval;
        chart2.ChartAreas[0].AxisX.Maximum =
chart2.ChartAreas[0].AxisX.Minimum + numberOfPointsInChart;
    }
    chart2.Invalidate();
    chart3.Series["A3"].Points.AddXY(pointIndex3,
Sensor3[pointIndex3]);
    ++pointIndex3;

    if (pointIndex3 < numberOfPointsInChart)
    {
        chart3.ChartAreas[0].AxisX.Maximum =
numberOfPointsInChart;
    }
    else chart3.ChartAreas[0].AxisX.Maximum = pointIndex3;
    while (chart3.Series["A3"].Points.Count >
numberOfPointsInChart)
    {
        while (chart3.Series["A3"].Points.Count >
numberOfPointsAfterRemoval)
        {
            chart3.Series["A3"].Points.RemoveAt(0);
        }
        chart3.ChartAreas[0].AxisX.Minimum = pointIndex3 -
numberOfPointsAfterRemoval;
        chart3.ChartAreas[0].AxisX.Maximum =
chart3.ChartAreas[0].AxisX.Minimum + numberOfPointsInChart;
    }
    chart3.Invalidate();
}

private void NewChart(object sender, EventArgs e)
{
    pointIndex = 0;
}

```

```

        pointIndex2 = 0;
        pointIndex3 = 0;
        chart1.ChartAreas[0].AxisX.Minimum = 1;
        chart1.ChartAreas[0].AxisX.Maximum = 20;
        chart2.ChartAreas[0].AxisX.Minimum = 1;
        chart2.ChartAreas[0].AxisX.Maximum = 20;
        chart3.ChartAreas[0].AxisX.Minimum = 1;
        chart3.ChartAreas[0].AxisX.Maximum = 20;
        chart1.Series.Clear();
        chart2.Series.Clear();
        chart3.Series.Clear();
        chart1.Series.Add("A1");
        chart1.Series["A1"].ChartType = Graph.SeriesChartType.Line;
        chart1.Series["A1"].Color = Color.DarkBlue;
        chart1.Series["A1"].Borderwidth = 3;
        chart2.Series.Add("A2");
        chart2.Series["A2"].ChartType = Graph.SeriesChartType.Line;
        chart2.Series["A2"].Color = Color.DarkRed;
        chart2.Series["A2"].Borderwidth = 3;
        chart3.Series.Add("A3");
        chart3.Series["A3"].ChartType = Graph.SeriesChartType.Line;
        chart3.Series["A3"].Color = Color.DarkOrange;
        chart3.Series["A3"].Borderwidth = 3;
    }

    private void ButtonMenu_MouseEnter(object sender, EventArgs e)
    {
        MenuBuffer = label4.Text;
        if (sender == buttonConf) { label4.Text = "Click para
configurar LoggingDAQ"; }
        if (sender == salir) { label4.Text = "Click para cerrar y
salir de LoggingDAQ"; }
    }
    private void ButtonMenu_MouseLeave(object sender, EventArgs e)
    {
        if (sender == buttonConf) { label4.Text = MenuBuffer; }
        if (sender == salir) { label4.Text = MenuBuffer; }
    }
    private void buttonConf_Click(object sender, EventArgs e)
    {
        Form2 fr2 = new Form2();
        fr2.Show();
        this.Hide();
    }
    private void salir_Click(object sender, EventArgs e)
    {
        if (serialPort1.IsOpen) serialPort1.Close();
        this.Close();
        Application.Exit();
    }

    private void realTimeLogToolStripMenuItem1_Click(object sender,
EventArgs e)
    {
        label1.Enabled = false; label2.Enabled = false;
label3.Enabled = false;
        label1.Visible = false; label2.Visible = false;
label3.Visible = false;
        chart1.Visible = true; chart2.Visible = true; chart3.Visible
= true;
    }

```

```
private void logNumericoToolStripMenuItem_Click(object sender,
EventArgs e)
{
    label1.Enabled = true; label2.Enabled = true; label3.Enabled
= true;
    label1.Visible = true; label2.Visible = true; label3.Visible
= true;
    chart1.Visible = false; chart2.Visible = false;
chart3.Visible = false;
}
}
```